



UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Proyecto de Fin de Grado en Ingeniería Informática

HERRAMIENTA PARA EL DIMENSIONAMIENTO DE LÍNEAS DE PRODUCCIÓN DE LA INDUSTRIA CONSERVERA

MIGUEL FARIÑA CORES

Dirigido por: Alfonso Urquía Moraleda

Codirigido por Carla Martín Villalba

Curso: 2023/2024, convocatoria Septiembre



HERRAMIENTA PARA EL DIMENSIONAMIENTO DE LÍNEAS DE PRODUCCIÓN DE LA INDUSTRIA CONSERVERA

**Proyecto de Fin de Grado en Ingeniería Informática
de modalidad específica**

Realizado por: MIGUEL FARIÑA CORES

Dirigido por: Alfonso Urquía Moraleda

Codirigido por: Carla Martín Villalba

Fecha de lectura y defensa: Septiembre 2024

Agradecimientos

A mi mujer y mis hijos, por todo ese tiempo que os he robado y por el amor y la paciencia con la que me apoyáis.

A mi madre, mi hermano y demás familia, por el apoyo incondicional y por haberme inculcado desde siempre la cultura del esfuerzo y la perseverancia.

A mis compañeros y amigos por seguir apoyándome a pesar de estar ausente en tantos eventos.

Finalmente, agradecer al director y codirectora de este proyecto por su guía, consejos, paciencia y premura en el apoyo mostrado siempre que lo he necesitado.

Gracias.

Resumen

En la industria de maquinaria e instalaciones a medida, en particular en la industria conservera, los primeros pasos de las negociaciones para un nuevo proyecto son cruciales y requieren cada vez más de una rápida y ágil respuesta por parte del proveedor (fabricante-integrador de la maquinaria). El paradigma de diseños personalizados para cada cliente impide disponer de información del producto final hasta pasos más avanzados del proyecto, lo que va en contra de la necesidad de datos en el inicio de las negociaciones para la toma de decisiones. Es de gran importancia entonces disponer de medios para facilitar al proveedor la obtención de datos aproximados del producto que quiere ofrecer a su cliente en el menor tiempo posible.

El objetivo de este proyecto de fin de grado es analizar este problema y diseñar y construir una herramienta de dimensionamiento de líneas de producción de la industria conservera, que dará en llamar DraftLine, que permitirá al usuario final, idóneamente un técnico-comercial de la empresa fabricante-integradora de maquinaria, generar listados de maquinaria que constituyan un primer borrador de la línea de producción requerida por su cliente, con un esquema-diagrama de su distribución, características básicas de la línea (producción, consumos, coste..., es decir los requerimientos iniciales del cliente) y finalmente un informe con esta información que pueda presentar al cliente, a modo de borrador de presupuesto para el inicio de las negociaciones.

Esta herramienta deberá permitir generar variaciones de los resultados para adaptarse a los comunes cambios de requisitos en los inicios de estos proyectos, sin que ello implique demoras en el tiempo. Para poder realizar todo ello, la herramienta permitirá gestionar un simple catálogo de máquinas, clasificadas en familias, de las cuales la aplicación podrá tomar la selección necesaria para conformar la línea requerida en función de las condiciones de dimensionamiento solicitadas por el cliente.

Se mostrará en este documento un pequeño contexto sobre las líneas de producción de la industria conservera para continuar con los pasos detallados de los procesos de desarrollo de software seguidos hasta la obtención de la aplicación final y su documentación relacionada.

Palabras clave

DraftLine, industria conservera, línea de producción a medida, dimensionamiento de línea de producción, presupuesto ágil, Visual Basic.NET, Microsoft Visual Studio 2022, Microsoft visual Studio Installer Projects, SandCastle Help File Builder, Astah UML, Icofx, MSTest

Summary

In the custom machinery and equipment industry, especially in food canning, the first negotiations stages for new projects are critical and increasingly demand a quick response from the supplier (manufacturer-integrator of Machinery). Custom designs paradigm for each client means that detailed product information isn't available until later in the Project, which conflicts with the need for data at the outset of negotiations to make informed decisions. Therefore, having tools to quickly provide approximate product data to the supplier is crucial.

The goal of this PFG is to analyze this problem and design and build a production line sizing tool for the food canning industry, which will be called "DraftLine". This tool will allow the end user, ideally a technical-sales representative of the Machinery manufacturing-integrating company, to generate equipment lists that constitute an initial draft of the production line required by their client. It will include a diagrammatic layout of the distribution, basic characteristics of the line (production, consumption, cost... i.e., the initial requirements of the client), and finally, a report with this information that can be presented to the client as a draft budget for the start of negotiations.

This tool should allow generating variations of the results to adapt to common changes in requirements at the beginning of these projects, without causing delays. To achieve this, the tool will manage a simple catalog of machines classified into families, from which the application can select the necessary equipment to configure the required line based on the sizing conditions requested by the client.

This document will provide a brief context on production lines in the food canning industry, followed by detailed steps of the software development processes pursued until obtaining the final application and its related documentation.

Keywords

DraftLine, canning industry, custom production line, production line sizing, agile budget, Visual Basic.NET, Microsoft Visual Studio 2022, Microsoft Visual Studio Installer Projects, SandCastle Help File Builder, Astah UML, Icofx, MSTest

Índice

Índice	11
Índice de Figuras	15
Índice de Tablas.....	17
Capítulo 1 Introducción, objetivos y estructura.....	19
1.1 Introducción	19
1.2 Objetivos	23
1.3 Estructura	25
Capítulo 2 Dimensionamiento de líneas de producción de la industria conservera	27
2.1 Introducción	27
2.2 Diseño de una línea de producción	27
2.3 Ejemplos de líneas de producción.....	31
2.3.1 Línea general de conserva multiproducto:.....	31
2.3.2 Línea de conserva de atún	32
2.4 Ejemplos de familia de máquinas.....	33
2.4.1 Autoclaves esterilizadores estáticos	33
2.4.2 Encestadores/desencestadores	35
2.5 Soluciones existentes.....	36
2.5.1 Autodesk Factory Design Utilities	37
2.5.2 Siemens Tecnomatix Plant Simulation	38
2.5.3 Dassault Systèmes DELMIA.....	39
2.5.4 Conclusiones sobre las soluciones existentes.....	40
2.6 Conclusión.....	41
Capítulo 3 Análisis	43
3.1 Introducción	43
3.2 Límites del sistema, actores y sus objetivos	43
3.3 Casos de uso	44
3.4 Requisitos.....	51
3.4.1 Requisitos funcionales.....	52
3.4.2 Facilidad de uso	53
3.4.3 Fiabilidad	53
3.4.4 Rendimiento	53
3.4.5 Soporte	53
3.4.6 Requisitos adicionales de implementación	54
3.4.7 Requisitos adicionales de operaciones	54
3.4.8 Requisitos adicionales de empaquetamiento	54
3.5 Diagrama de casos de uso.....	55
3.6 Diagrama de modelo del dominio	56
3.7 Conclusiones	57

Capítulo 4	Diseño.....	59
4.1	Introducción.....	59
4.2	Modelo de diseño.....	60
4.2.1	Diagramas de secuencia.....	60
4.2.2	Diagrama de clases.....	69
4.2.3	Detalle de clases.....	69
4.3	Cumplimiento de requisitos.....	73
4.3.1	Facilidad de uso.....	73
4.3.2	Fiabilidad.....	73
4.3.3	Rendimiento.....	73
4.3.4	Soporte.....	73
4.3.5	Implementación.....	74
4.3.6	Operaciones.....	74
4.3.7	Empaquetamiento.....	74
4.4	Cobertura de requisitos.....	74
4.5	Conclusiones.....	76
Capítulo 5	Implementación y pruebas.....	77
5.1	Introducción.....	77
5.2	Entorno de desarrollo.....	77
5.2.1	Equipo hardware.....	77
5.2.2	Elección del lenguaje.....	78
5.2.3	Entorno software.....	79
5.3	Implementación de la solución.....	80
5.3.1	Fase 1: Desarrollo de la lógica de negocio.....	80
5.3.2	Fase 2: Diseño y lógica de la interfaz de usuario.....	83
5.3.3	Fase 3: Unión, depuración y ajustes de lógica de negocio y lógica de interfaz.....	86
5.3.4	Fase 4: Empaquetado y generación de documentación.....	86
5.4	Pruebas.....	87
5.4.1	Pruebas unitarias.....	88
5.4.2	Pruebas de integración.....	88
5.5	Conclusiones.....	88
Capítulo 6	Conclusiones y trabajos futuros.....	89
6.1	Introducción.....	89
6.2	Conclusiones.....	89
6.3	Trabajos futuros.....	91
Bibliografía.....		93
Glosario.....		95
Anexos.....		96
Anexo A	Manual de usuario.....	96
	Índice del manual de usuario.....	97

A.1	Introducción	98
A.2	Requerimientos del sistema.....	98
A.3	Instalación.....	99
A.4	Interfaz del usuario	100
A.5	Crear y gestionar borradores de presupuestos de líneas de producción.....	113
A.6	Crear y gestionar catálogo de máquinas	116
A.7	Solución de problemas comunes.....	119
A.8	Soporte técnico y actualizaciones.....	120
A.9	Glosario de términos del manual de usuario	120
Anexo B	Código fuente.....	121
B.1	Clase FormPrincipal.....	121
B.2	Clase ServicioMaquinas	130
B.3	Clase ServicioLineas	135
B.4	Clase Dimensionar.....	137
B.5	Clase Informe	139
B.6	Clase FamiliaMaquina	142
B.7	Clase Maquina.....	144
B.8	Clase Linea	146
B.9	Clase Condiciones.....	148
B.10	Clase Imprimir.....	149
B.11	Clase FormFamilia.....	150
B.12	Clase FormMaquina.....	152
B.13	Clase FormSelecFam	154
B.14	Clase RepresentacionFamilia	160
B.15	Clase FormDatosLinea	162
B.16	Clase FormCondiciones.....	163
B.17	Clase FormAcercaDe.....	166
Anexo C	Documentación de código.....	167
Anexo D	Código fuente pruebas.....	377
D.1	Clase ServicioMaquinasPrueba.....	377
D.2	Clase ServicioLineasPrueba.....	380
D.3	Clase FamiliaMaquinaPrueba.....	381
D.4	Clase MaquinaPrueba	382
D.5	Clase LineaPrueba	384
D.6	Clase CondicionesPrueba	384
Anexo E	Documentación de código pruebas.....	385

Índice de Figuras

Figura 1.1. Línea de preparación de anchoa (tacore, 2024).	20
Figura 2.1. Ejemplo máquina. Empacadora (Hermasa, 2024).	29
Figura 2.2. Esquema básico conjunto general de líneas de producción.	29
Figura 2.3. Esquema líneas de producción para tratamiento de insumos.	30
Figura 2.4. Línea multiproducto ejemplar (Emerito, 2024).	32
Figura 2.5. Línea de conserva de atún (tacore, 2024).	33
Figura 2.6. Autoclaves esterilizadores horizontales estáticos (tacore, 2024).	34
Figura 2.7. Diferentes medidas para escoger una máquina de una familia de máquinas autoclaves (tacore, 2024).	34
Figura 2.8. Encestonado/desencestonado de envases (Emerito, 2024).	35
Figura 2.9. Diferentes tipos de contenedores para esterilización.	35
Figura 2.10. Factory Design en AutoDesk Inventor (Autodesk, 2024).	37
Figura 2.11. Plant Simulation de Siemens (Siemens, 2024).	38
Figura 2.12. Factory Layout Designer de DELMIA (Dassault, 2024).	39
Figura 2.14. Interconexión de diferentes etapas del proyecto completo en Autodesk Factory Utilities.	40
Figura 2.13. Flujo de trabajo y comunicación de información en Siemens Tecnomatix.	40
Figura 3.1. Diagrama de casos de uso.	55
Figura 3.2. Diagrama de Modelo de Dominio.	56
Figura 4.1. Diagrama de secuencia del inicio de la aplicación.	60
Figura 4.2. Diagrama de secuencia del caso de uso: Nuevo catálogo.	61
Figura 4.3. Diagrama de secuencia del caso de uso: Exportar catálogo.	62
Figura 4.4. Diagrama de secuencia del caso de uso: Importar catálogo.	62
Figura 4.5. Diagrama de secuencia del caso de uso: Guardar catálogo.	63
Figura 4.6. Diagrama de secuencia del caso de uso: Nueva familia de máquinas.	63
Figura 4.7. Diagrama de secuencia del caso de uso: Nueva máquina.	64
Figura 4.8. Diagrama de secuencia del caso de uso: Editar familia o máquina.	64
Figura 4.9. Diagrama de secuencia del caso de uso: Nueva línea.	65
Figura 4.10. Diagrama de secuencia del caso de uso: Exportar línea.	65
Figura 4.11. Diagrama de secuencia del caso de uso: Importar línea.	66
Figura 4.12. Diagrama de secuencia del caso de uso: Editar datos línea.	66
Figura 4.13. Diagrama de secuencia del caso de uso: Selección de familias de máquinas para la línea.	67
Figura 4.14. Diagrama de secuencia del caso de uso: Editar condiciones.	67
Figura 4.15. Diagrama de secuencia del caso de uso: Imprimir informe.	68
Figura 4.16. Diagrama de secuencia del caso de uso: Mostrar manual de usuario.	68
Figura 4.17. Diagrama de clases simplificado.	69
Figura 5.1. Primeras versiones de la interfaz de usuario.	84
Figura 5.2. Captura de la pantalla principal.	84
Figura 5.3. Captura de formulario para editar condiciones.	85
Figura 5.4. Ejemplo de tooltip de ayuda sobre un botón.	85
Figura A.1. Pantalla bienvenida de la instalación.	99
Figura A.2. Selección carpeta de instalación.	99
Figura A.3. Confirmación de la instalación.	100
Figura A.4. Componentes pantalla principal.	101
Figura A.5. Menú principal.	101

Figura A.6. Pantalla principal en modo dimensionar línea.	103
Figura A.7. Botones acceso rápido en modo dimensionar línea.	104
Figura A.8. Visor informe línea dimensionada.	105
Figura A.9. Pantalla principal en modo editar catálogo.	105
Figura A.10. Botones acceso rápido modo editar catálogo.	106
Figura A.11. Visor catálogo.	107
Figura A.12. Barra de estado de la pantalla principal.	107
Figura A.13. Ventana "Datos de la línea a dimensionar".	108
Figura A.14. Ventana "Selección de familias de máquinas de la línea a dimensionar.	108
Figura A.15. Ventana "Condiciones para el dimensionamiento".	109
Figura A.16. Ventana "Nueva familia de máquinas".	109
Figura A.17. Ventana "Nueva máquina".	110
Figura A.18. Ventanas "Editar".	110
Figura A.19. Ventana "Importar".	111
Figura A.20. Ventana "Exportar".	111
Figura A.21. Ventana "Imprimir".	112
Figura A.22. Ventana "Acerca de..."	112
Figura A.23. Cambio orden de familias de máquinas seleccionadas al arrastrar una de ellas.	114
Figura A.24. Botón eliminar (botón derecho de ratón sobre representación).	114
Figura A.25. Información de una familia de máquinas seleccionada al pasar el ratón por encima.	114

Índice de Tablas

Tabla 3.1. Objetivos del actor principal	44
Tabla 3.2. Caso de uso UC1: Crear nuevo catálogo de Máquinas	44
Tabla 3.3. Caso de uso UC2: Exportar catálogo	45
Tabla 3.4. Caso de uso UC3: Importar catálogo	45
Tabla 3.5. Caso de uso UC4: Guardar catálogo.....	46
Tabla 3.6. Caso de uso UC5: Crear nueva Familia de máquinas	46
Tabla 3.7. Caso de uso UC6: Crear nueva Máquina.....	47
Tabla 3.8. Caso de uso UC7: Editar Familias o Máquinas del catálogo actual	47
Tabla 3.9. Caso de uso UC9: Crear nueva Línea.....	48
Tabla 3.10. Caso de uso UC9: Exportar Línea.....	48
Tabla 3.11. Caso de uso UC10: Importar Línea	49
Tabla 3.12. Caso de uso UC11: Editar Datos Línea.....	49
Tabla 3.13. Caso de uso UC12: Editar esquema de Familias de máquinas de la Línea.....	50
Tabla 3.14. Caso de uso UC13: Editar condiciones de dimensionamiento	50
Tabla 3.15. Caso de uso UC14: Imprimir informe de Línea dimensionada	51
Tabla 3.16. Caso de uso UC15: Ver Manual de usuario	51
Tabla 4.1. Matriz Requisitos-Componentes.....	75

Capítulo 1

Introducción, objetivos y estructura

1.1 INTRODUCCIÓN

En la industria productiva en general, y en la conservera en particular, los fabricantes necesitan continuamente adaptar la producción de sus plantas a las cambiantes demandas del mercado, ya sea por aumento de la demanda (incrementar sus líneas de producción o aumentar su capacidad) como por cambios solicitados por sus departamentos de ventas y marketing en cuanto a formatos, tamaños y presentación de sus productos. Pero deben comprobar que estas adaptaciones son viables antes de acometer estos nuevos proyectos. Es por ello que las empresas fabricantes e integradoras de líneas de producción necesitan de capacidades para dar una respuesta rápida a las solicitudes de sus clientes.

Una nueva línea de producción para una planta de un cliente normalmente se engloba en un proyecto mayor. El cliente necesita información a la mayor brevedad posible de lo que el mercado (los fabricantes e integradores de líneas) pueden ofrecerle para cumplir con sus requerimientos. Pero esta celeridad necesaria entra en conflicto con la necesidad de líneas de producción a medida del cliente, cuyo diseño requiere de un trabajo y un tiempo previo antes de poder entregar toda la información necesaria (coste, características productivas, plazos, ...).

Estas líneas de producción son en realidad cadenas de fabricación compuestas por módulos (máquinas) que aportan valor a los productos que en ellas se producen, iniciándose la producción en el primer módulo al recibir insumos (que podrán ser externos o provenientes de otras líneas de producción) y finalizándose en el último módulo de donde saldrán los elementos producidos, después de haber recibido los correspondientes añadidos de insumos y/o procesos necesarios en cada uno de los módulos intermedios. Cada uno de estos módulos estará compuesto realmente por una o varias máquinas específicas (o sistema, o puesto de operario) que realizan un tipo de función en particular. Denominaremos familia de máquinas al conjunto de máquinas que tienen la misma finalidad aún con distintas características. Es decir, pertenecerán a una misma familia de máquinas aquellas que, aun produciendo el mismo tipo de trabajo, lo hacen con diferente velocidad ("producción"), diferente número de operarios necesarios ("operarios"), distinto consumo eléctrico ("potencia eléctrica"), distinto coste de fabricación o compra ("valor coste") y distinto precio de venta ("valor venta"), entre otras características.

Dentro de las industrias alimentarias, se pueden definir como conserveras aquellas cuyo principal cometido es la conservación de alimentos en envases sin necesidad de refrigeración, manteniendo sus cualidades organolépticas y de seguridad alimentaria, generalmente mediante procesos térmicos de esterilización. En ellas, pueden darse muy diversos tipos de líneas de producción, con diferentes finalidades, requisitos, tipos de funcionamiento, etc. Se muestra a continuación una clasificación general sin ánimo de ser exhaustiva:

- Según el tipo de alimento a tratar: conservera de vegetales, lácteas, cárnicas, de la pesca o derivada de ella, etc.
- Según la zona de planta/etapa de producción: Líneas de insumos, líneas de almacén, líneas de cocción, líneas de congelado, líneas de conserva (propiamente dicha, donde se realiza el cierre de envase y esterilización o tratamiento del producto), etc.
- Según los métodos de trabajo: Líneas de operación manual, semiautomática, totalmente automática/robotizada.
- Según el tipo de producción: Líneas de alta producción de un solo tipo de producto, líneas multiproducto, trabajo continuo o por lotes.
- Según los elementos físicos a manipular: Líneas de cajas, de estuches, de envases de hojalata, de vidrio, de aluminio, producto alimentario en contacto directo, ...
- Otros.

Como ejemplo para el lector, se muestran en el apartado 2.3 algunas líneas de producción y, en el apartado 2.4, algunas familias de máquinas, con una pequeña descripción e imágenes. Valga de ejemplo, en este apartado, la siguiente línea de preparación de anchoa para semiconserva:

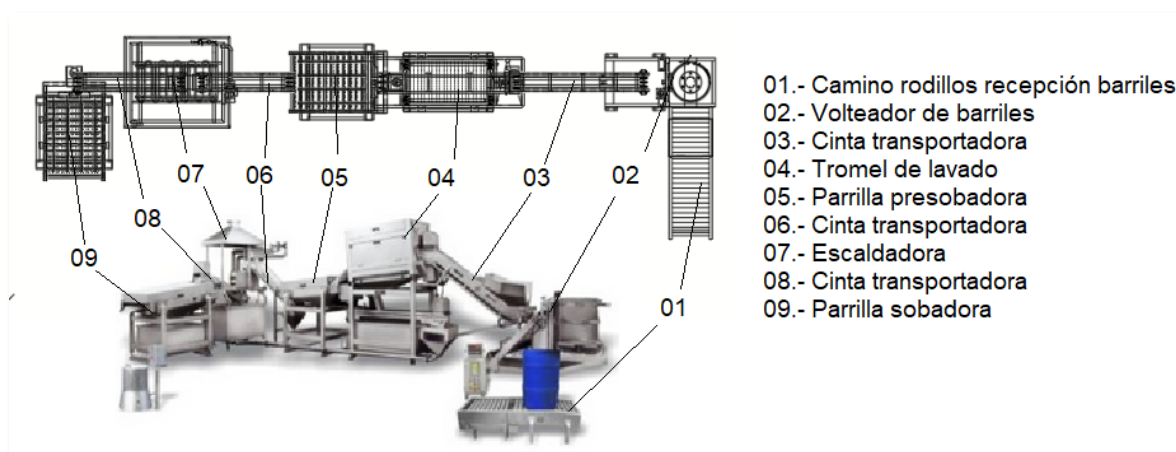


Figura 1.1. Línea de preparación de anchoa (tacore, 2024).

Cada elemento numerado indica una familia de máquinas que podría ejercer la funcionalidad requerida en esa posición de la lista. La selección de las máquinas en particular a instalar en esas posiciones dará lugar a la línea de producción determinada, con sus características definidas por las del conjunto de máquinas instaladas.

Pueden verse otro tipo de líneas de producción y familias de máquinas en: tacore (2024), Maconse (2024), Gaitech (2024), Hermasa (2024), JBT Corporation (2024), Emerito (2024), y otros. Para información sobre los procesos a realizar en diferentes líneas se pueden consultar (Sánchez 2003), aunque también incluye otros procesos de elaboración de alimentos no necesariamente de la industria conservera, pero muestra de forma clara que se deben seguir diversos pasos para cada proceso; pasos que definen una línea de producción. Además, se puede consultar más información en la Asociación de fabricantes de conserva “Anfaco Cecopesca” (Anfaco, 2024).

El diseño de una línea en función de sus familias de máquinas determinará su funcionalidad; es decir, qué es lo que producen. La selección de las máquinas de cada tipo de familia de máquinas determinará las características productivas de la línea (producción, consumo, número de operarios necesarios, coste, ...). El diseño de una línea de producción así para un potencial cliente requiere de un trabajo completo de definición en el que se deberán tener en cuenta múltiples variables además de las propias características productivas de la línea, y que, generalmente, no están del todo claras al inicio de la recopilación de requerimientos. Entre otras podemos destacar:

- Volumen ocupado por la futura línea y su integración en cuanto a espacio en la planta productiva del cliente.
- Consideraciones de diseño a nivel estético.
- Consideraciones de protocolos de comunicación y funcionamiento acordes a la cultura tecnológica de la empresa cliente.
- Consideraciones de mantenimiento y limpieza.
- Consideraciones de durabilidad de los elementos.
- Arreglo a diferentes normativas (maquinaria, seguridad, eléctricas, sociales, ...), tanto internas al cliente, como generales por nación o región.
- Adaptación a solicitudes logísticas del cliente.
- Otras.

Es por ello que, si se le quiere ofrecer al cliente una rápida respuesta en cuanto a un borrador de presupuesto para que pueda tener en cuenta la oferta del fabricante-integrador de líneas de producción, no es posible esperar a que dicha línea esté completamente diseñada. De hecho, sería ideal poder ofrecer un borrador de presupuesto inicial de manera instantánea y contar con la suficiente flexibilidad para modificar dicho borrador en tiempos muy cortos en caso de que el cliente desee distintas configuraciones o sus requisitos generales vayan cambiando (lo cual es muy habitual). De esta manera se evitarían gastos asociados al comienzo de diseño de una versión de una línea que quizás al cliente no le interese, además de facilitarle al propio cliente la toma de decisiones por disponer de la información necesaria en un menor tiempo.

En la actualidad, esta “información temprana” o no existe, y al cliente se le entregará un presupuesto más avanzado y detallado, tras cierto trabajo por parte del equipo técnico-comercial del fabricante-integrador (con el consiguiente consumo de tiempo y recursos) o se realiza de forma no formal, a mano, en base a la experiencia del equipo comercial (requiere de personal muy experimentado y no genera heurística de manera automática).

Es en esta tesitura donde se encuentra la razón de ser de la herramienta “DraftLine” que aquí se desarrollará. Permitirá generar un catálogo básico de máquinas (las máquinas, sistemas, puestos de operarios, ..., que la empresa integradora puede fabricar o comprar para integrar), pertenecientes a diferentes categorías (familias de máquinas). Con este catálogo generado, la aplicación permitirá diseñar un esquema de las familias de máquinas que compondrían la línea requerida (de manera muy simple y visual), especificar los requerimientos básicos de producción y características de la línea esperados por el cliente, y generará un listado de las máquinas que compondrían dicha línea, en forma de un borrador de presupuesto que se podría, si así se desea, entregar directamente al cliente. Todo ese proceso, sin contar la alimentación de datos al catálogo, no debería durar más de unos pocos minutos, y cualquier cambio de requerimientos posterior, aún menos tiempo.

Otras herramientas utilizadas para estos trabajos van más allá en el nivel de detalle y, aunque son útiles y necesarias, por esa misma razón pierden la utilidad que se busca en esta herramienta: abstracción, acuerdo, flexibilidad y, sobre todo, velocidad. Requieren más horas de trabajo y más tiempo para obtener un resultado que ofrecer al cliente; tiempo que es muy valioso en los primeros compases de una negociación de este tipo de proyectos.

1.2 OBJETIVOS

La finalidad del presente proyecto es realizar el análisis del problema, diseñar la solución a ese problema, construirla (codificarla) y probarla, siguiendo una metodología de desarrollo iterativa e incremental, como se describirá en los siguientes capítulos, con el fin de obtener, finalmente, la herramienta denominada “DraftLine” (de borrador de línea), que sirva como un ágil primer paso en el comienzo de un proyecto de diseño de una línea de producción industrial ad-hoc, en particular, de la industria conservera, aunque su uso se podría extender a otras industrias cuya estructura productiva mantenga una estructura similar.

El objetivo a conseguir puede verse desde dos prismas diferentes, según las partes implicadas:

- Para el cliente: Necesita un borrador de presupuesto de la línea solicitada, con una solución a sus requerimientos y un coste aproximados, dentro de un margen acordado. Además, es posible que necesite diferentes versiones de la misma solución, para escoger la más adecuada según sus criterios.
- Para el fabricante-integrador: Necesita poder generar un esquema de una línea de producción a ofrecer, con un nivel de abstracción suficiente, manteniendo un registro de la documentación generada para poder realizar variaciones, mostrando sus características y costes, con seguridad de que serán posibles de conseguir a posteriori. Todo ello en un tiempo mínimo, con el menor gasto de recursos.

El problema al que intenta dar respuesta la aplicación aquí desarrollada se da cuando un cliente (o potencial cliente) solicita información (cotización y tipo de solución) a una empresa fabricante-integradora de líneas de producción para una nueva línea de producción (o parte de ésta). Como se ha pretendido mostrar, estas líneas no son simples, y la selección de componentes de las mismas no es trivial. Se trata de líneas a medida, con componentes escogidos o incluso fabricados según requerimientos del cliente, y cuyos requerimientos varían mucho durante el propio proceso de diseño ya que la propia necesidad del cliente es un procedimiento vivo (adaptarse a la cambiante demanda del mercado). Por ello, para poder obtener una información completa y concisa sobre cómo será la línea a fabricar, sus costes, etc., es necesario un proceso de definición y diseño que, por definición, requiere de recursos, en particular del tiempo, que generan un gasto para el fabricante-instalador e incluso pueden dar lugar a la pérdida de la venta por el retraso en la entrega de la información en relación a la competencia. El problema es, entonces, que no es sencillo entregar el mínimo de información suficiente sobre la línea a ofrecer al cliente en un tiempo adecuado, manteniendo

coherencia con lo que finalmente se le podrá entregar, y generando el orden y concordancia adecuada en cuanto a información creada.

En este proyecto se conseguirá una aplicación Windows simple y clara que produzca el resultado requerido, que será un borrador de una línea de producción bajo los requerimientos de un cliente, con sus características y coste, tanto individuales como del total de la línea. La aplicación será simple, tanto en su propia naturaleza, al abstraer todos los requerimientos y diseño de una línea de producción a los niveles más básicos que permitan ofrecerle al cliente una primera y veraz idea de qué es lo que se está negociando y su coste; como en el propio uso de la herramienta, con una interfaz basada en los más que conocidos formularios Windows, así como un esquema gráfico y modificable para la línea. Una vez disponible el catálogo de máquinas, se podrán obtener borradores de líneas de producción de manera instantánea tras introducir los datos de los requerimientos. Tan sólo se requerirán 3 pasos para obtener un borrador:

- Introducir los datos informativos del propio proyecto (nombre del proyecto, del cliente, descripción de la línea, ...).
- Realizar el esquema de la línea en base a sus familias de máquinas, añadiendo familias de máquinas de las existentes en el catálogo y ordenándolas según las necesidades de la línea.
- Seleccionar los requerimientos a tener en cuenta en el dimensionamiento. En este caso en particular de líneas de producción de conserva: seleccionando la producción mínima requerida para la línea, la cantidad de máquinas que podrán seleccionarse para trabajar en paralelo sumando sus producciones, y el valor principal (y el secundario) a minimizar en cuanto a otras características de la misma.

Con esa información introducida, la aplicación generará directamente el informe-borrador de presupuesto de la línea así dimensionada.

Se podrán generar, consultar, modificar, exportar e importar el catálogo de familias y máquinas. El usuario podrá crear nuevas familias de máquinas, para las cuales también podrá generar nuevas máquinas, así como editar los valores de las características de cualquiera de ellas. Dado un conjunto de éstas, puede determinar exportarlas como un catálogo aparte (que también podría utilizar como copia de respaldo del catálogo actual), e importarlas cuando considere necesario.

Se podrán generar, consultar, modificar, exportar, importar y actualizar diferentes líneas de producción. Una vez dimensionada una línea, además de poder imprimirla en una de las impresoras del sistema donde se aloje la aplicación (impresora física, pdf, ...), también podrá exportar dicha configuración de línea para

efectos de archivo o para recuperarla posteriormente. Esto permitirá actualizar borradores antiguos con los valores de las características de las máquinas actuales si éstos han cambiado (por ejemplo, aumento del coste de venta).

Existirá la opción de utilizar diferentes catálogos exportados para diferentes tipos de líneas o clientes. Ya que existe la opción de exportar el catálogo sobre el que se está trabajando, se podrían generar diferentes catálogos con distintos tipos de máquinas que se utilizarían según la línea a dimensionar. El usuario cargaría el catálogo que necesite antes del dimensionamiento. Otra opción es generar catálogos con las mismas máquinas, pero diferentes costes (u otras características) para su uso en diferentes clientes.

Por último, también se podrán utilizar líneas exportadas como plantillas para futuras líneas. Aunque la configuración de una línea es muy rápida (los 3 pasos comentados anteriormente), si el usuario así lo desea, puede generar diferentes configuraciones y guardarlas para luego utilizarlas como plantillas para, por ejemplo, generar diferentes líneas para un mismo usuario, o la misma línea para diferentes usuarios, así como otras posibilidades.

1.3 ESTRUCTURA

Se estructura el presente documento en seis capítulos, cinco anexos y el contenido digital que acompaña a esta memoria:

- Capítulo 1. Introducción, Objetivos y Estructura: El capítulo actual, donde se muestra una pequeña descripción del contexto de negocio (líneas de producción de la industria conservera) para el cual se orienta el proyecto, los objetivos que se buscan y la estructura del documento.
- Capítulo 2.- Dimensionamiento de líneas de producción de la industria conservera. Se realiza aquí una introducción sobre las características y complicaciones de un proyecto industrial para el diseño de una línea de producción, mostrando algunos ejemplos de líneas, familias de máquinas y máquinas a modo de contexto. Se presenta el problema al que se quiere dar respuesta con el presente proyecto (el primer paso en la definición de una línea de producción desde el punto de vista del proveedor de la línea) y también se muestran algunas soluciones existentes, sus peculiaridades y la razón de por qué no logran dar respuesta al problema planteado.
- Capítulo 3.- Análisis. Ya desde el punto de vista de desarrollo del software, se lleva a cabo aquí la fase de análisis, donde se definirán los límites del sistema a desarrollar, sus actores y se estudiarán los casos de uso que serán necesarios para la funcionalidad esperada de la aplicación a desarrollar. A partir de estos resultados, y de los datos anteriores sobre el planteamiento del problema, se

recopilan los requisitos para la aplicación, basándose en el modelo FURPS+ y se muestran los diagramas de caso de uso y de modelo de dominio como figura de las conclusiones obtenidas del análisis.

- Capítulo 4.- Diseño: Donde se describen los pasos seguidos para crear la representación, modelos, de la aplicación, basados en los resultados del análisis y que se utilizarán para la codificación. Para ello se realizan diagramas de secuencia, partiendo de los casos de uso, diagramas de clases, partiendo de los anteriores y del modelo de dominio y toma de decisiones para el cumplimiento del resto de requisitos no cubiertos por los elementos anteriores. Finalmente se utiliza una matriz de cobertura de requisitos como comprobación del diseño frente al análisis.
- Capítulo 5.- Implementación y pruebas: Se expone aquí el entorno hardware y software utilizado para esta etapa, así como las pruebas realizadas durante la codificación y una vez finalizada ésta.
- Capítulo 6.-Conclusiones y trabajos futuros: Se resumen aquí las conclusiones finales sobre el trabajo, sus objetivos y las metas alcanzadas. Además, se recogen varios planteamientos para posibles ampliaciones del proyecto e integraciones de la aplicación en otras soluciones software mayores.
- Anexo A: Manual de Usuario. Contiene la información del manual que explica a un posible usuario el funcionamiento de la aplicación.
- Anexo B: Código fuente. Contiene el código en vb.NET de las clases y formularios que componen la aplicación.
- Anexo C: Documentación de código: Resumen de la documentación detallada y estructurada de las clases y formularios de la aplicación, con descripciones, referencias y explicación de su funcionamiento y sus métodos.
- Anexo D: Código fuente pruebas: Contiene el código en vb.NET de las clases utilizadas para realizar las pruebas unitarias y de integración, usando el marco de pruebas MSTest (Microsoft Testing Framework)
- Anexo E: Documentación de código de pruebas: Resumen de la documentación detallada y estructurada de las clases creadas para las pruebas.
- Contenido digital: En soporte físico adjunto a esta memoria se incluye una copia en pdf de esta memoria, la “Autorización para la publicación” (Modelo B.5), el código fuente, organizado como proyecto Visual Studio, las dos documentaciones de código (de la aplicación y de las pruebas) en formato Web HTML, y el archivo instalador de la propia aplicación desarrollada en este proyecto.

Capítulo 2

Dimensionamiento de líneas de producción de la industria conservera

2.1 INTRODUCCIÓN

Aunque la funcionalidad de la aplicación puede no restringirse a la industria conservera, e incluso, por su simplicidad, ir más allá de la propia industria en general para aplicarse a otros problemas que por su semejanza con líneas de elementos seleccionables, con varios elementos específicos para elegir en cada elemento seleccionable (familias de máquinas y sus máquinas), se expondrá a continuación el contexto alrededor de su aplicación directa en el servicio a la industria conservera, por ser ésta la que motivó la necesidad de la herramienta y formar parte de la experiencia que posee el que escribe.

Se muestran a continuación los conceptos básicos y un ligero contexto que facilite la visión del problema y su solución, así como entender el papel, en cualquier proyecto de cierta importancia de diseño de líneas de producción, que desempeñaría la aplicación aquí desarrollada. También se intentará mostrar que otras soluciones disponibles y utilizadas en el sector no llegan a cumplir totalmente con lo que se obtendría con esta herramienta, centrándose más en los siguientes pasos de un proyecto de este tipo.

2.2 DISEÑO DE UNA LÍNEA DE PRODUCCIÓN

La industria conservera actual se basa en plantas de producción que, dependiendo de su tamaño y ámbito, pueden ser de mayor o menor producción y con más o menos variedad de productos producidos.

Una planta de producción continua fabrica un alto volumen de productos estandarizados. En el caso de la industria conservera, se podría afinar más la clasificación y denominarlas plantas de producción por lote: producen lotes que varían en tamaño desde unas decenas de miles de unidades de envases a cientos de miles (Sipper y Bulfin, 1998, p. 11). Estos lotes podrán venir dados por diferencia de materia prima (el tipo de pescado a conservar, por ejemplo), por cosechas o temporadas de captura (cuando la materia prima se obtiene sólo en ciertas temporadas, como las distintas cosechas en la conserva vegetal o los distintos productos del mar en fechas determinadas en las que es posible su captura), o incluso por tipo de producto final (tiradas de unidades limitadas, promocionales, ...) (Sánchez 2003).

Podemos definir ahora el concepto de línea de producción: "Una línea de producción es un conjunto organizado de procesos y recursos interconectados, diseñados para fabricar productos específicos de manera eficiente y continua. Cada estación en la línea de producción realiza una operación específica que se suma al valor del producto final." Traspuesto y traducido desde (Groover 2015, ch 38.2, p.1000).

En nuestro caso, cada estación en una línea de producción será una máquina, perteneciente a una familia de máquinas, con unas características propias. Para la abstracción necesaria para un primer dimensionamiento de una línea de producción, se han reducido estas características a las básicas y comunes entre muy diferentes tipos de máquinas y líneas, para la aproximación de la herramienta a desarrollar en este proyecto:

- **Producción:** Reflejada en kilogramos por hora (kg/h) de producto a tratar. Tanto si se trata de máquinas que procesan directamente el producto principal, como si se trata de una variación del mismo, u otros auxiliares (envases llenos o vacíos, cajas, palets, ...) se podrá realizar una equivalencia a kg/h del producto principal. Es la principal característica a tener en cuenta ya que definirá la producción total de la línea que a la postre es la finalidad buscada por el cliente.
- **Operarios:** Cantidad de personal trabajador necesario para operar el equipo. Según la naturaleza de la máquina y su grado de automatización, este valor puede variar significativamente. Es una característica importante ya que el número total de operarios necesarios para la línea de producción es un importante indicador para el cliente en el estudio de viabilidad económica del proyecto que incluya la línea de producción a dimensionar.
- **Potencia eléctrica:** Reflejada en kilovatios (kW). Aunque se podría entrar en mucho detalle y diferenciar entre potencia instalada (fija y estática) y potencia consumida (variable en el tiempo en función del momento de funcionamiento de la línea), tener en cuenta la potencia eléctrica instalada es un buen indicador del consumo eléctrico esperado. Valor de gran importancia también en la viabilidad del proyecto.
- **Valor de coste:** Reflejado en euros (€). Representa el coste de compra o fabricación de la máquina para el fabricante-integrador de líneas de producción. Teniendo en cuenta el valor de venta, será un indicador de la viabilidad del proyecto desde el punto de vista del proveedor de la línea.
- **Valor de venta:** Reflejado en euros (€). Representa el valor por el cual se vende el equipo al cliente final. Por supuesto, es, quizás, el indicador más importante, después de la producción requerida, para que la selección de la línea para el proyecto sea un éxito.

Sirva como ejemplo, la máquina “Tunipack 500”, perteneciente a la familia de máquinas “Empacadoras de atún”. En la imagen que se muestra (véase la Figura 2.1.), se puede observar a la operaria necesaria para la máquina. Sus características básicas serían:



Figura 2.1. Ejemplo máquina. Empacadora (Hermasa, 2024).

- **Producción:** 500 envases RO-80 por minuto. Aprox. 56 gramos de producto por envase. Entonces: 1680 kg/h.
- **Operarios:** 1 operario introduciendo el producto.
- **Potencia Eléctrica:** 14.5kW (en ficha técnica).
- **Valor coste:** El valor de compra por parte del integrador es muy variable, debido a acuerdos comerciales, opciones y otros. Como valor aproximado podemos indicar: 100000 €.
- **Valor venta:** Aquí entra en juego el margen comercial que el integrador desee incluir, además de gastos adicionales de gestión y otros. Podríamos indicar un 30% por encima del valor de coste: 130000€.

En cualquier fábrica de la industria, en especial conservera, son necesarias múltiples líneas de producción para su funcionamiento, tanto para las distintas partes del proceso completo como para diferentes productos, formatos e incluso temporadas de producción.

Poniendo un ejemplo básico en la industria que nos ocupa, un conjunto de líneas de producción habitual en una planta conservera sería la mostrada en la Figura 2.2:

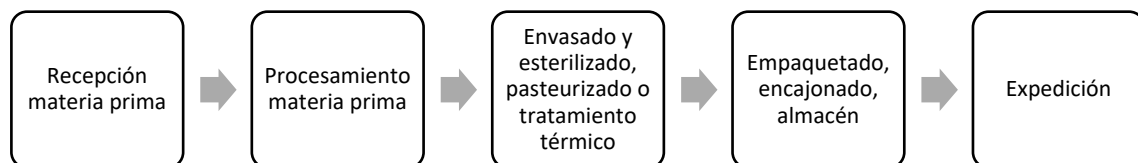


Figura 2.2. Esquema básico conjunto general de líneas de producción.

A su vez, cada una de estas etapas podría contener diferentes líneas independientes. La primera etapa podría disponer en paralelo de las siguientes líneas (y otras); por ejemplo (véase Figura 2.3):

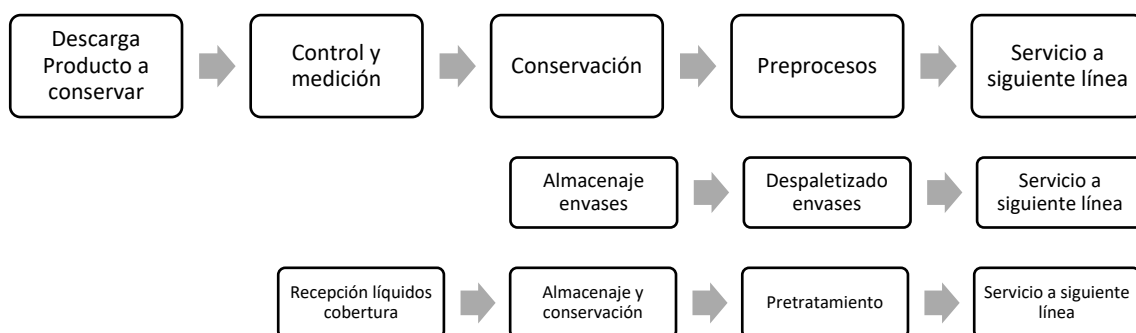


Figura 2.3. Esquema líneas de producción para tratamiento de insumos.

Y así con todas las etapas (y otras omitidas).

Para cada combinación posible de producto a conservar, líquidos de cobertura a utilizar con ese producto, formato del envase (latas metálicas de diferente forma y material, botes de cristal, envases plásticos de diferentes formas, etc.), procesos térmicos a aplicar (esterilización, pasteurización, tratamiento térmico semiconserva, ...), formato y disposición de paquetes, cajas y enfardados del producto final, ..., será necesaria una línea de producción a medida (Sánchez 2003). Esta línea a su vez se podrá dividir en una serie de más pequeñas líneas de producción específicas para cada etapa. El cambio en una o varias de estas líneas de producción componentes, serán las que permitan la flexibilidad en el tiempo de la línea de producción completa, permitiendo a la planta de producción variar y adaptar su producción al mercado.

Los responsables de estas industrias conserveras requieren pues de una agilidad en sus proveedores de maquinaria y sistemas suficiente para tomar rápidas decisiones en cuanto a los cambios que necesiten realizar en sus plantas. Estos cambios generalmente conllevan la sustitución o nueva instalación de parte de estas líneas de producción o incluso líneas completas. En un mercado tan competitivo y global como el actual, en cuanto a líneas de producción a medida, el fabricante-integrador de líneas de producción tiene la necesidad de dar una rápida y ágil respuesta a las peticiones de sus clientes. Este problema abarca todas las etapas de trabajo, desde los propios plazos y costes de fabricación hasta la entrega de la información, en forma de oferta, al cliente. Este último punto es el que toma importancia para el proyecto que aquí se trata.

2.3 EJEMPLOS DE LÍNEAS DE PRODUCCIÓN

Con ánimo de ampliar el contexto sobre líneas de producción en la industria conservera, se muestran a continuación dos ejemplos de líneas, una teórica, de carácter general, y otra real y, en cierta manera, cuasi-estandar en la producción de atún en conserva.

2.3.1 Línea general de conserva multiproducto:

En este caso se trata de una línea ejemplo de un catálogo, como se muestra en la Figura 2.4., con afán ilustrativo, que recoge las siguientes funcionalidades:

Despaletizado (1) de envases vacíos, pues llegan desde la fábrica de envases en palets, higienizado de los envases (2), usualmente con inyección de vapor, dosificador de sólidos (3), por ejemplo para el llenado de cada lata con el producto a envasar, como pueden ser verduras u otros productos en pequeños sólidos, conformado del producto en el envase (4), introducción de los líquidos de gobierno necesarios (5), como pueden ser agua, aceites, salsas, ..., cierre de envases (6), donde se le coloca la tapa a cada envase, llenado de cestos contenedores para esterilización (7), proceso de esterilización (8) o de pasteurización (9), mesas de trabajo para procesos manuales sobre el producto (10), vaciado de cestos de esterilización (11), comprobaciones (12), acumulación (buffer) para el funcionamiento de la línea (13), lavado y secado del exterior de los envases (14 y 15), colocación de etiquetas en envases (16), embalaje de los envases en cajas, fajas, bandejas, ... (17), cobertura plástica sobre el producto final (18) y paletizado para su almacenaje o envío a consumidor (19).

En este ejemplo se puede observar de manera clara, que en función de la selección de unas u otras máquinas de las mostradas permitirá reconfigurar la línea de producción para la realización de un producto final u otro. Podrían instalarse sólo ciertas máquinas y restringir la producción a un tipo de producto final, o instalar todas, con sus adaptaciones para los cambios de producción, y permitir fabricar un producto u otros según el ajuste sobre la línea.

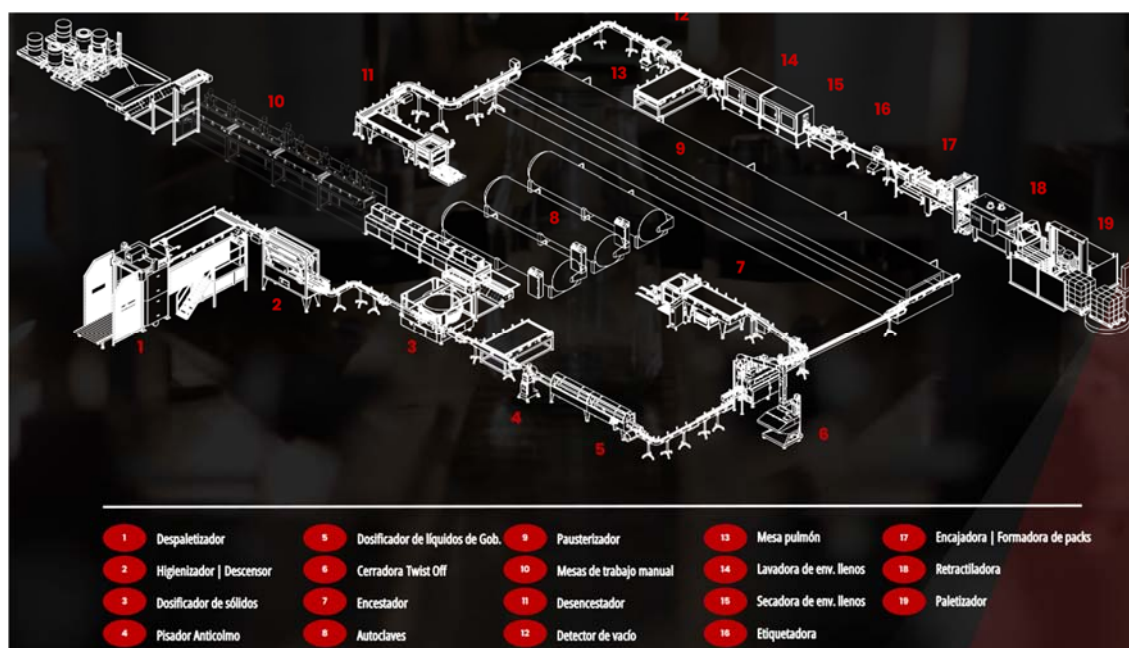


Figura 2.4. Línea multiproducto ejemplar (Emerito, 2024).

2.3.2 Línea de conserva de atún

En este caso se muestra un plano en planta con una línea de producción de atún en conserva en envases metálicos (latas) redondos de diferentes diámetros (véase Figura 2.5.). Abarca la parte productiva desde el empaque (introducir el atún ya cocido y limpio en las latas), hasta su envío al almacén de la planta industrial, ya convertido en conserva.

A modo resumen, los pasos dados (el trabajo de cada familia de máquinas incluida en la línea) serían: Introducir la cantidad adecuada de atún en cada envase. Adición de cada uno de los diferentes líquidos de gobierno (Aceites, salsas, salmueras, aditivos, ...) en cada envase. Cierre hermético de cada envase (colocación de tapa). Lavado externo de las latas. Llenado de cestos contenedores de envases para su introducción en autoclaves esterilizadores. Proceso térmico de esterilización en entorno controlado (podemos decir que es en este punto cuando se “convierte en conserva”). Vaciado de cestos de esterilización. Secado de cada envase y envío a la siguiente línea (almacén).

En este caso en particular se trata de una línea con una producción de 4000Kg/h, 150 kW, y un alto grado de optimización con sólo 4 operarios necesarios.

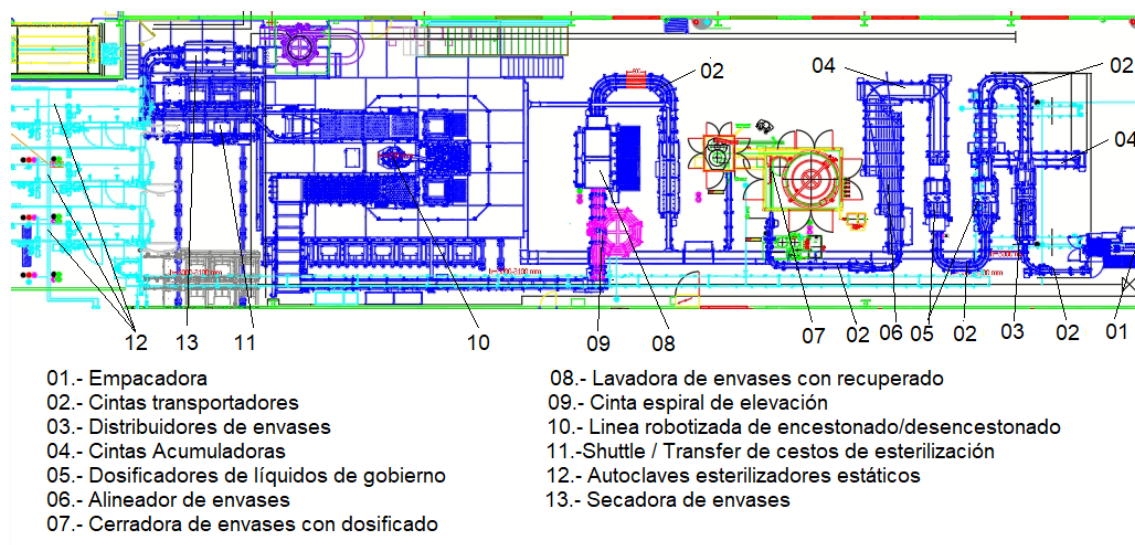


Figura 2.5. Línea de conserva de atún (tacore, 2024).

2.4 EJEMPLOS DE FAMILIA DE MÁQUINAS

Continuando con el contexto necesario, se muestran a continuación dos ejemplos de familias de máquinas en particular. En ellos se puede observar que para cada familia de máquinas existen diversos parámetros que, una vez elegidos, darán lugar a máquinas físicas distintas con sus propias características, capacidades y consumos.

2.4.1 Autoclaves esterilizadores estáticos

Se tratan de máquinas destinadas a la realización de la esterilización comercial del producto a conservar, mediante tratamiento térmico. Todas sus variantes comparten además la característica de ser físicamente horizontales, realizar el tratamiento del producto introducido en contenedores con forma cúbica y no presentar movimiento del producto durante el tratamiento. Por ejemplo: Autoclaves esterilizadores horizontales marca tacore (familia de máquinas) como se muestra en la Figura 2.6. Disponibles en diferentes medidas y capacidades, diferentes niveles de automatización y otras configuraciones a medida (máquina) como se muestra en la Figura 2.7. Dentro de esta familia de máquinas se podrá escoger la adecuada para la producción (capacidad de la máquina), consumo (tipo de solución adoptada para el aporte energético y el enfriamiento), operarios necesarios para su manejo (nivel de automatización de la introducción y extracción de producto, así como del sistema de control del proceso) requeridos por el resto de la línea de producción.



Figura 2.6. Autoclaves esterilizadores horizontales estáticos (tacore, 2024).

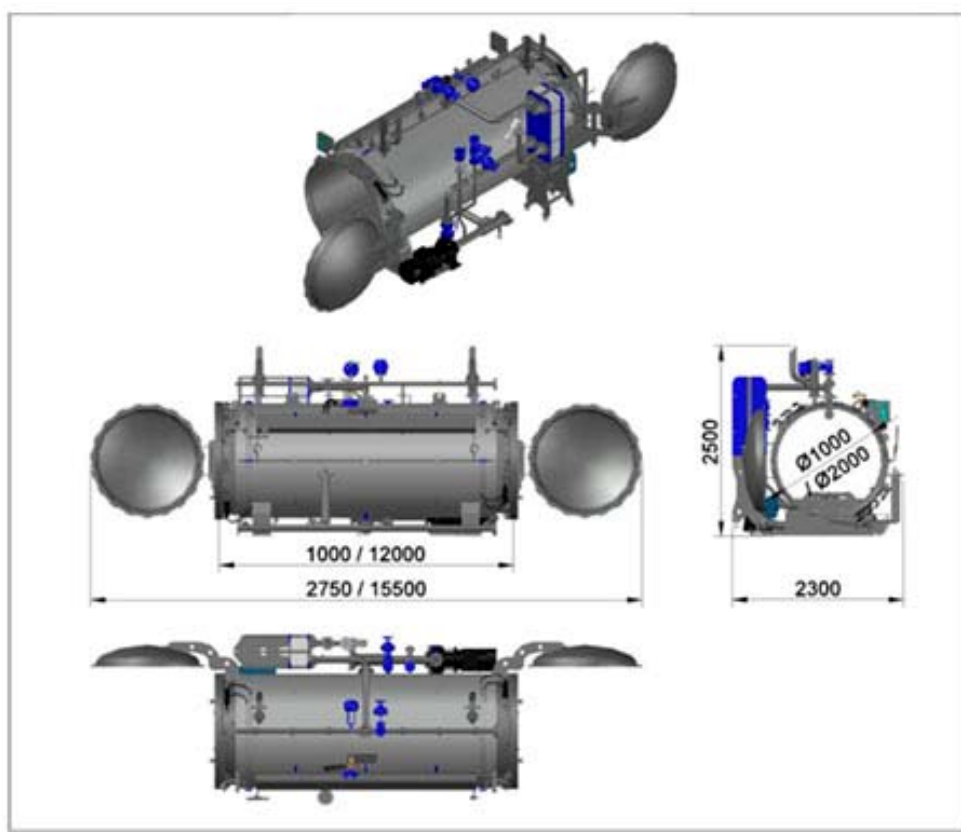


Figura 2.7. Diferentes medidas para escoger una máquina de una familia de máquinas autoclaves (tacore, 2024).

2.4.2 Encestadores/desencestadores

Según lo referido anteriormente, algunos tipos de procesos requieren de la introducción (y posterior extracción) del producto una vez envasado y cerrado, en contenedores (véase Figura 2.9.) para su posterior proceso. Para ello se disponen los encestadores/Desencestadores como el mostrado en la Figura 2.8: máquinas diseñadas para introducir y/o extraer los envases ya cerrados dentro de los contenedores utilizados habitualmente en autoclaves de tipo estáticos horizontales. Pueden ser automáticos, semiautomáticos o manuales, y pueden realizar sólo una de las funciones (encestar o desencestar) o ambas secuencialmente.



Figura 2.8. Encestado/desencestador de envases (Emerito, 2024).



Figura 2.9. Diferentes tipos de contenedores para esterilización.

2.5 SOLUCIONES EXISTENTES

Aunque en la mayoría de empresas este primer paso en el proceso de definición y diseño de una línea de producción se realiza de forma cuasi-manual y haciendo uso de la experiencia del técnico-comercial en cuestión, existen soluciones comerciales que pueden ayudar al proceso.

En general, las tres soluciones que se mostrarán en los siguientes apartados están compuestas por un mínimo de 4 módulos interrelacionados (simulador de planta/línea de producción, módulo de diseño en detalle, simulador de funcionamiento o revisión de diseño y finalmente uno o más módulos de simulación de costes, producción, gestión de almacén, ...). Estas soluciones aparecen en todo caso como complemento a las herramientas iniciales dedicadas al diseño en detalle de componentes y máquinas, inicialmente en 2D y posteriormente en 3D. Es por ello que requieren de un trabajo inicial en este apartado (ya sean diseño en detalle de máquinas existentes o diseño esquematizado de máquinas prototipos) con suficiente información para poder alimentar a los otros módulos.

Generalizando, la forma de trabajar con estas soluciones será:

- Por una parte, se trabajará con el módulo de diseño para crear máquinas, en detalle o prototipadas, que conformarán el catálogo de máquinas disponibles. También se podrán incorporar diseños de máquinas de terceros (compradas). A partir de este trabajo se podrá continuar finalizando los diseños y creando la documentación requerida (planos de fabricación y montaje) para cada máquina que sea requerida para fabricación.
- Por otra parte, utilizando el módulo de simulación de planta/línea de producción, se conformará el layout en 2D y/o 3D de la línea a ofertar sobre el plano de la planta del cliente, insertando esas representaciones de máquinas diseñadas en un dibujo CAD que se mostrará al cliente.
- Así mismo, utilizando el correspondiente módulo de costes, se generará el presupuesto correspondiente para esa selección de máquinas y su resultado, en formato documento, se convertirá, junto al dibujo CAD, en la oferta al cliente.

Las soluciones de este tipo más utilizadas son:

2.5.1 Autodesk Factory Design Utilities

Más información y manual de instrucciones en Autodesk Factory Design (Autodesk 2024).

Parte de la suite de Autodesk, sus funciones son ayudar en el diseño y optimización de plantas industriales. En la Figura 2.10. se muestra la interfaz de Factory Design durante la configuración de una línea robotizada genérica.



Figura 2.10. Factory Design en AutoDesk Inventor (Autodesk , 2024).

Características Básicas:

Permite crear diseños detallados en 2D y 3D de líneas de producción. Incluye un extenso catálogo de componentes estándar y específicos de la industria. Facilita la simulación del flujo de trabajo y la identificación de posibles cuellos de botella. Automatiza la creación de listas de materiales y estimaciones de costos. Además, permite la integración con el resto de software de Autodesk, tal como AutoCAD (muy utilizado en este sector para la representación de layouts 2D de las plantas de producción), Inventor (para el diseño en detalle de los componentes de la línea), NavisWorks (para la revisión del diseño antes de la construcción), etc.

Requerimientos del sistema:

Microsoft Windows 11 o 10.

CPU 2.5 GHz (3GHz recomendado, 4 o más núcleos).

8GB RAM (20 GB recomendado).

40 GB espacio en disco (160 GB recomendados).

Tarjeta gráfica 2GB con 29 GB/s y DirectX11 (8GB y 106GB/s recomendados).

Resolución gráfica 1280x1024 (4K recomendado).

2.5.2 Siemens Tecnomatix Plant Simulation

Parte de la suite Tecnomatix de Siemens, esta herramienta avanzada de simulación y modelado se utiliza para diseñar y optimizar sistemas de producción y logística (Siemens, 2024). En la Figura 2.11. se muestra la interfaz de Plant Simulation.

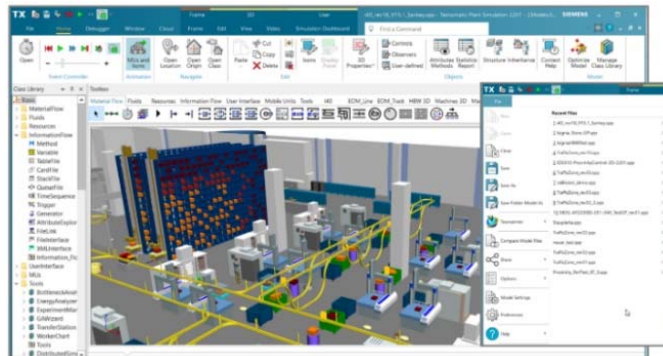


Figura 2.11. Plant Simulation de Siemens (Siemens, 2024)

Características Básicas:

Permite crear modelos tridimensionales detallados de líneas de producción. Analiza y optimiza el rendimiento de la producción, incluyendo el flujo de materiales y la utilización de recursos. Incluye una amplia gama de componentes específicos de la industria. Identifica y propone soluciones para mejorar la eficiencia de la línea de producción. Facilita la integración con sistemas de gestión de recursos empresariales (ERP) y de ejecución de fabricación (MES). Además, permite su integración con Siemens NX (para el diseño en detalle de cada componente), y con el resto de la suite de software de Siemens para, incluso, generar un gemelo digital de la línea.

Requerimientos del sistema:

Microsoft Windows 11.

CPU 2.5 GHz (3GHz recomendado, 4 o más núcleos).

4GB RAM (16 GB recomendado).

2 GB espacio en disco.

Tarjeta gráfica 2GB y OpenGL4.5 (8GB o más recomendados).

Resolución gráfica 1920x1080.

2.5.3 Dassault Systèmes DELMIA

Parte de la plataforma 3DEXPERIENCE de Dassault Systèmes, DELMIA se centra en la simulación y planificación de la fabricación. Para la solución aquí buscada, el componente de las soluciones de Dassault DELMIA a utilizar sería: Factory Layout Designer (Dassault, 2024), véase la Figura 2.12. donde se muestra su interfaz durante la configuración de una línea productiva.

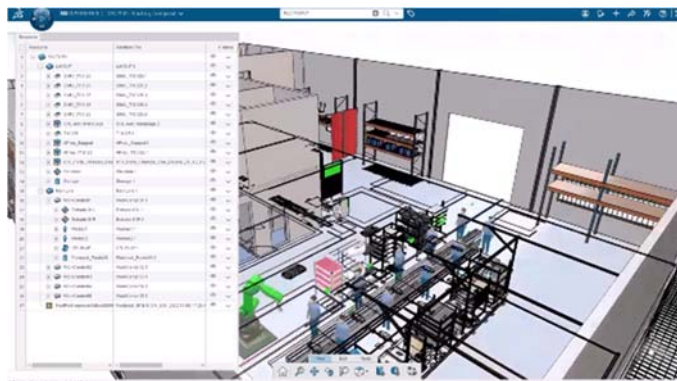


Figura 2.12. Factory Layout Designer de DELMIA (Dassault, 2024)

Características Básicas:

Crea modelos tridimensionales precisos de las líneas de producción. Permite reutilizar equipos industriales (máquinas y conjuntos de máquinas) almacenados en bibliotecas de la plataforma. Realiza análisis detallados del rendimiento de la producción para identificar áreas de mejora. Facilita la gestión de recursos y la optimización del uso de materiales y mano de obra. Además de otras, permite su integración con los softwares de diseño Catia y SolidWorks para el diseño en detalle de los componentes de la línea.

Requerimientos del sistema:

Microsoft Windows 10 o 11.

CPU 2.5 GHz (3GHz recomendado, 4 o más núcleos).

8GB RAM (32GB recomendado).

50 GB espacio en disco (256GB recomendados).

Tarjeta gráfica 1GB y OpenGL4.5 (5GB y 140GB/s o más recomendados).

Resolución gráfica 1920x1080 (2560x1440 recomendado).

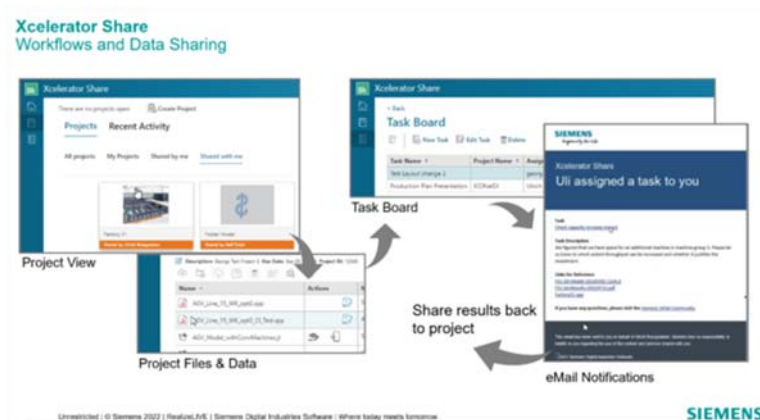


Figura 2.14. Flujo de trabajo y comunicación de información en Siemens Tecnomatix.

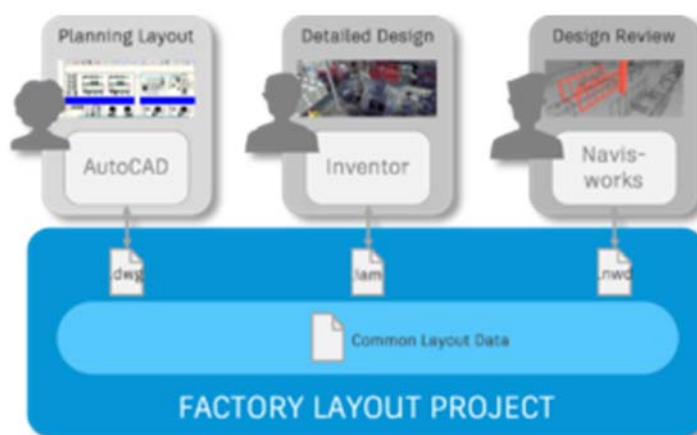


Figura 2.13. Interconexión de diferentes etapas del proyecto completo en Autodesk Factory Utilities

2.5.4 Conclusiones sobre las soluciones existentes

El principal problema del uso de estas soluciones para el problema aquí presentado, es que van más allá del primer paso comentado. Aglutinan todos los pasos del proceso de venta – diseño – fabricación de una línea de producción y los interconectan para un mejor control de todo el proyecto. Como se muestra en las Figura 2.13. y 2.14., involucran diversos departamentos de la empresa que lo utiliza para facilitar la información requerida y para servir los resultados. Debido a esto, su precio y sus requerimientos del sistema donde serán ejecutados son altos. Se trata de soluciones muy potentes que requieren, además de una curva de aprendizaje importante y de conocimientos técnicos de diseño, un tiempo no despreciable para obtener un mínimo de información suficiente para dar una primera respuesta al cliente, y eso, exactamente, es lo que quiere evitar la aplicación aquí desarrollada. Estas herramientas abordan el problema completo para el desarrollo de una nueva línea de producción, pero no aportan ventajas a la resolución de la primera etapa

de un proyecto de este tipo: ponerse de acuerdo con el cliente para ofertarle lo que quiere, en un tiempo mínimo, con un mínimo de información.

2.6 CONCLUSIÓN

Como se ha intentado mostrar, las líneas de producción de la industria conservera son elementos complejos, con multitud de variables que, además, cambian con el tiempo.

La primera respuesta a un cliente para una nueva línea de producción por parte de un fabricante-integrador debe contener un mínimo de información suficiente para su toma de decisiones y ofrecerse en un tiempo mínimo. La generación de estas respuestas debe ser flexible y ágil para permitir ofrecer diversas variaciones de la solicitud del cliente, y no deben aumentar el tiempo de respuesta.

Las soluciones comerciales existentes trabajan en un nivel de abstracción inferior, con mucha más necesidad de detalle, lo que impide una pronta respuesta al ser necesaria mucha más información, tiempo y demás recursos, para generar una primera respuesta.

Capítulo 3

Análisis

3.1 INTRODUCCIÓN

En el presente capítulo se realizará un estudio del problema y sus requisitos, es decir, el análisis del problema. Siguiendo la terminología y recomendaciones mostradas en (Larman, 2002), se tratará por una parte el análisis de requisitos (estudio de los requisitos) y por otra el análisis de objetos (estudio de los objetos del dominio).

Para la primera parte se descubrirán los requisitos del sistema relacionados con los casos de uso, después de haber definido el límite del sistema y los actores principales y sus objetivos. Finalmente se creará un diagrama de casos de uso para ilustrarla.

Para la segunda parte se creará una descripción del dominio, identificando los conceptos, atributos y asociaciones significativas, dando lugar a un modelo del dominio. Para ello se utilizarán los casos de uso creados en el punto anterior como la fuente a partir de la cual extraer este análisis (Larman, 2002, p.43) y (Pressman y Maxim, 2021, p.131).

3.2 LÍMITES DEL SISTEMA, ACTORES Y SUS OBJETIVOS

La definición de los límites del sistema para esta herramienta es sencilla, puesto que la propia aplicación “DraftLine” es el sistema a estudiar y todo lo externo a ella está fuera de los límites. Por lo tanto, quedarían fuera solamente los actores (por definición externos). En este caso se puede definir al único actor principal como el **usuario de la aplicación**, y dos actores de apoyo correspondientes a la comunicación con el sistema operativo para escribir y leer ficheros y para hacer uso de las impresoras del sistema: **sistema de archivos del sistema operativo** y **sistema de impresión del sistema operativo**.

Dadas las necesidades de la herramienta, se toman los objetivos del actor principal según la siguiente tabla:

Actor	Objetivo	
Usuario de la aplicación	• Crear nuevo catálogo de máquinas. • Exportar e importar catálogo. • Guardar cambios en catálogo actual. • Crear nueva familia de máquinas. • Crear nueva máquina. • Editar familia o máquinas del catálogo actual. • Crear nueva línea.	• Exportar línea. • Importar línea. • Editar datos línea. • Editar esquema de familias de máquinas de línea. • Editar condiciones de dimensionamiento. • Imprimir Informe de línea dimensionada. • Ver manual de usuario.

Tabla 3.1. Objetivos del actor principal

3.3 CASOS DE USO

A continuación, se recogen los casos de uso extraídos de los objetivos del actor principal, siguiendo el formato de dos columnas presentado en (Wirfs-Brock, 1993), añadiendo la numeración de pasos del escenario de éxito y de las extensiones, así como otros detalles, según el formato usecases.org, de (Lamar 2002). Además, se muestran subrayadas las entidades que se utilizarán posteriormente como fuente para la creación del modelo de dominio del problema (Ver apartado 3.6).

Caso de uso UC1: Crear nuevo catálogo de Máquinas	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado y se encuentra en la pantalla inicial por defecto.	
Postcondiciones (garantías de éxito): Hay un catálogo vacío, listo para incluir nuevas Familias de máquinas, con sus Máquinas.	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona <u>modo Editar catálogo</u>	2. Muestra la pantalla con las utilidades del <u>catálogo</u>
3. Selecciona Crear nuevo <u>catálogo</u>	4. Informa al usuario que se eliminarán todas las máquinas existentes en el catálogo actual y pide confirmación
5. Confirma	6. Se eliminan todas las máquinas existentes y se muestra un catálogo vacío.
Excepciones	
5b. El usuario cancela (no confirma)	7. La aplicación acepta y vuelve a 2.

Tabla 3.2. Caso de uso UC1: Crear nuevo catálogo de Máquinas

Caso de uso UC2: Exportar catálogo	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado y se encuentra en la pantalla inicial por defecto. Hay un catálogo actual cargado.	
Postcondiciones (garantías de éxito): Se generó un fichero conteniendo el catálogo actual, en un directorio del sistema	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Editar catálogo	2. Muestra la pantalla con las utilidades del catálogo
3. Selecciona Exportar catálogo	4. Muestra una ventana para permitir al usuario escoger nombre y ruta para el fichero a generar
5. El usuario introduce los datos	6. Comprueba que puede escribir el nuevo fichero, con el nombre indicado en el directorio indicado y escribe en él los datos del catálogo actual. Finalmente vuelve a 2.
Excepciones	
5b. El usuario cancela	7. La aplicación acepta y vuelve a 2.
5c. Introduce datos no válidos	8. Muestra el error en los datos introducidos y vuelve a 2.

Tabla 3.3. Caso de uso UC2: Exportar catálogo

Caso de uso UC3: Importar catálogo	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado y se encuentra en la pantalla inicial por defecto.	
Postcondiciones (garantías de éxito): Se cargaron los datos de un catálogo existente en un fichero, en el catálogo actual cargado en la aplicación.	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Editar catálogo	2. Muestra la pantalla con las utilidades del catálogo
3. Selecciona Importar catálogo	4. Informa al usuario que se eliminarán todas las máquinas existentes en el catálogo actual y pide confirmación
5. Confirma	6. Muestra una ventana para permitir al usuario seleccionar el fichero del que tomar los datos.
7. El usuario selecciona el fichero	8. Comprueba que el fichero existe y los datos son válidos. Elimina las máquinas existentes en el catálogo actual y carga las que se encuentran en el fichero seleccionado. Finalmente vuelve a 2
Excepciones	
5b. El usuario cancela	7. La aplicación acepta y vuelve a 2.
7b. El usuario cancela	8. La aplicación acepta y vuelve a 2.
7c. Introduce datos no válidos	9. Muestra el error en los datos introducidos y vuelve a 2.

Tabla 3.4. Caso de uso UC3: Importar catálogo

Caso de uso UC4: Guardar catálogo	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado y se encuentra en la pantalla inicial por defecto. Hay un catálogo actual cargado.	
Postcondiciones (garantías de éxito): Se guardaron los datos del catálogo actualmente cargado en la aplicación en el fichero predeterminado de la aplicación.	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Editar catálogo	2. Muestra la pantalla con las utilidades del catálogo
3. Selecciona Guardar catálogo	4. Consulta si se quieren guardar los cambios en el catálogo actual y pide confirmación
5. Confirma	6. Escribe en el fichero predeterminado los datos del catálogo actual. Finalmente vuelve a 2.
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2.

Tabla 3.5. Caso de uso UC4: Guardar catálogo

Caso de uso UC5: Crear nueva Familia de máquinas	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado y se encuentra en la pantalla inicial por defecto. Hay un catálogo actual cargado.	
Postcondiciones (garantías de éxito): Se guardaron los datos de una nueva Familia de máquinas en el catálogo actualmente cargado en la aplicación	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo <u>Editar catálogo</u>	2. Muestra la pantalla con las utilidades del <u>catálogo</u>
3. Selecciona Nuevo <u>Familia de máquinas</u>	4. Muestra una ventana con los campos de información a editar
5. Introduce los datos y acepta	6. Crea la nueva <u>Familia de máquinas</u> con los datos introducidos, la añade al catálogo y vuelve a 2
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2.
5c. Introduce datos no válidos.	8. Informa de los datos erróneos y vuelve a 4.

Tabla 3.6. Caso de uso UC5: Crear nueva Familia de máquinas

Caso de uso UC6: Crear nueva Máquina	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado y se encuentra en la pantalla inicial por defecto. Hay un catálogo actual cargado con, al menos, una Familia de máquinas existente.	
Postcondiciones (garantías de éxito): Se guardaron los datos de una nueva Máquina en el catálogo actualmente cargado en la aplicación	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Editar catálogo	2. Muestra la pantalla con las utilidades del catálogo
3. Selecciona una de las Familias de máquinas existentes en el catálogo y solicita crear Nueva Máquina	4. Muestra una ventana con los campos de información a editar
5. Introduce los datos y acepta	6. Crea la nueva Máquina en la Familia de máquinas seleccionada, con los datos introducidos, la añade al catálogo, vuelve a 2
Excepciones	
3b. El usuario solicita crear Nueva Máquina sin haber seleccionado un Familia de máquinas.	7. Informa que se debe seleccionar una Familia de máquinas y vuelve a 2.
5b. El usuario cancela.	8. La aplicación acepta y vuelve a 2
5c. Introduce datos no válidos.	9. Informa de los datos erróneos y vuelve a 4.

Tabla 3.7. Caso de uso UC6: Crear nueva Máquina

Caso de uso UC7: Editar Familias o Máquinas del catálogo actual	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado y se encuentra en la pantalla inicial por defecto. Hay un catálogo actual cargado con, al menos, una Familia de máquinas existente.	
Postcondiciones (garantías de éxito): Se guardaron los nuevos datos de una Familia o máquina en el catálogo actualmente cargado en la aplicación	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Editar catálogo	2. Muestra la pantalla con las utilidades del catálogo
3. Selecciona una de los Familias de máquinas o Máquinas existentes en el catálogo y solicita editar sus datos	4. Muestra una ventana con los campos de información a editar
5. Edita los datos y acepta	6. Guarda los cambios en la familia o máquina editada y vuelve a 2
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2
5c. Introduce datos no válidos.	8. Informa de los datos erróneos y vuelve a 4.

Tabla 3.8. Caso de uso UC7: Editar Familias o Máquinas del catálogo actual

Caso de uso UC8: Crear nueva Línea	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado.	
Postcondiciones (garantías de éxito): Se generó una nueva línea con los datos por defecto, sin esquema de selección de Familias de máquinas. Se muestra un Informe de la línea generada.	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Dimensionar Línea	2. Muestra la pantalla con las utilidades del dimensionamiento de líneas
3. Selecciona crear una nueva Línea	4. Informa al usuario que se perderán los datos de cualquier línea actualmente cargada. Pide confirmación.
5. Confirma	6. Crea una Línea con los valores por defecto.
	7. Vuelve a 2, mostrando el informe de la línea en su estado actual.
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2
5c. Introduce datos no válidos.	8. Informa de los datos erróneos y vuelve a 4.

Tabla 3.9. Caso de uso UC9: Crear nueva Línea

Caso de uso UC9: Exportar Línea	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado. Hay una Línea dimensionada cargada en la aplicación	
Postcondiciones (garantías de éxito): Se generó un fichero conteniendo la línea dimensionada, en un directorio del sistema	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Dimensionar Línea	2. Muestra la pantalla con las utilidades del dimensionamiento de líneas
3. Selecciona Exportar Línea	4. Muestra una ventana para permitir al usuario escoger nombre y ruta para el fichero a generar
5. El usuario introduce los datos	6. Comprueba que puede escribir el nuevo fichero, con el nombre indicado en el directorio indicado y escribe en él los datos de la línea actual. Finalmente vuelve a 2.
Excepciones	
5b. El usuario cancela	7. La aplicación acepta y vuelve a 2.
5c. Introduce datos no válidos	8. Muestra el error en los datos introducidos y vuelve a 2.

Tabla 3.10. Caso de uso UC9: Exportar Línea

Caso de uso UC10: Importar Línea	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado.	
Postcondiciones (garantías de éxito): Se cargaron los datos de una línea existente en un fichero, en la aplicación.	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Dimensionar Línea	2. Muestra la pantalla con las utilidades del dimensionamiento de líneas
3. Selecciona Importar Línea	4. Informa al usuario que se perderán los datos de cualquier línea actualmente cargada. Pide confirmación
5. Confirma	6. Muestra una ventana para permitir al usuario seleccionar el fichero del que tomar los datos.
7. El usuario selecciona el fichero	8. Comprueba que el fichero existe y los datos son válidos. Comprueba si el catálogo actual contiene las máquinas necesarias para cargar la línea importada. En caso afirmativo pasa a 11. Si no es así, consulta al usuario si desea cargarlas desde la Línea importada en el catálogo actual.
9. El usuario confirma	10. Toma los datos de las máquinas existentes en la línea a importar y los añade en el catálogo actual cargado en la aplicación.
	11. Carga la línea importada desde el fichero indicado a la aplicación.
Excepciones	
5b. El usuario cancela	12. La aplicación acepta y vuelve a 2.
7b. El usuario cancela	13. La aplicación acepta y vuelve a 2.
7c. Introduce datos no válidos	14. Muestra el error en los datos introducidos y vuelve a 2.
9b. El usuario cancela	15. La aplicación acepta y vuelve a 2.

Tabla 3.11. Caso de uso UC10: Importar Línea

Caso de uso UC11: Editar Datos Línea	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado. Hay una Línea cargada en la aplicación.	
Postcondiciones (garantías de éxito): La Línea cargada muestra los nuevos datos de Línea	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Dimensionar Línea	2. Muestra la pantalla con las utilidades del dimensionamiento de líneas
3. Selecciona Editar Datos Línea	4. Muestra una ventana con los campos de información a editar
5. Edita los datos y acepta	6. Actualiza los datos de la línea y vuelve a 2, mostrando el informe de la línea en su estado actual.
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2
5c. Introduce datos no válidos.	8. Informa de los datos erróneos y vuelve a 4.

Tabla 3.12. Caso de uso UC11: Editar Datos Línea

Caso de uso UC12: Editar esquema de Familias de máquinas de la Línea	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado. Hay una Línea cargada en la aplicación.	
Postcondiciones (garantías de éxito): La línea cargada muestra el esquema de Familias de máquinas seleccionados para ella.	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Dimensionar Línea	2. Muestra la pantalla con las utilidades del dimensionamiento de líneas
3. Selecciona Editar esquema de Familias de máquinas	4. Muestra una ventana con los Familias de máquinas disponibles en el catálogo actual y la interfaz para seleccionarlos y ordenarlos para definirlos como esquema de la línea a dimensionar. Si la línea tenía anteriormente un esquema asociado, lo muestra para editar.
5. Realiza la selección y posición de los Familias de máquinas y acepta	6. Actualiza el esquema de Familias de máquinas de la Línea y vuelve a 2, mostrando el informe de la línea en su estado actual.
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2
5c. No selecciona un Familia de máquinas al intentar añadir al esquema	8. Informa de que se debe seleccionar un Familia de máquinas y vuelve a 4.

Tabla 3.13. Caso de uso UC12: Editar esquema de Familias de máquinas de la Línea

Caso de uso UC13: Editar condiciones de dimensionamiento	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado. Hay una Línea cargada en la aplicación	
Postcondiciones (garantías de éxito): La línea cargada muestra las condiciones de dimensionamiento seleccionadas, ha realizado los cálculos para la selección de Máquinas y muestra la selección realizada.	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Dimensionar Línea	2. Muestra la pantalla con las utilidades del dimensionamiento de líneas
3. Selecciona Editar condiciones de dimensionamiento	4. Muestra una ventana con los campos de datos a introducir por el usuario y las opciones objetivo del dimensionamiento.
5. Introduce los datos y selecciona los objetivos de dimensionamiento	6. Guarda los datos en la Línea y realiza el dimensionamiento, seleccionando las Máquinas, pertenecientes a las Familias de máquinas del esquema de la Línea, que cumplan con los objetivos de dimensionamiento. Guarda esa selección en la Línea y vuelve a 2, mostrando el informe de la línea en su estado actual.
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2
5c. Introduce datos no válidos.	8. Informa de los datos erróneos y vuelve a 4.

Tabla 3.14. Caso de uso UC13: Editar condiciones de dimensionamiento

Caso de uso UC14: Imprimir Informe de Línea Dimensionada	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado. Hay una Línea cargada en la aplicación y su informe generado.	
Postcondiciones (garantías de éxito): Se ha impreso el informe de la línea	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el modo Dimensionar Línea	2. Muestra la pantalla con las utilidades del dimensionamiento de líneas
3. Selecciona Imprimir informe	4. Muestra la ventana de configuración de impresión.
5. Selecciona la impresora a utilizar y demás configuraciones de impresión y acepta.	6. Toma el Informe de la línea actual y lo envía al gestor de impresión del Sistema Operativo junto con la configuración de impresión seleccionada. Finalmente vuelve a 2.
Excepciones	
5b. El usuario cancela.	7. La aplicación acepta y vuelve a 2
5c. Introduce datos no válidos.	8. Informa de los datos erróneos y vuelve a 4.

Tabla 3.15. Caso de uso UC14: Imprimir informe de Línea dimensionada

Caso de uso UC15: Ver Manual de Usuario	
Actor: Usuario de la aplicación	
Precondiciones: La aplicación se encuentra instalada en el sistema y el usuario ya la ha iniciado.	
Postcondiciones (garantías de éxito): Se ha cargado el manual del usuario en el visor de la aplicación para su consulta por el usuario	
Escenario principal de éxito	
Acción del Actor	Respuesta del Sistema
1. Selecciona el <u>modo Ayuda</u>	2. Muestra la pantalla con el visor donde carga el <u>Manual de Usuario</u>

Tabla 3.16: Caso de uso UC15: Ver Manual de usuario

3.4 REQUISITOS

Se recogen a continuación los requisitos con los cuales debe ser conforme el sistema, tomando como principal fuente de información los casos de uso del apartado anterior, así como de las necesidades del proyecto.

Siguiendo las recomendaciones de (Larman 2002, p.40), serán aquí clasificados según el modelo FURPS+ (Grady 1992).

En cuanto a la numeración, se utilizará la mostrada en (Gómez y Moraleda, 2014).

3.4.1 Requisitos funcionales

- **R.1.1.-** La aplicación permitirá crear nuevos catálogos de familias de máquinas.
- **R.1.2.-** La aplicación permitirá exportar el catálogo de familias de máquinas actual a un fichero de datos en un directorio del sistema indicado por el usuario.
- **R.1.3.-** La aplicación permitirá importar un catálogo de familias de máquinas, desde un fichero de datos en un directorio del sistema indicado por el usuario, al catálogo actual de la aplicación.
- **R.1.4.-** La aplicación permitirá guardar los datos del catálogo actual en un fichero propio, para guardar los cambios que el usuario realice sobre el catálogo actual.
- **R.1.5.-** La aplicación permitirá crear nuevas familias de máquinas y guardarlas en el catálogo actual.
- **R.1.6.-** La aplicación permitirá crear nuevas máquinas, pertenecientes a alguna familia de máquinas, y guardarlas en el catálogo actual.
- **R.1.7.-** Las familias de máquinas y las propias máquinas existentes en el catálogo podrán ser editados para cambiar los valores de sus datos y características.
- **R.1.8.-** La aplicación permitirá crear una nueva línea a dimensionar, cargando en la propia aplicación una línea con los valores por defecto y generando el informe sobre la misma.
- **R.1.9.-** La aplicación permitirá exportar la línea que se encuentre cargada en la aplicación, a un fichero de datos en un directorio del sistema indicado por el usuario.
- **R.1.10.-** La aplicación permitirá importar una línea, previamente generada por la propia aplicación, desde un fichero de datos, en un directorio del sistema, indicado por el usuario. Si el catálogo de la aplicación ya no contiene las máquinas utilizadas en la línea a importar, deberá consultar al usuario para actualizar el catálogo con éstas o cancelar la importación. De la misma manera, deberá generar el informe sobre la línea cargada.
- **R.1.11.-** Los datos informativos de la línea cargada en la aplicación se podrán editar, para corregir o introducir nueva información que se verá en el informe de la línea.
- **R.1.12.-** La selección de familias de máquinas para la línea, en forma de esquema, se podrá editar para añadir, eliminar o reordenar los modelos de dicho esquema, de una manera clara y simple. De la misma manera, deberá actualizar el informe sobre la línea cargada.
- **R.1.13.-** Las condiciones de dimensionamiento de la línea se podrán añadir a su información e informar a la aplicación cómo debe realizar el dimensionamiento de la misma. De la misma manera, deberá actualizar el informe sobre la línea cargada.

- **R.1.14.-** La aplicación permitirá imprimir el informe de la línea cargada (que mantendrá en todo momento actualizado) en una de las impresoras del sistema, según selección y configuración de impresión por parte del usuario.
- **R.1.15.-** La aplicación dispondrá de un visor para que el usuario pueda consultar el manual de usuario de la propia aplicación.

3.4.2 Facilidad de uso

- **R.2.1.-** La aplicación dispondrá de una interfaz de usuario clara, simple y familiar, facilitando la realización de los casos de uso de la aplicación.
- **R.2.2.-** Se mostrarán ayudas al usuario para el entendimiento de cada funcionalidad de la aplicación.
- **R.2.3.-** La aplicación permitirá consultar su manual de usuario en cualquier momento, sin perder el trabajo realizado hasta el momento.

3.4.3 Fiabilidad

- **R.3.1.-** Para disponer de respaldo de los datos generados hasta el momento, la aplicación permitirá realizar exportaciones tanto del catálogo de familia de máquinas creado, como de la línea dimensionada.

3.4.4 Rendimiento

- **R.4.1.-** Los cálculos para el dimensionamiento de la línea deben generarse rápidamente, sin un tiempo de espera significativo.

3.4.5 Soporte

- **R.5.1.-** La aplicación permitirá volver a un estado inicial fácilmente, permitiendo descartar los cambios realizados en el catálogo o los datos introducidos en la línea a dimensionar.
- **R.5.2.-** La aplicación dispondrá de un manual de usuario a disposición del propio usuario en cualquier momento de su uso.
- **R.5.3.-** Se facilitarán medios de contacto para la asistencia técnica que permitan al usuario resolver problemas y obtener respuestas sobre el uso de la aplicación.

3.4.6 Requisitos adicionales de implementación

- **R.6.1.-** La aplicación deberá poder ejecutarse en las distintas versiones del Sistema Operativo Microsoft Windows usuales a fecha actual (Microsoft Windows 11, 10, 8.1, 7 SP1) (Versiones que todavía disponen de servicio oficial en la actualidad – septiembre 2024-).
- **R.6.2.-** La aplicación no necesitará más recursos hardware que los mínimos para ejecutar el sistema operativo en el que se ejecute.
- **R.6.3.-** El lenguaje para la realización de la aplicación será Visual Basic.Net, permitiendo una posible integración sencilla en un paquete de software de gestión en este lenguaje en la empresa en la que se realizarán las pruebas.

3.4.7 Requisitos adicionales de operaciones

- **R.7.1.-** La instalación de la aplicación se deberá realizar mediante instalador guiado (tipo Wizard) y no necesitará más que el permiso del administrador para su instalación en el sistema del usuario.

3.4.8 Requisitos adicionales de empaquetamiento

- **R.8.1.-** La distribución de la aplicación se realizará mediante un archivo instalador ejecutable que incluya la aplicación en sí y el Manual de Usuario necesario. Cualquier otro fichero de datos necesario deberá ser creado por la propia aplicación en su primera ejecución (por ejemplo, el fichero de datos para el guardado del catálogo actual de la aplicación).

3.5 DIAGRAMA DE CASOS DE USO

Con la información de los apartados anteriores, se puede mostrar a continuación un diagrama de casos de uso de la aplicación, que sirva para ilustrar los límites de la aplicación, los actores involucrados y los diferentes casos de uso. Generado con la herramienta Astah UML (Astah, 2024).

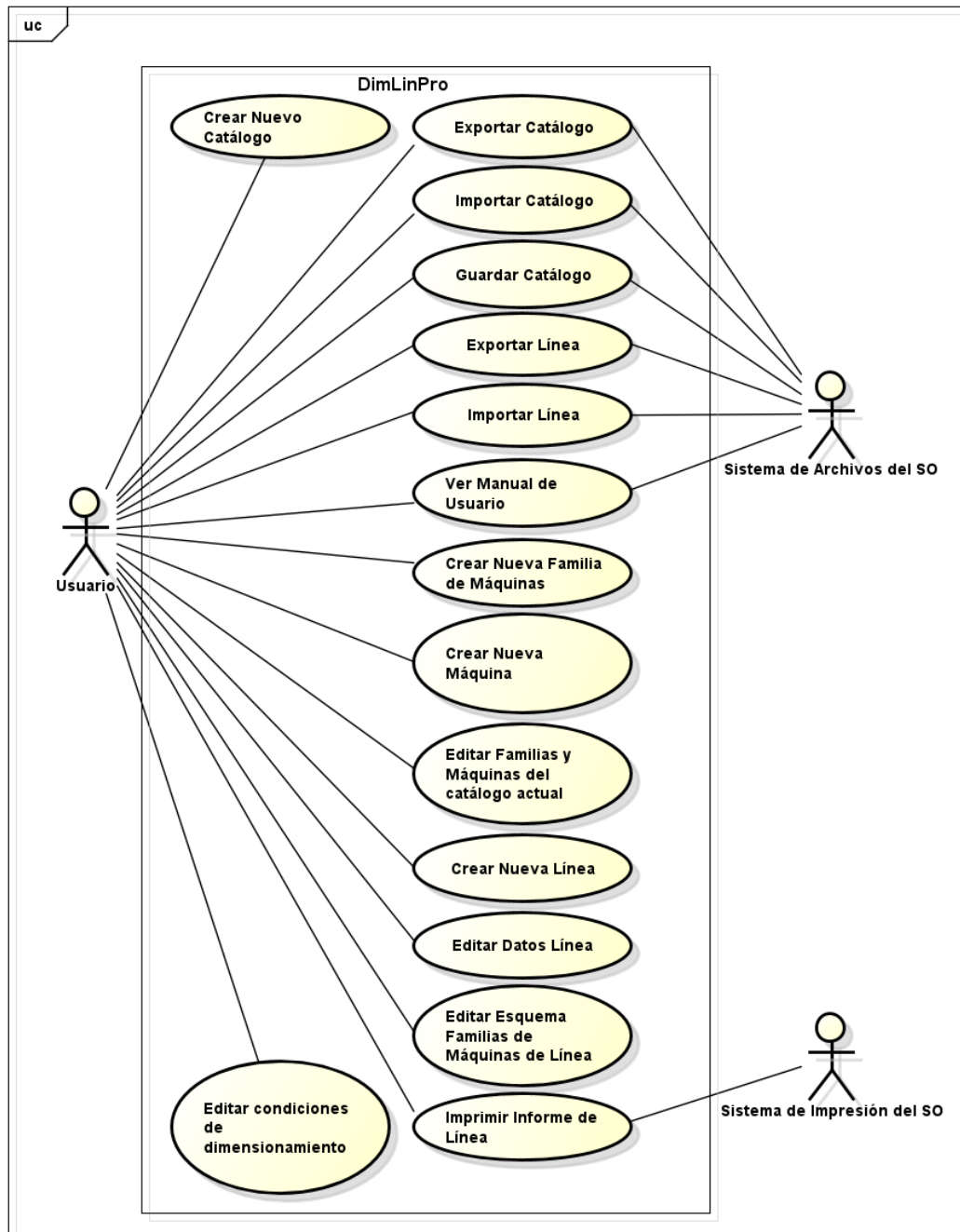


Figura 3.1. Diagrama de casos de uso.

3.6 DIAGRAMA DE MODELO DEL DOMINIO

Se crea a continuación un modelo de dominio que servirá de base para la selección de los objetos software que compondrán la aplicación. Para descubrir los elementos conceptuales que compondrán el modelo de dominio se aplica la identificación de frases nominales en los casos de uso, según (Larman 2002, p.128). Ver en el apartado 3.3 los elementos subrayados que se han utilizado como fuente para esta selección.

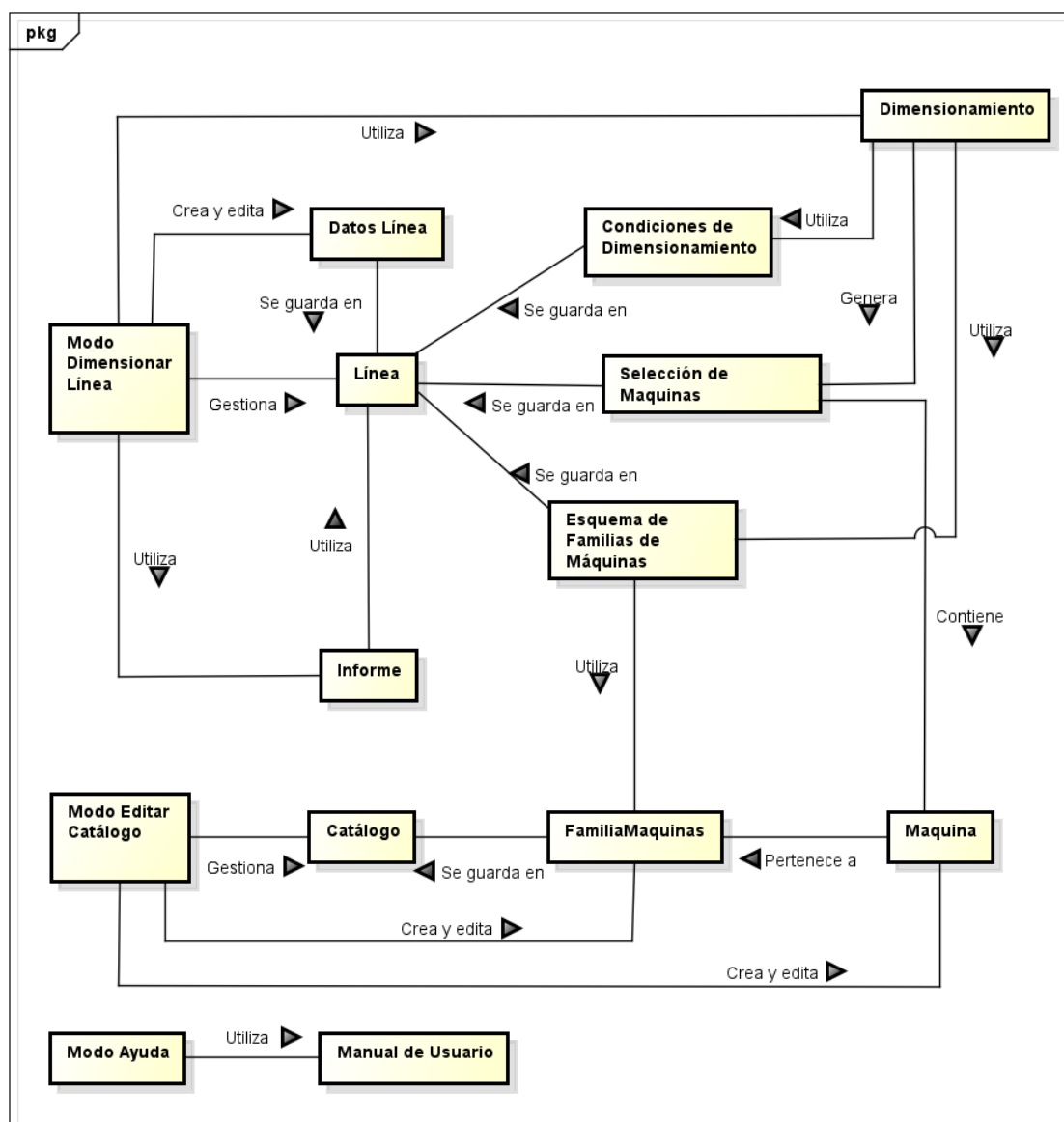


Figura 3.2. Diagrama de Modelo de Dominio.

3.7 CONCLUSIONES

Para el análisis del problema, se han seguido secuencialmente los pasos que conforman los apartados anteriores:

- Se han definido los **límites** de la herramienta, sus **actores** y los **objetivos** del actor principal.
- Se han identificado y recopilado los **casos de uso** de la herramienta.
- Basándose en el modelo FURPS+, se han listado los **requisitos** descubiertos, tanto en los casos de uso (mayormente los requisitos funcionales) como en la descripción y contexto del problema (el resto de tipo de requisitos).
- También se ha generado un **diagrama de casos de uso**, a partir de los casos de uso identificados, para ilustrar el problema.
- Finalmente, por identificación de las frases nominales en los casos de uso, se ha realizado un **diagrama de modelo del dominio** del problema con los elementos conceptuales que tendrán participación en el problema, y que servirán de inspiración para la organización en objetos de software de la aplicación.

Con esta información se procederá al diseño de la aplicación que dé solución al problema planteado.

Capítulo 4

Diseño

4.1 INTRODUCCIÓN

Parafraseando a (Gómez y Moraleda, 2014, p.113) “Después de haber especificado QUÉ se quiere resolver (durante el análisis), en el diseño se comenzará a determinar CÓMO se debe resolver el proyecto, seguido de la codificación y de las etapas de pruebas.”

Manteniendo de guía las recomendaciones de (Lamar 2002), y, ya que el lenguaje seleccionado para la realización del proyecto (Visual Basic.Net) lo permite, se orientará el diseño hacia el paradigma orientado a objetos. Se crearán los **diagramas de interacción**, que representan el modo en el que los objetos colaboran para satisfacer los requisitos, así como los **diagramas de clases** que resumen la definición de las clases software que se van a implementar en la aplicación. Los diagramas de interacción pueden realizarse mediante diagramas de secuencia y/o diagramas de colaboración. Aunque estos últimos permiten una visualización en páginas físicas más sencilla, se han escogido los **diagramas de secuencia** para la representación de la interacción en este proyecto, por mostrar más claramente la secuencia de mensajes entre objetos. Además, para simplificar la vista, se presentará un diagrama de clases simplificado (sin atributos, ni métodos o funciones) para mostrar la arquitectura del sistema. A continuación, se dará una pequeña explicación de cada clase por separado. Estos diagramas formarán parte del **modelo de diseño**. Una información más detallada de cada clase, sus atributos y métodos, puede encontrarse en el [anexo C “Documentación del código”](#).

Por último, se estudia cómo se ha dado respuesta a cada requisito de los encontrados durante el análisis y se muestra la relación entre los requisitos y los componentes del proyecto que les dan respuesta en una **matriz requisitos-componentes** detallando así la cobertura de requisitos aplicada, y se recogen las conclusiones extraídas del proceso del diseño en el último apartado de este capítulo.

4.2 MODELO DE DISEÑO

4.2.1 Diagramas de secuencia

4.2.1.1 Diagrama de secuencia del inicio de la aplicación

Para ilustrar la iniciación de la aplicación, y tomando licencias en la simplificación de la llamada desde el SO, se muestra a continuación el diagrama de secuencia al iniciar el formulario principal de la aplicación.

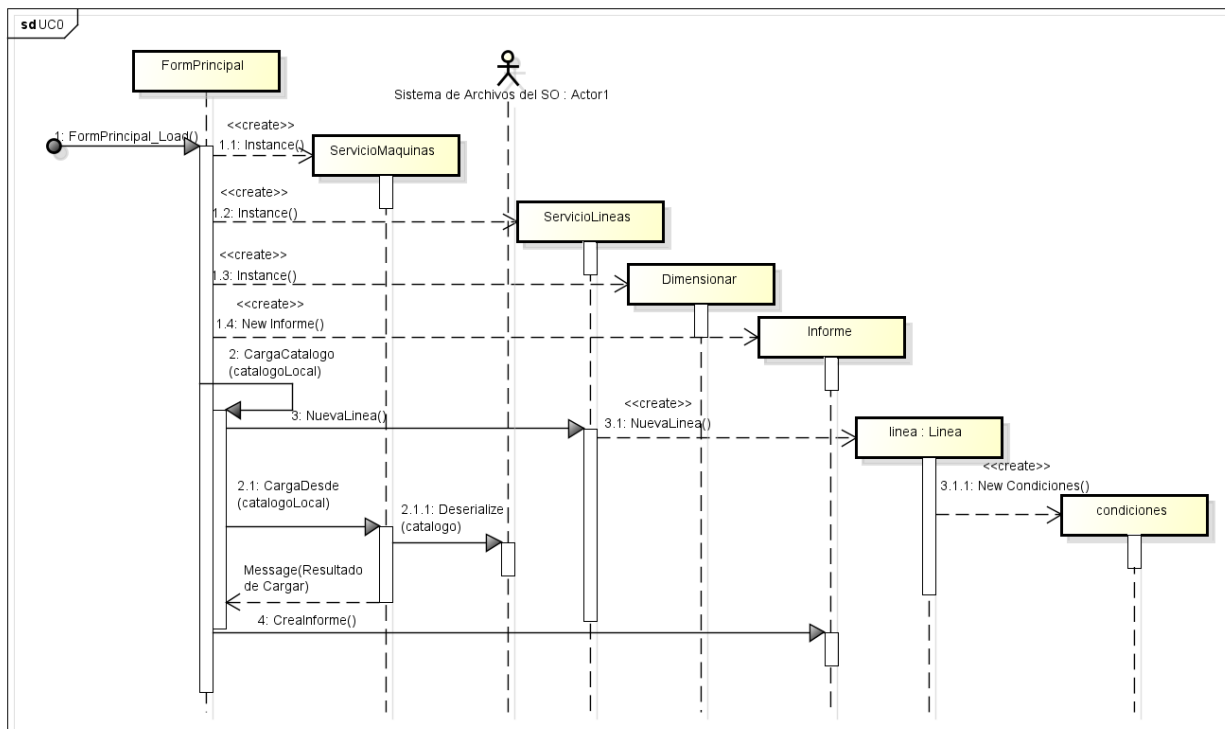


Figura 4.1. Diagrama de secuencia del inicio de la aplicación.

4.2.1.2 Diagrama de secuencia del caso de uso UC1

Se han tomado las siguientes decisiones para el caso de uso Crear nuevo catálogo:

FormPrincipal: formulario inicial de la aplicación, que sirve de núcleo de la misma.

ServicioModelos: gestionará todo lo relacionado con el catálogo y sus familias de máquinas y máquinas (experto en Información). Almacena una lista de familias de máquinas, que representa al catálogo en sí.

ServicioLineas: gestionará lo relacionado con la línea a dimensionar, sus datos y el informe de la misma (experto en Información). Almacena la línea a dimensionar.

Línea: objeto software que representa la propia línea a dimensionar. Se genera aquí una nueva línea por si la modificación en el catálogo ha afectado a la línea existente.

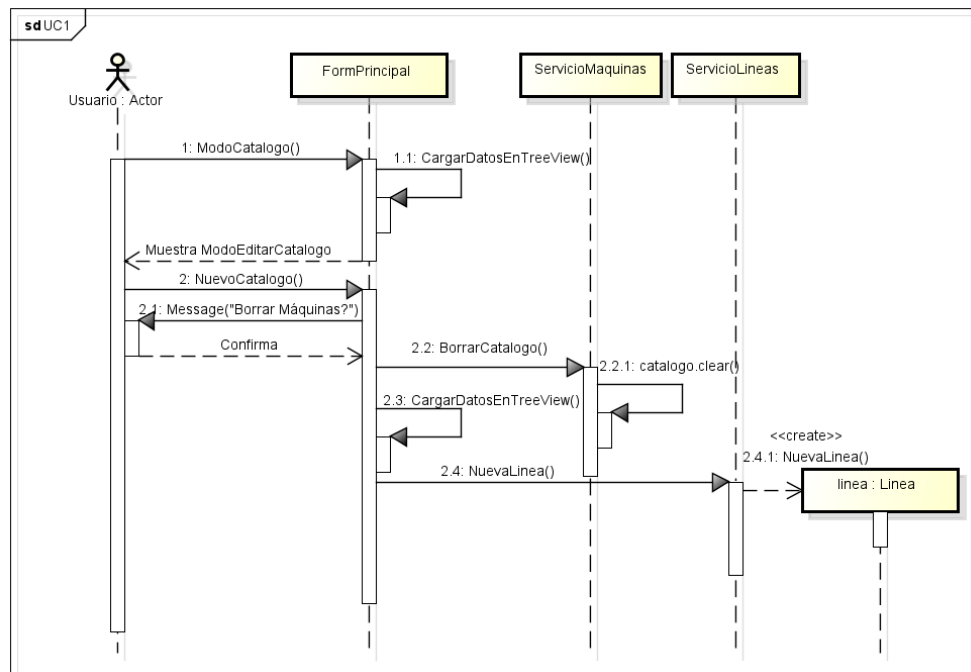


Figura 4.2. Diagrama de secuencia del caso de uso: Nuevo catálogo.

4.2.1.3 Diagrama de secuencia del caso de uso UC2

Para ilustrar el diagrama, se simplifica el sistema de archivos del sistema operativo para facilitar la comprensión, de la misma manera que la consulta con la ventana de diálogo `SaveFileDialog` y los argumentos de su respuesta.

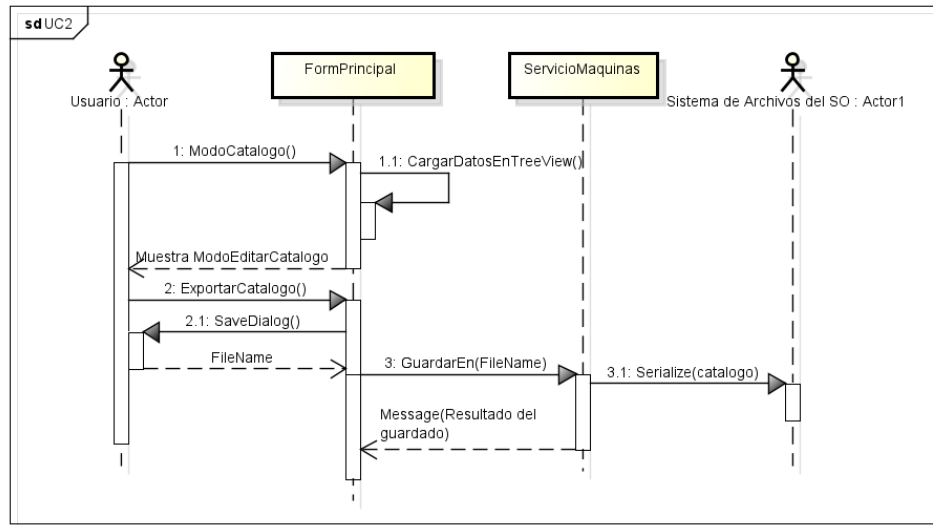


Figura 4.3. Diagrama de secuencia del caso de uso: Exportar catálogo.

4.2.1.4 Diagrama de secuencia del caso de uso UC3

Para ilustrar el diagrama, se simplifica el sistema de archivos del sistema operativo para facilitar la comprensión, de la misma manera que la consulta con la ventana de diálogo `OpenFileDialog` y los argumentos de su respuesta. Además, se genera aquí una nueva línea por si la modificación en el catálogo ha afectado a la línea existente.

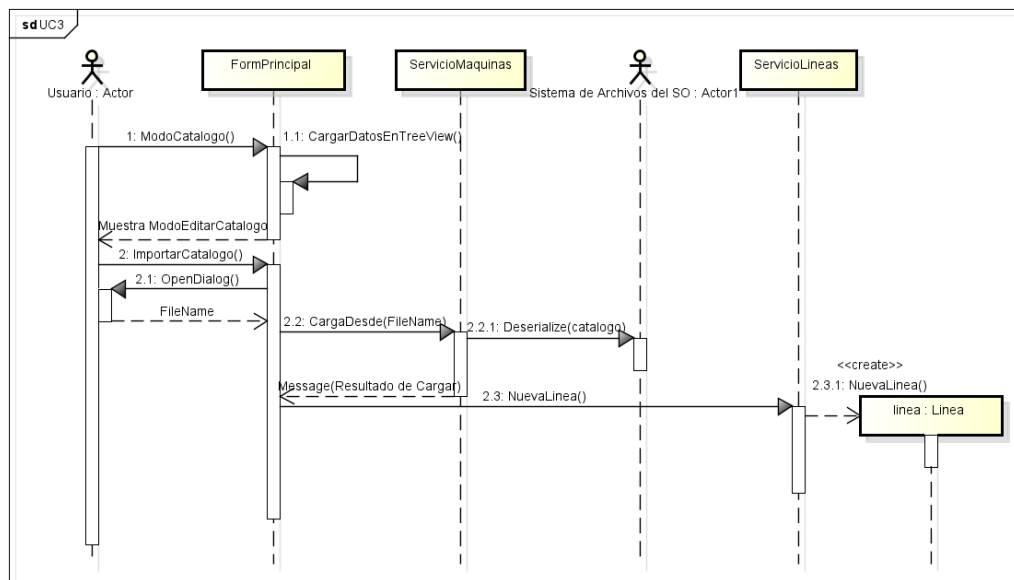


Figura 4.4. Diagrama de secuencia del caso de uso: Importar catálogo.

4.2.1.5 Diagrama de secuencia del caso de uso UC4

Para el caso de uso de guardar catálogo, se usará `catalogoLocal` que representa el fichero que utiliza la aplicación para guardar el catálogo que cargará en memoria en el siguiente inicio.

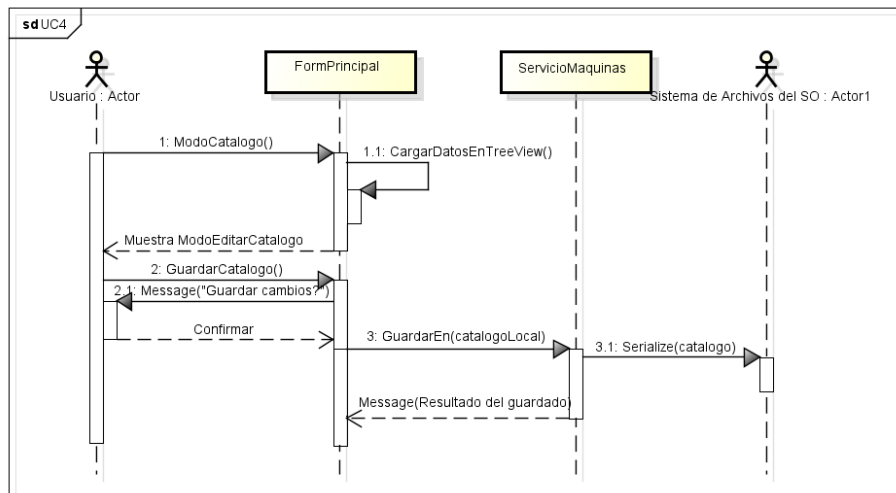


Figura 4.5. Diagrama de secuencia del caso de uso: Guardar catálogo.

4.2.1.6 Diagrama de secuencia del caso de uso UC5

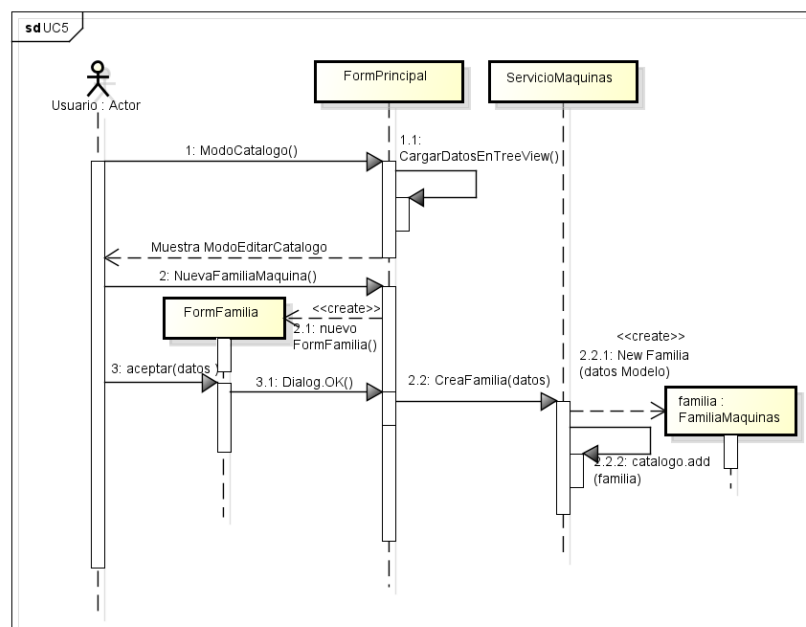


Figura 4.6. Diagrama de secuencia del caso de uso: Nueva familia de máquinas.

4.2.1.7 Diagrama de secuencia del caso de uso UC6

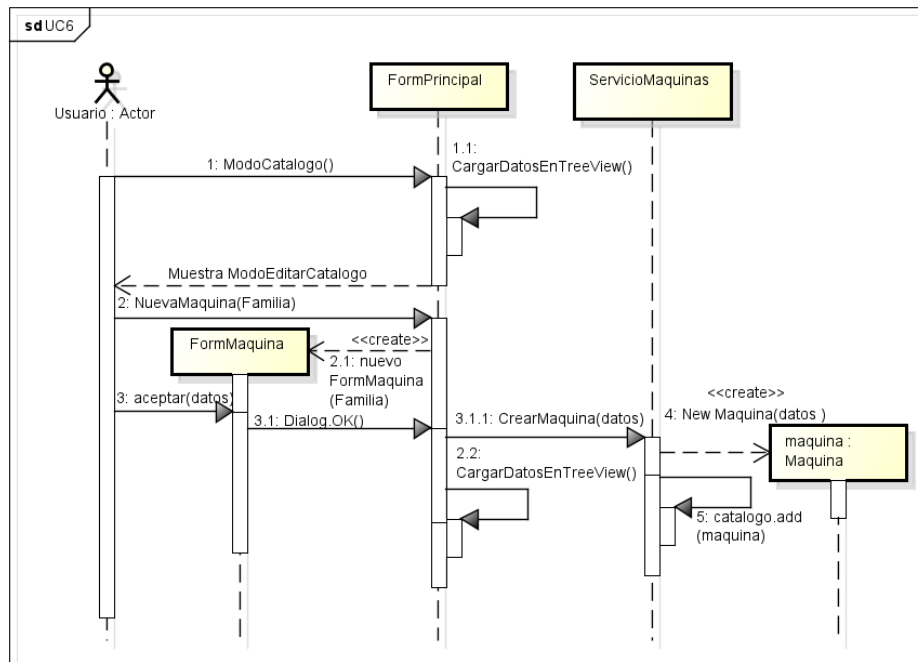


Figura 4.7. Diagrama de secuencia del caso de uso: Nueva máquina.

4.2.1.8 Diagrama de secuencia del caso de uso UC7

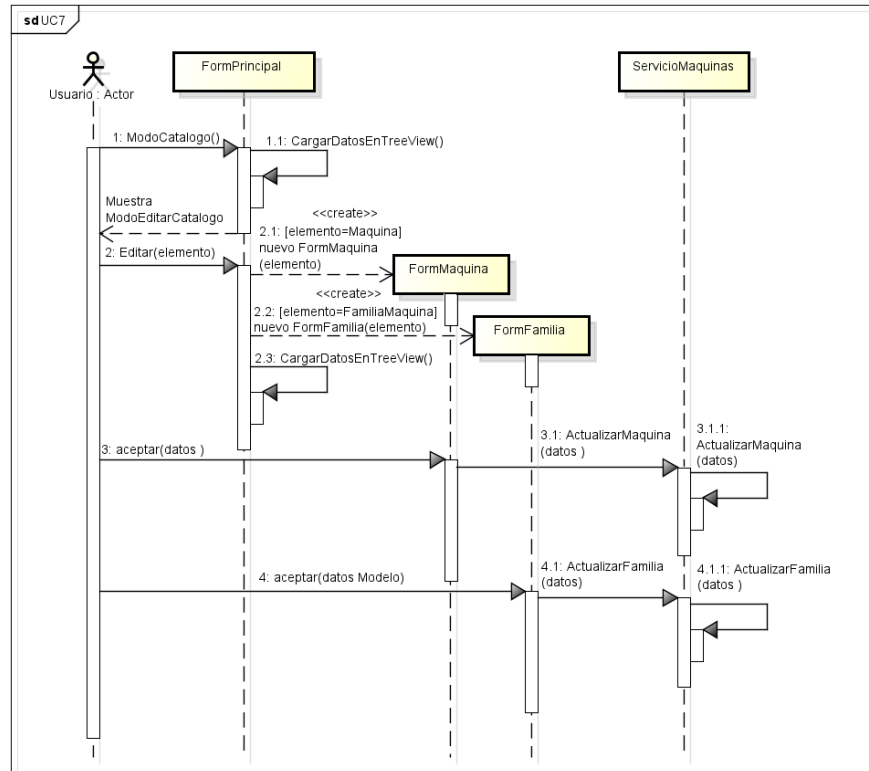


Figura 4.8. Diagrama de secuencia del caso de uso: Editar familia o máquina.

4.2.1.9 Diagrama de secuencia del caso de uso UC8

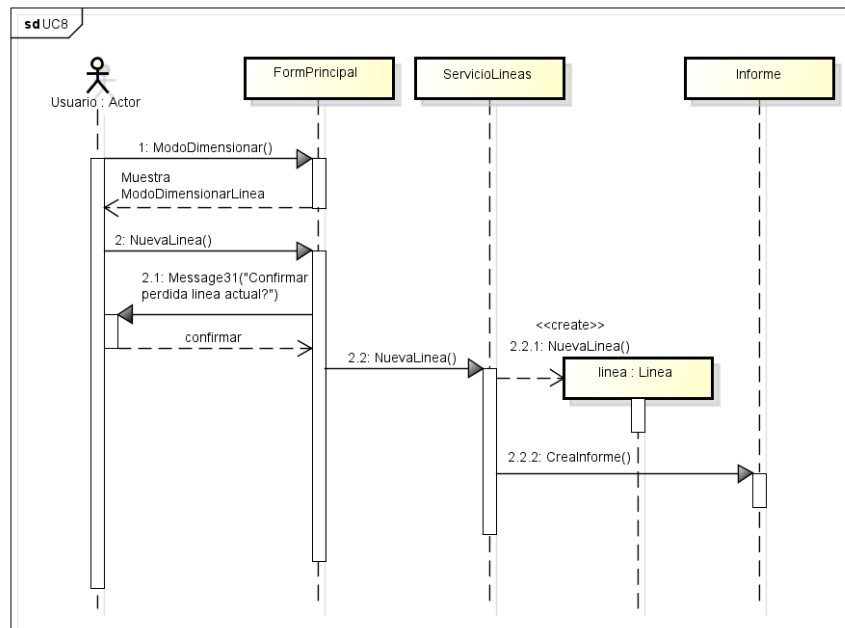


Figura 4.9. Diagrama de secuencia del caso de uso: Nueva línea.

4.2.1.10 Diagrama de secuencia del caso de uso UC9

Al exportar una línea, se añaden a ésta las familias de máquinas utilizadas por ella para disponer de la opción de recuperarlas en el momento que se importe, si en ese momento el catálogo cargado en la aplicación no dispone de ellas.

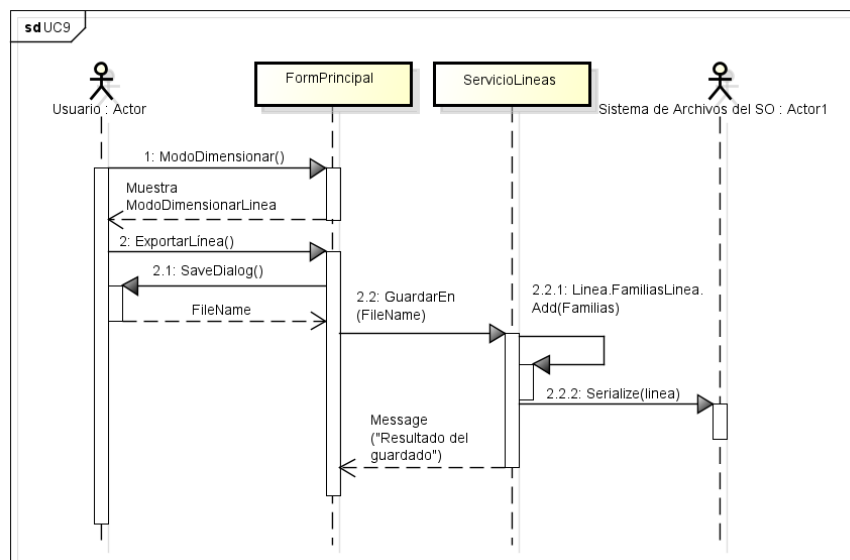


Figura 4.10. Diagrama de secuencia del caso de uso: Exportar línea.

4.2.1.11 Diagrama de secuencia del caso de uso UC10

Al importar una línea, si las máquinas utilizadas por ésta ya se encuentran en el catálogo actual de la aplicación, importa la línea sin más. Si, por el contrario, no están en el catálogo, la aplicación preguntará al usuario si quiere añadirlas desde las almacenadas en la línea a importar.

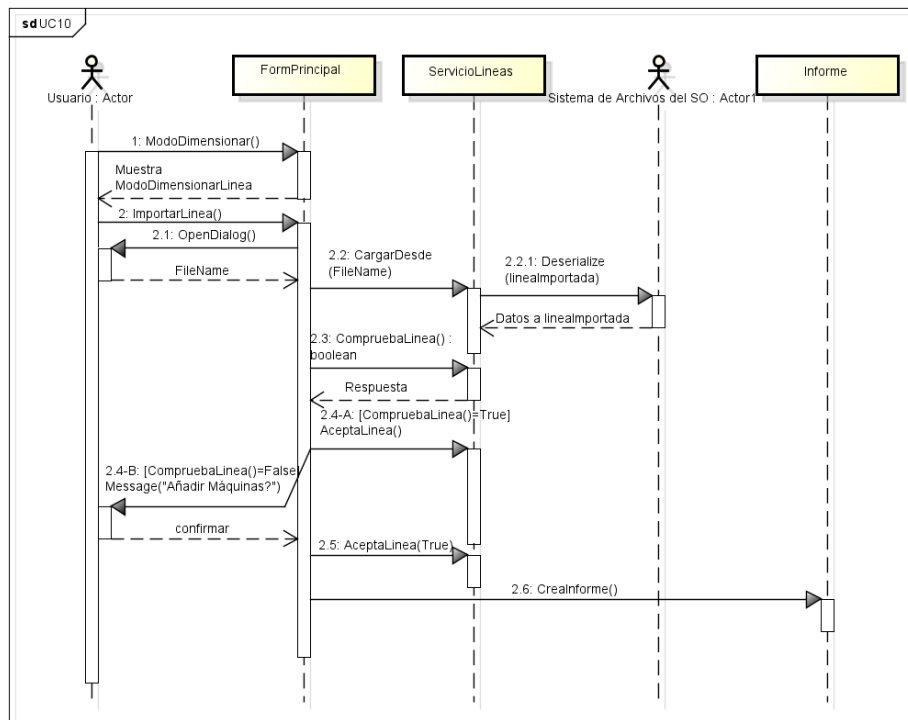


Figura 4.11. Diagrama de secuencia del caso de uso: Importar línea.

4.2.1.12 Diagrama de secuencia del caso de uso UC11

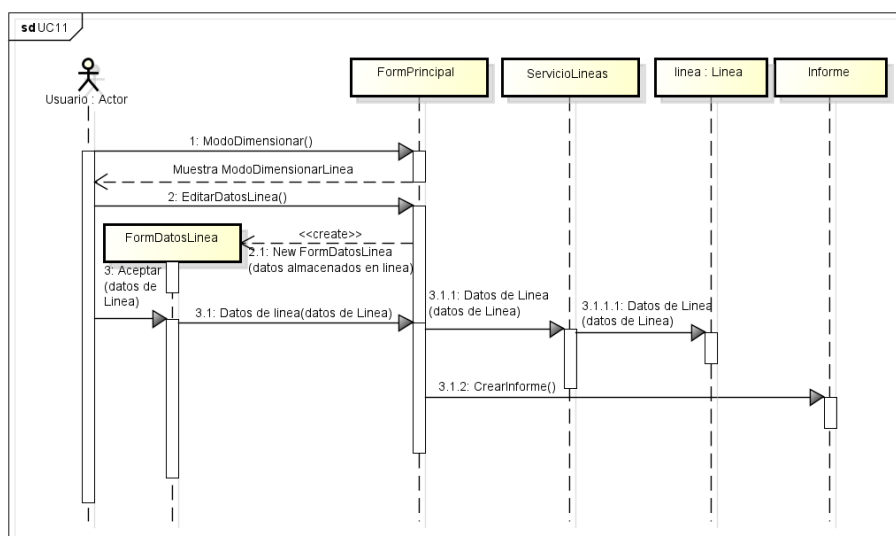


Figura 4.12. Diagrama de secuencia del caso de uso: Editar datos línea.

4.2.1.13 Diagrama de secuencia del caso de uso UC12

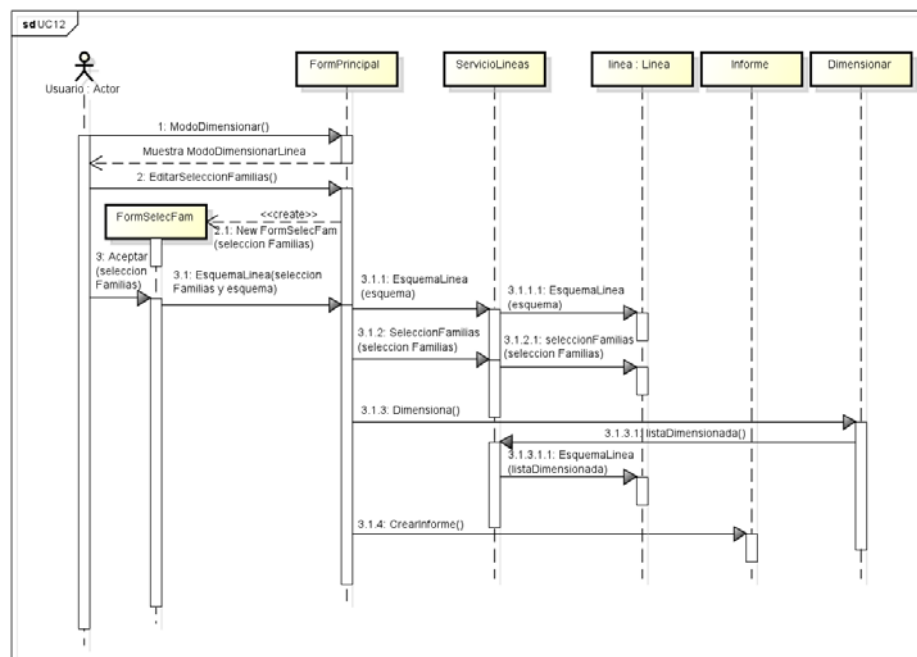


Figura 4.13. Diagrama de secuencia del caso de uso: Selección de familias de máquinas para la línea.

4.2.1.14 Diagrama de secuencia del caso de uso UC13

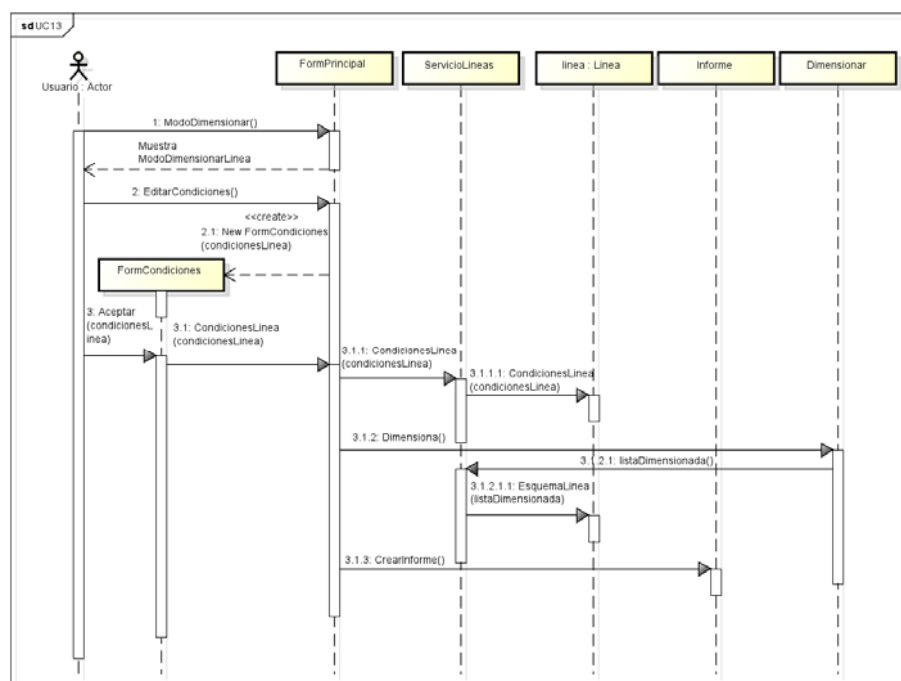


Figura 4.14. Diagrama de secuencia del caso de uso: Editar condiciones.

4.2.1.15 Diagrama de secuencia del caso de uso UC14

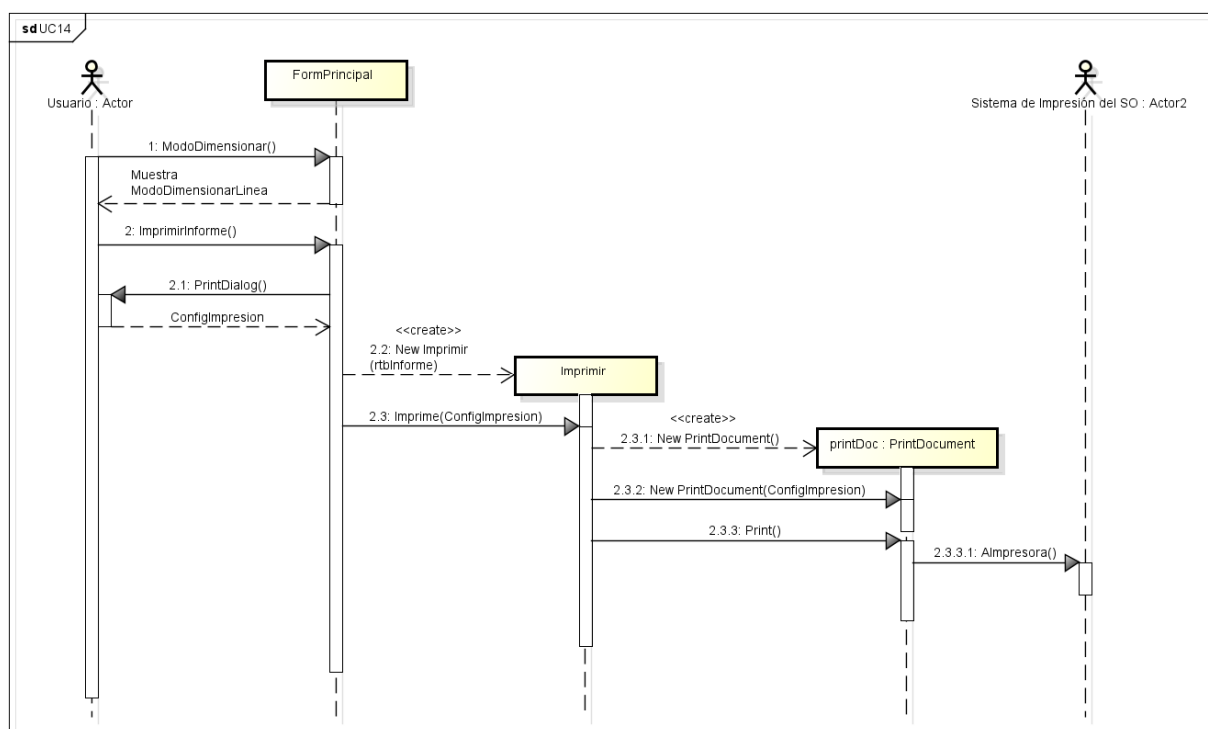


Figura 4.15. Diagrama de secuencia del caso de uso: Imprimir informe.

4.2.1.16 Diagrama de secuencia del caso de uso UC15

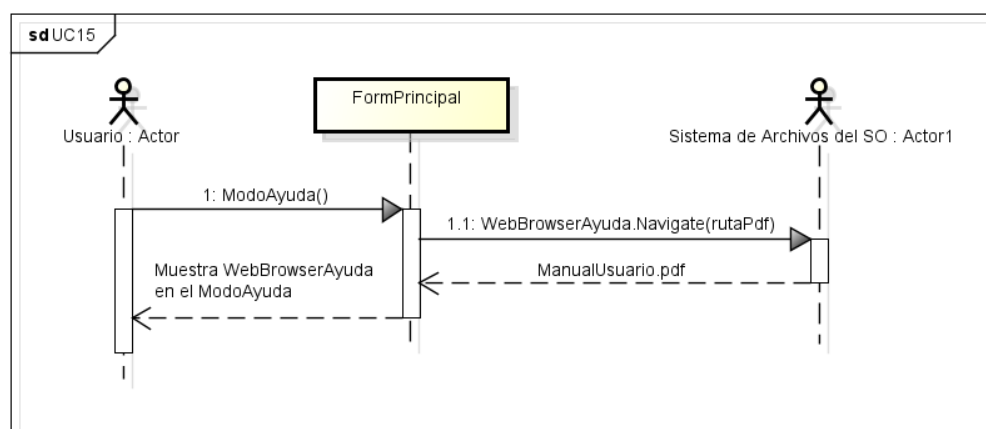


Figura 4.16. Diagrama de secuencia del caso de uso: Mostrar manual de usuario.

formularios que se le presentarán al usuario. Es este formulario el que tomará los datos de esos formularios para después utilizar el resto de clases y cumplir los casos de uso definidos.

4.2.3.2 *Formulario FormFamilia*

Formulario para introducir los datos de la nueva familia de máquinas a crear o editar si ya existente en el catálogo actual de la aplicación. Si se trata de crear una nueva familia, mostrará sus campos en blanco al mostrarse al usuario, para que éste los rellene. Si se trata de editar una familia existente, tomará los valores actuales de esa familia y los mostrará en sus campos para que el usuario los edite. Cuando el usuario acepte, guardará los datos en sus variables para que FormPrincipal pueda disponer de ellos.

4.2.3.3 *Formulario FormMaquina*

Formulario para introducir los datos de la nueva máquina a crear o editar una existente. Si se trata de crear una nueva máquina, mostrará sus campos en blanco al usuario, para que éste los rellene. Si se trata de editar una existente, tomará los valores actuales de esa máquina y los mostrará en sus campos para que el usuario los edite. Cuando el usuario acepte, guardará los datos en sus variables para que FormPrincipal pueda disponer de ellos.

4.2.3.4 *Formulario FormDatosLinea*

Formulario para introducir los datos informativos de la línea a dimensionar. Estos datos se mostrarán al principio del informe sobre la línea dimensionada. Se trata de datos que el usuario puede introducir según sus necesidades para que queden reflejados en el informe sobre la línea, para clasificación y/o mostrar cualquier otra información que, en lenguaje natural, quiera mostrar al destinatario del informe.

4.2.3.5 *Formulario FormSelecFam*

Formulario para que el usuario seleccione las familias de máquinas que compondrán la línea, así como su orden. Esa selección se realiza añadiendo familias de máquinas representadas gráficamente como una caja con su nombre, a un panel, donde el usuario podrá reorganizarlos arrastrando cada caja a la posición deseada, eliminando la que desee y añadiendo nuevas. Se obtiene una lista con los identificadores de las familias de máquinas seleccionadas, así como una imagen del esquema formado por esa lista.

4.2.3.6 *Formulario FormCondiciones*

Formulario para la selección de las condiciones de dimensionamiento de la línea. Contendrá la información relacionada con la producción mínima requerida, el paralelismo de máquinas (cantidad máxima de máquinas trabajando en paralelo para cumplir la producción mínima), y las características a minimizar en la selección.

4.2.3.7 *Formulario FormAcercaDe*

Formulario informativo que muestra datos sobre la aplicación. Muestra: nombre, versión, compañía y una descripción.

4.2.3.8 *Clase RepresentacionFamilia*

Clase para definir un control personalizado que representará a las familias de máquinas en el esquema de selección de familias de máquinas de la línea a dimensionar (las cajas mostradas en el panel). Hereda de la clase "Control" de Windows Forms. Es utilizada por el formulario FormSelecModClases.

4.2.3.9 *Clase Imprimir*

Clase donde se encapsula todo lo necesario para realizar la impresión del informe. Toma el RichTextBox del formulario principal en el cual se muestra el informe generado, tal cual; define el área a imprimir, a partir de la configuración de impresión pasada y la va llenando con el contenido del RichTextBox. Cuando completa un área, pero aún no ha acabado el RichTextBox, informa al sistema de impresión de que aún quedan páginas por imprimir y vuelve a rellenar otra área de impresión, hasta que no queda más contenido en el RichTextBox.

4.2.3.10 *Clase Informe*

Clase para instanciar los informes que la aplicación irá generando. Servirá para almacenar los datos que se incluirán en el informe y para formatear el texto antes de mostrarlo en el formulario principal y enviarlo a impresión. Hace uso del ServicioLineas para obtener los datos de la línea que se mostrarán en el informe. Su resultado quedará guardado en un RichTextBox que FormPrincipal utilizará para mostrar al usuario.

4.2.3.11 *Clase Dimensionar*

Esta clase encapsula todo lo relacionado con los cálculos de selección según las condiciones del usuario. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame. Tomará cada una de las familias de máquinas, seleccionadas por el usuario para la línea, y seleccionará la máquina, de cada familia de máquinas, que mejor se adapte a las condiciones indicadas por el usuario. El resultado será una lista de máquinas que conformarán la línea dimensionada.

4.2.3.12 Clase ServicioMaquinas

Esta clase es el gestor de las familias y máquinas. Alojará el catálogo (listado de familias de máquinas) y todos los métodos para crear, editar, añadir o eliminar del catálogo y demás interacciones con ellas. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame.

4.2.3.13 Clase FamiliaMaquina

Clase que representa las familias de máquinas. Posee los atributos que reflejan las características de cada familia de máquinas, así como una lista para albergar las máquinas que pertenezcan a esta familia.

4.2.3.14 Clase Maquina

Clase que representa a cada máquina. Posee los atributos que reflejan las características de cada máquina (nombre, descripción, producción, operarios, potencia eléctrica, valor de coste y valor de venta. También dispone de los métodos para actualizar dichos atributos, así como para mostrarlos en texto, con o sin el valor de coste.

4.2.3.15 Clase ServicioLineas

Esta clase es el gestor de las líneas a dimensionar. Alojará la línea sobre la que se está trabajando en la aplicación y todos los métodos para crearla, editarla, exportarla o importar una existente en fichero. Para la importación, gestionará la carga o no de las familias de máquinas incluidas en la línea si no se encuentran en el catálogo actual. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame.

4.2.3.16 Clase Linea

Clase que representa la línea a dimensionar. Contendrá los atributos que definen sus propiedades, tales como los datos identificativos a mostrar en el informe de la propia línea (nombre, cliente, proyecto, versión, descripción), la fecha en que se creó, las condiciones de dimensionamiento seleccionadas para esta línea, la selección de familias de máquinas que forman su estructura y la selección de máquinas resultado del dimensionamiento. Además, dispone de una variable para almacenar las familias utilizadas en ella en caso de que se exporte a un fichero.

4.2.3.17 Clase Condiciones

Clase que encapsula las condiciones de dimensionamiento, elegidas por el usuario, para una línea en particular. Formará una característica de esa línea y será utilizada para el cálculo de dimensionamiento. Cuenta con las siguientes variables: la producción mínima requerida, el paralelismo (cantidad máxima de máquinas en paralelo para conseguir la producción mínima) y las características a minimizar en la selección.

4.3 CUMPLIMIENTO DE REQUISITOS

Para que el diseño de la aplicación cumpla con lo esperado, se debe dar respuesta a los requisitos analizados en el capítulo anterior. Como se ha mostrado, la respuesta a los requisitos funcionales (R.1.x) se da con los diagramas de secuencia mostrados en los apartados anteriores. Para los demás requisitos, se toman las siguientes decisiones:

4.3.1 Facilidad de uso

R.2.1: Se realizará la interfaz de usuario mediante formularios de Windows con un patrón de distribución clásico siguiendo los principios de usabilidad y buenas prácticas de diseño de interfaces de usuario extraídas de DesignPrinciples de Microsoft.com (DesignPrinciples, WEB, 2023).

R.2.2: Para facilitar el uso de la aplicación, además del manual de usuario que se incluye con la misma (y se puede consultar desde la propia aplicación), se incluirán tooltips para los controles relevantes de la interfaz de usuario (un pequeño cuadro de texto, con información sobre la acción que realiza el control, que aparecerá al situar el ratón sobre él).

R.2.3: Como ya se ha comentado, la aplicación permitirá consultar su manual de usuario directamente en su ventana principal, sin perjuicio de las operaciones que se estén realizando en ese momento, es decir, la consulta del manual no entrará en conflicto con el estado actual del catálogo ni de la línea a dimensionar.

4.3.2 Fiabilidad

R.3.1: Como se ha visto en los casos de uso correspondientes, la aplicación permitirá la exportación (y obviamente, también la importación) del catálogo actual y de la línea a dimensionar, lo que permitirá al usuario realizar copias de respaldo si así lo desea.

4.3.3 Rendimiento

R.4.1: La complejidad de los cálculos a realizar por la aplicación es baja. Se ha pretendido que, con un tamaño de problema estándar, los tiempos de ejecución sean despreciables.

4.3.4 Soporte

- **R.5.1.-** Por la lógica de la aplicación, si no se han guardado explícitamente los cambios realizados en el catálogo, sean ediciones de familias o máquinas o importación de nuevos catálogos, el hecho de salir sin guardar y volver a ejecutar la aplicación devolverá a ésta a su estado inicial (cargará el catálogo actual que se hubiese guardado con anterioridad). En cuanto a la línea a dimensionar, ocurrirá lo mismo, aunque

puede hacerse más rápido pulsando directamente el botón de “Nueva Línea”, que cargará una línea por defecto en la aplicación.

- **R.5.2.-** En el [anexo A “Manual de usuario”](#) del presente documento se muestra una copia del manual que acompañará a la aplicación y que se podrá consultar desde la misma en el “modo ayuda”. Consultar ese manual en la aplicación no afectará al estado del catálogo actual ni de la línea dimensionada.
- **R.5.3.-** Se pondrán a disposición medios de contacto para soporte (e-mail) indicados en la ventana “Acerca de” de la aplicación y en su correspondiente apartado en el manual de usuario.

4.3.5 Implementación

- **R.6.1.-** La aplicación será desarrollada en Visual Basic.NET utilizando .NET Framework 4.7.2, por lo que su ejecución estará asegurada en los sistemas operativos que den soporte a este framework: Microsoft Windows 11, 10, 8.1, 7 SP1 (Requisitos .NET, WEB, 2024).
- **R.6.2.-** Como se ha visto en la explicación para R.4.1, la aplicación no necesitará más capacidad de procesador que el requerido por el sistema operativo menos demandante, en este caso W 7.1SP1.
- **R.6.3.-** Se codificará la aplicación utilizando Visual Basic.NET y se generará una completa documentación de su código (Ver [Anexo Documentación de código](#)) que permitirá su reutilización.

4.3.6 Operaciones

- **R.7.1.-** Se utilizará Microsoft Visual Studio Installer Projects para el empaquetado de la aplicación, generando un instalador tipo Wizard que indicará los pasos a seguir al usuario.

4.3.7 Empaquetamiento

- **R.8.1.-** Siguiendo lo comentado en el apartado anterior, en el empaquetado se incluirá el manual de usuario, que el propio instalador ubicará en el mismo directorio donde instale la aplicación. Después será la aplicación la que, además, generará allí su fichero para guardar el estado del catálogo actual cuando sea necesario por primera vez.

4.4 COBERTURA DE REQUISITOS

Se presenta en este apartado cómo se ha resuelto cumplir cada uno de los requisitos descubiertos en el análisis del capítulo anterior, con los componentes y decisiones tomadas en el presente capítulo. La idea de este apartado nace de la presentación de la matriz requisitos-componentes mostrada en (Gómez y Moraleda, 2014), debidamente adaptada al presente proyecto.

	FromPrincipal	ServicioMaquinas	ServicioLineas	Linea	FormFamilia	FamiliaMaquina	FormMaquina	Maquina	Informe	FormDatosLinea	FormSelecFam	Dimensionar	FormCondiciones	Imprimir	ManualUsuario.pdf	FormAcercaDe	Condiciones
R.1.1	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-
R.1.2	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R.1.3	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-
R.1.4	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R.1.5	✓	✓	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-
R.1.6	✓	✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-
R.1.7	✓	✓	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-
R.1.8	✓	-	✓	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-
R.1.9	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R.1.10	✓	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
R.1.11	✓	-	✓	✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-
R.1.12	✓	-	✓	✓	-	-	-	-	✓	-	✓	✓	-	-	-	-	-
R.1.13	✓	-	✓	✓	-	-	-	-	✓	-	-	✓	✓	-	-	-	✓
R.1.14	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
R.1.15	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-
R.2.1	✓	-	-	-	✓	-	✓	-	-	✓	✓	-	✓	-	-	✓	-
R.2.2	✓	-	-	-	✓	-	✓	-	-	✓	✓	-	✓	-	✓	✓	-
R.2.3	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-
R.3.1	✓	✓	✓	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-
R.4.1	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-
R.5.1	✓	✓	✓	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-
R.5.2	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-
R.5.3	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-
R.6.1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
R.6.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
R.6.3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
R.7.1 *	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R.8.1 *	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Tabla 4.1. Matriz Requisitos-Componentes

Nota: Los requisitos R.7.1 y R.8.1 no se cumplen con los componentes aquí representados, si no según lo explicado en el apartado anterior. [Ver 4.3.- Cumplimiento de requisitos.](#)

4.5 CONCLUSIONES

Durante el presente capítulo se ha pasado de los resultados del análisis del capítulo anterior, a la obtención de un diseño del proyecto, los planos que indican cómo se ha de construir, cómo se debe codificar, la solución.

Para ello se ha estudiado cada requisito del problema, utilizando los casos de uso para desarrollar los diagramas de interacción que dan lugar a los componentes necesarios de la aplicación. Ya que, como se ha comentado, se ha decidido realizar este proyecto bajo el paradigma orientado a objetos, con estos componentes y tomando de base el diagrama de modelo de dominio del análisis, se ha podido crear un diagrama de clases que muestra la arquitectura de la aplicación. El resto de requisitos, que no generan casos de uso, se han estudiado por separado, asignando componentes y/o decisiones de diseño para darles respuesta. Finalmente se ha comprobado la cobertura de requisitos, asegurando que ninguno haya quedado sin recibir respuesta en la solución diseñada.

Con esta información ya es posible pasar a la codificación final de la solución.

Capítulo 5

Implementación y pruebas

5.1 INTRODUCCIÓN

Partiendo de los resultados del diseño sobre la información del análisis del problema, que muestran los “planos” de la herramienta que se está desarrollando, se construye ahora la aplicación a nivel de código fuente (código de la aplicación), ficheros XML (ficheros para gestión de los datos generados por la aplicación), páginas XML y HTML (documentación del código generado), etc. Este conjunto de elementos forma parte del **modelo de implementación**, definido así por el proceso Unificado (Larman 2002). Se detallarán los medios utilizados (hardware y software), el método tomado para la creación del código y las fases en que se ha dividido esta etapa, con los pormenores de cada parte, y se comentarán las pruebas realizadas.

5.2 ENTORNO DE DESARROLLO

Se presenta una pequeña descripción de los medios hardware y software utilizados para esta etapa:

5.2.1 Equipo hardware

Equipo de sobremesa destinado al desarrollo de la aplicación y pruebas

Procesador i5, 48GB RAM, 500GB almacenamiento flash. Sistema operativo: Windows 10 Pro 64bits.

Resolución de pantalla: 1920x1080. Medios de entrada: teclado y ratón estándar.

Equipo portátil para pruebas (Lenovo Thinkpad L390 Yoga)

Procesador i7, 16GB RAM, 500GB almacenamiento flash. Sistema operativo: Windows 11 HOME 64bits.

Resolución de pantalla: 1920x1080. Medios de entrada: teclado, ratón estándar y pantalla táctil.

Equipo portátil para pruebas (Packard Bell Easynote TJ66)

Procesador Core2 Duo, 4GB RAM, 320GB almacenamiento. Sistema operativo: Windows 7 SP1.

Resolución de pantalla: 1366x768. Medios de entrada: teclado y ratón estándar.

5.2.2 Elección del lenguaje

Visual basic.NET: Para el código de las clases y demás componentes de la aplicación. La elección de este lenguaje vino dada por varias razones:

- Facilidad de aprendizaje, orientado a objetos e integración con Visual Studio. Como alumno que no tenía familiaridad con ningún otro lenguaje fuera de los usados durante los estudios del grado, deseaba utilizar un nuevo lenguaje que mantuviese la facilidad de comprensión y curva de aprendizaje de alguno de los ya vistos durante cursos anteriores (como Java), que utilizase Visual Studio (ya que tenía especial interés en familiarizarme con su uso), y que permitiese la programación orientada a objetos, pues desde que se me presentó durante los estudios, siempre me ha parecido un gran paradigma para pensar en la arquitectura de una aplicación.
- Facilidad para realizar interfaces de usuario. Gracias al uso de Windows Forms para .NET, la creación de los formularios para el usuario ha sido un problema menor.
- Fácil acceso a todo tipo de documentación, manuales, tutoriales y ejemplos. Aunque parece que Microsoft ha detenido el desarrollo de VB.NET, mantiene una amplia documentación actualizada, así como cursos y tutoriales en learn.microsoft.com, que permiten a cualquier interesado informarse sobre cómo realizar quizás cualquier solución sobre este lenguaje.

XML: Para los ficheros donde se alojan los datos generados durante el uso de la aplicación, así como para la definición de la documentación de las clases durante el desarrollo del código fuente. Se utiliza este lenguaje como formato de intercambio de datos (para el almacenaje en ficheros de los catálogos de familias de máquinas y de las líneas de producción), por ser un lenguaje universal, reconocido internacionalmente, con más que suficiente soporte de muchas herramientas (entre las que están, por supuesto, las que he utilizado) y la posibilidad de la lectura de los datos escritos por un humano directamente con un editor de texto (Harold y Means, 2004); algo que he encontrado muy útil durante la depuración de los primeros problemas durante la codificación. Además, se utiliza XML como el lenguaje por defecto en vb.NET bajo Visual Studio 2022 para la documentación del código; casualidad que he aprovechado para comentar completamente el proyecto y generar finalmente una documentación detallada del código.

5.2.3 Entorno software

IDE:

[Microsoft Visual studio 2022](#): “El mejor IDE completo para .NET” (Microsoft.com). IDE de Visual Studio es una plataforma de lanzamiento creativa que se puede utilizar para editar, depurar y compilar código y, finalmente, publicar una aplicación. Además del editor y depurador estándar que ofrecen la mayoría de IDE, Visual Studio incluye compiladores, herramientas de completado de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software.

Además, dispone de extensiones muy útiles para la realización de diferentes trabajos adicionales sobre el código, directamente en el propio IDE, como indico a continuación.

Empaquetado:

[Microsoft visual studio installer projects](#): Extensión de Visual Studio para crear instaladores Microsoft Installer de manera sencilla desde el IDE. Se ha usado en este proyecto para la generación del instalador de la aplicación, una vez finalizada la codificación y depuración.

Generación documentación código:

[SandCastle Help File Builder](#): Continuación de la herramienta, creada originalmente por Microsoft en 2006, que permite, entre otras cosas, generar una completa documentación del código de una solución en Visual Studio, partiendo de las etiquetas XML usadas por el programador para documentar y comentar los componentes software. Se ha usado en este proyecto para la generación de la documentación de código, tanto del código de la aplicación como del código de pruebas.

Generación de diagramas durante el análisis y diseño:

[Astah UML](#): Herramienta para el dibujo de diagramas, esquemas y modelado UML. Se ha utilizado para crear los diagramas de casos de uso, diagramas de secuencia y diagrama de modelo de dominio, mostrados en esta memoria. Se ha seleccionado esta aplicación por haber sido utilizada ya en anteriores asignaturas del grado, y por disponer de una licencia de estudiante para su uso.

Diseño de iconos y otros gráficos de la interfaz de usuario:

[Icofx](#): Herramienta para la creación y edición de iconos. Se ha utilizado esta aplicación por su facilidad de uso y simplicidad.

Pruebas

[MSTest](#): Microsoft Testing Framework. Se trata de una extensión para Visual Studio como marco de pruebas de Microsoft para aplicaciones .NET. Permite escribir pruebas, usar el Explorador de Pruebas en el IDE, crear suites de pruebas, ...

Redacción de la presente memoria:

Microsoft Word - Microsoft Office Professional Plus 2019.

5.3 IMPLEMENTACIÓN DE LA SOLUCIÓN

Para construir la aplicación, a nivel codificación, se ha tomado la aproximación de intentar dar solución a cada caso de uso, mediante el seguimiento de los diagramas de secuencia obtenidos en la fase de diseño, a partir de cada caso de uso funcional. Se ha comenzado por la creación y gestión del catálogo de máquinas que serán la base de la aplicación, a continuación, la lógica de selección para el dimensionamiento, siguiendo con la mejora de la interfaz de usuario y finalmente la integración de todos estos elementos. Para cada caso de uso se creaba el código, formulario y demás clases necesarias para cumplirlo. En una primera aproximación, se ha puesto el foco en la lógica del negocio y no en la interfaz de usuario, generando formularios muy sencillos, con los controles más básicos para centrarse más en la funcionalidad que requería cada caso de uso. Esto dio lugar a una primera versión de la aplicación y a una arquitectura de clases y formularios que servirían de partida para centrarse posteriormente en la lógica de la interacción con el usuario. En esta segunda etapa se generaron varios prototipos de formularios, formas de los controles, disposiciones de los elementos, colores, iconos e imágenes con el fin de obtener una interfaz amigable, sencilla y clara. Como tercera etapa, se refinó lo obtenido en estas fases para dar lugar a la aplicación final. Finalmente, en la cuarta etapa, se realizó el empaquetado y la generación de documentación necesaria.

El proceso de desarrollo detallado ha sido el siguiente:

5.3.1 Fase 1: Desarrollo de la lógica de negocio

En las primeras versiones del proyecto, el trabajo se centró en la codificación de la lógica de negocio de la aplicación. Utilizando la bibliografía necesaria sobre VB.NET (Olmsted, 2023) (Ceballos, 2021) y Visual Studio (Pérez y Teheran, 2022), se implementaron las clases y métodos que forman el núcleo funcional de la aplicación: `ServicioMaquinas`, `Dimensionar`, `ServicioLineas`, `FamiliaMaquina`,

Maquina, Linea, Condiciones, Informe, Imprimir, además de un primer diseño simple del formulario principal `FormPrincipal`, y otros, para poder ejecutar lo codificado.

Creación de las clases de servicios: `serviciosMaquinas` y `serviciosLineas`

Las primeras clases en las que se puso el foco a la hora de codificar fueron las que dieran servicio a los grupos de caso de uso más importantes: los que implican la gestión del catálogo de máquinas y los que implican la gestión de las líneas a dimensionar. Se agrupó en sus respectivas clases de servicios los métodos y atributos necesarios para la creación, almacenamiento y gestión de las partes que afectaban; esto es: `servicioMaquinas` se encargará del catálogo de familias de máquinas y por ello todo lo relacionado con las clases `FamiliaMaquina` y `maquina`. `servicioLineas` se encargará de la gestión de la línea a dimensionar y las opciones para importación y exportación de la misma. Es decir, se les ha asignado las responsabilidades de “creador” y de “experto en Información”.

La gestión de información en ficheros XML

Para el intercambio de datos (guardado, exportación e importación del catálogo de máquinas, en `servicioMaquinas`, y exportación e importación de línea en `servicioLineas`) se ha recurrido al uso del paquete `XML.Serialization`, que facilita la serialización de objetos software a ficheros XML (y viceversa con la deserialización) y a `System.IO` que proporciona las operaciones de entrada y salida para la gestión de esos ficheros de datos (Serialización, WEB, 2024).

Método dimensiona

En relación al requisito R.4.1. sobre rendimiento, el algoritmo más complejo se da al realizar la selección de máquinas, en la clase `Dimensionar`. El método que realiza este cálculo es `Dimensiona()`, que posee la estructura mostrada en la tabla siguiente. Se puede consultar el código completo del método al final de este documento (Ver [anexo B: Código Fuente](#)).

El tamaño del problema vendrá dado por:

SeleccionFam: Se toma este valor como x . Se trata del número de familias de máquinas seleccionadas para la línea. En la práctica, será inusual el caso en que pase de las decenas, y totalmente improbable que pase de las centenas. Además, por programación, se ha limitado este número a 48 para mantener el esquema a generar legible (ver atributo `limiteSeleccion` en la clase `FormSelecFam`).

familia.Maquinas: Se toma este valor como y . Se trata del número de máquinas disponibles para selección en esa familia de máquinas en particular. En la práctica, también sería improbable que pase de las centenas; más de mil máquinas en una misma clasificación no tiene sentido.

Paralelismo: Se toma este valor como z . Se trata del número de máquinas en paralelo que se permiten seleccionar para cada familia de máquinas. Es raro ver hoy en día una línea de producción de la industria conservera que disponga de más de 10 máquinas trabajando en paralelo, aún en los procesos más lentos que más requieren de paralelismo.

Estudiando ahora el algoritmo, se tiene un bucle exterior (For each id In SeleccionFam) que se ejecuta x veces $O(x)$. Dentro de él, por un lado, otro bucle (For each maq In familia.Maquinas) que se ejecuta y veces $O(y)$ y que a su vez posee otro bucle (For i As Integer = 1 To ...Paralelismo) que se ejecuta z veces. Por otro lado, hay una ordenación dentro del bucle exterior (OrderBy...ThenBy...) que se ejecutarán, en el peor de los casos, $(x \cdot y)$ veces $O((x \cdot y)\log(x \cdot y))$.

```
Public Sub Dimensiona()  
...  
For Each id In ...SeleccionFam  
...  
    familia = ...GetFamiliaID(id)  
    For Each maq In familia.Maquinas  
        For i As Integer = 1 To ...Paralelismo  
            If maq.Produccion * i >= ...ProduccionMin Then  
                If i = 1 Then ...  
                Else ...  
                End If  
            End If  
        Next  
    Next  
    If listaCumplenProduccion.Count = 0 Then ...  
    End If  
    If ...MinimizarPrimaria IsNot Nothing AndAlso  
        ...MinimizarSecundaria IsNot Nothing Then  
        Dim c1 As String = ...MinimizarPrimaria  
        Dim c2 As String = ...MinimizarSecundaria  
        listaOrdenados = listaCumplenProduccion _  
            .OrderBy(Function(obj)obj.GetType().GetProperty(c1).GetValue(obj)) _  
            .ThenBy(Function(obj)obj.GetType().GetProperty(c2).GetValue(obj)) _  
            .ToList()  
        ...  
    ElseIf ...MinimizarPrimaria IsNot Nothing Then ...  
        listaOrdenados = listaCumplenProduccion _  
            .OrderBy(Function(obj)obj.GetType().GetProperty(c1).GetValue(obj)) _  
            .ToList()  
        ...  
    Else ...  
    End If  
Next  
...  
End Sub
```

Listado de Código 5.1.- Algoritmo del método Dimensiona() (Simplificado)

Se tiene entonces una complejidad total de $O(x \cdot y \cdot z + (x \cdot y) \log(x \cdot y))$.

Un límite superior para el peor caso sería cuando $x=y=z$, entonces se tendría $O(n^3 + n^2 \log n^2)$ que se trataría de una complejidad inviable para tamaños de problema elevados, pues “los algoritmos cuadráticos no resultan prácticos para tamaños de entrada superiores a unos pocos miles” (Weiss, 2013), pero en el caso de este proyecto, en el que $x \leq 48$, $y < 1000$, $z < 100$, el tiempo de ejecución en un equipo con los requisitos mínimos solicitados (los de Windows 7 sp1, x86 a 1GHz) sería en torno a 0.005 segundos, valor totalmente válido y que no justificaría una reestructuración integral del código, que enmascararía la legibilidad del algoritmo, estructuras y atributos para lograr una mejora no apreciable por el usuario.

$$\text{Nº de operaciones: } (48 \times 1000 \times 100) + (48 \times 1000) \log(48 \times 1000) = 5544000 \text{ op}$$

$$\text{CPU 1 GHz : } 1000000000 \text{ op/s}$$

$$\text{Tiempo ejecución aproximado: } \frac{5544000 \text{ op}}{1000000000 \text{ op/s}} = \mathbf{0.005544 \text{ s}}$$

Nota: El tiempo real depende de factores adicionales, pero el cálculo valdrá para conocer la magnitud del valor

5.3.2 Fase 2: Diseño y lógica de la interfaz de usuario

Una vez establecida la lógica de negocio, se desarrollaron versiones centradas en crear una interfaz de usuario amigable, sencilla y clara. Esta fase se enfocó en los formularios: `FormPrincipal`, `FormFamilia`, `FormMaquina`, `FormDatosLinea`, `FormSelecFam`, `FormCondiciones`, `FormAcercaDe`.

Durante esta fase se pasó por varios conjuntos de diseños para la interfaz de usuario, construyendo y modificando las clases formularios en distintas versiones: las primeras sólo funcionales sin ajuste a ningún razonamiento en particular, las siguientes ajustándose a razonamientos estéticos y, finalmente, las últimas versiones ya aplicando lo básico y sencillo, según los principios de usabilidad y buenas prácticas de formularios de Microsoft (Design Principles, web, 2023), ya que lo más lógico es tomar el consejo de quien ha creado los propios windows forms.



Figura 5.1. Primeras versiones de la interfaz de usuario.

El formulario principal consiste, finalmente, en una distribución clásica de un menú principal y un panel de trabajo, con una fila de botones para acciones rápidas en la parte superior (en cualquiera de los modos), a los que se les añade un par de botones para cambio rápido entre modos en una barra de estado inferior. El resto de ventanas utilizadas siguen el estándar de diálogos de mensajes de windows forms.

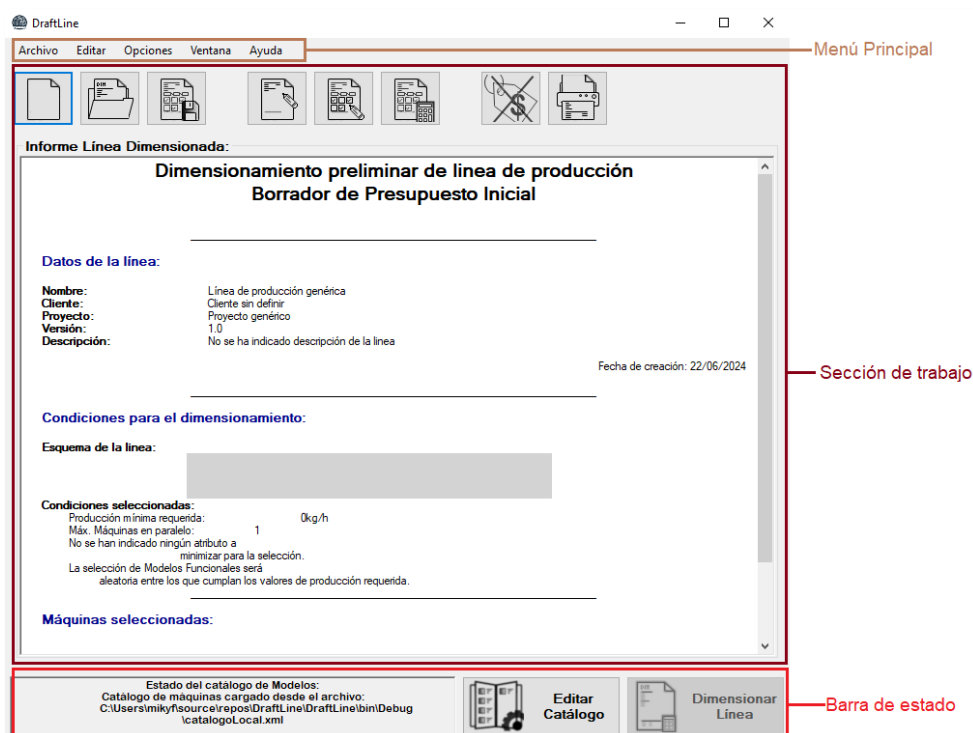


Figura 5.2. Captura de la pantalla principal.

Figura 5.3. Captura de formulario para editar condiciones.

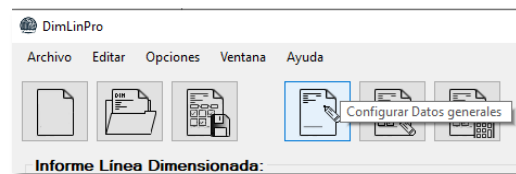


Figura 5.4. Ejemplo de tooltip de ayuda sobre un botón.

Solución gráfica para la selección de familias de máquinas

El punto más importante en cuanto a la interfaz de usuario de este proyecto es, quizás, la facilidad y claridad a la hora de realizar la selección de familias de máquinas para el dimensionamiento de la línea, cumpliendo así el requisito funcional R.1.12.

Para ello se ha dispuesto el formulario `FormSeLecFam`, en el cual el usuario puede seleccionar en un listado la familia de máquinas que desea añadir a la selección. Esta selección se ve reflejada en un esquema gráfico simple que muestra cada una de las familias añadidas para la selección, interconectadas con flechas indicando la lógica de la línea de producción. Este esquema se dibuja sobre un panel con el cual puede interactuar el usuario: mediante selección con el ratón, podrá reordenar la línea que está creando (arrastrando cada familia a su nueva ubicación) o eliminando la familia que desee quitar. De esta forma, la selección es visual, clara y sencilla.

Para lograr esto, se ha creado una nueva clase (`RepresentacionFamilia`), que define un nuevo control de formulario, heredando de la clase predefinida `Control` los métodos para su manejo en el formulario de selección, y sobrescribiendo los métodos requeridos (`OnPaint` para definir su forma, y `OnMouseEnter` y `OnMouseLeave` para capturar los eventos al pasar el ratón y definir el comportamiento de la representación -en este caso sólo cambiar el color).

5.3.3 Fase 3: Unión, depuración y ajustes de lógica de negocio y lógica de interfaz

En esta fase final, una vez definido el funcionamiento de la aplicación y su interfaz de usuario, se realizaron los ajustes finales para que la información proporcionada por el usuario a través de los formularios fuese la esperada por la lógica de negocio definida en la fase 1. Por ejemplo, se ajustó aquí la lógica interna del formulario `FormSelecFam` para que existiese perfecta correspondencia entre la selección de familias desde el punto de vista gráfico con la selección desde el punto de vista lógico (la propia lista de clases familia seleccionada) que después utilizaría el `servicioLineas` para el dimensionamiento.

También se puso especial énfasis en una reordenación de código y clases para lograr la asignación de responsabilidad “controlador” a los formularios, pretendiendo mantener el bajo acoplamiento con las clases que gestionan los datos que éstos obtienen del usuario (Larman, 2002, p.206).

5.3.4 Fase 4: Empaquetado y generación de documentación

Se realiza el empaquetado de la aplicación para obtener un archivo instalador sencillo y de fácil uso por el usuario final. Para ello se utilizó la extensión Microsoft Visual Studio Installer Projects que, una vez instalada en el IDE, permite generar un nuevo proyecto dentro de la misma solución de la aplicación, con la finalidad de configurar el empaquetado de la misma. Se utiliza la opción Wizard, que permite generar una pequeña guía durante la instalación para que el usuario seleccione si desea instalar esta aplicación sólo para él o para todos los usuarios de su equipo. Esta extensión permite determinar qué otros ficheros se deben incluir en la instalación, opción que se ha utilizado para incluir, directamente, el manual de usuario, que después la aplicación, una vez instalada, utilizará para mostrar en su pantalla principal cuando el usuario lo requiera.

Finalmente, se ha generado una detallada documentación de las clases, sus métodos y atributos, de la aplicación mediante la extracción y transformación en página web de ayuda, de los comentarios y descripciones incluidas, en formato XML, a lo largo de todo el código creado, utilizando la herramienta SandCastle Help File Builder. Su funcionamiento es sencillo: tras instalarla en el equipo, se crea un nuevo proyecto de fichero de ayuda, configurando qué tipo de fichero se desea e indicándole qué proyecto de Visual Studio contiene el código a documentar. De esta manera generará una página web, y toda su estructura, con la documentación deseada. Para ello, se ha pretendido ser exhaustivo en la inclusión de información (comentarios y descripciones) a lo largo de todo el código generado durante la escritura de la aplicación.

5.4 PRUEBAS

Para ir comprobando el correcto funcionamiento de las partes que se iban codificando, se realizan pruebas con el objetivo de descubrir errores. Se comienza con pruebas unitarias sobre cada clase y después pruebas basadas en hilos para la integración (siguiendo la nomenclatura de Pressman y Maxim, 2021). Los hilos serán conjuntos de clases que responden a una entrada o evento. Como los casos de prueba deben poder rastrearse hasta los requerimientos de la aplicación, se tomará cada caso de prueba encontrado en el análisis como fuente para realizar estas pruebas de hilos.

Para ambos tipos de prueba se utilizará la herramienta [MSTest](#) de Microsoft en [Visual Studio](#). Para ello, una vez descargada e instalada la extensión MSTest en Visual Studio, se crea un nuevo proyecto (proyecto de pruebas) en el cual se construye una clase de pruebas para cada clase existente en la aplicación, definiendo en ella las pruebas que se deben comprobar, según los métodos, responsabilidades e interacción entre clases. Ver el [anexo D “Código fuente pruebas”](#) y [anexo C “Documentación de código pruebas”](#) al final de este documento.

Estas clases de pruebas llevarán un nombre indicativo tal como “Clase ServicioMaquinasPrueba” que indica que es una clase que prueba la clase original “ServicioMaquinas”.

Además, se utiliza un atributo “TestClass” que marca la clase como prueba y, opcionalmente, inicializa o limpia métodos. Por ejemplo, en la clase ServicioMaquinasPrueba, se marca la clase como “TestClass” y se utiliza “TestInitialize” para marcar un método (“InicializaPrueba”) que borra el catálogo antes de cada prueba.

A continuación, se crean los métodos que comprobarán el comportamiento de la clase a probar. Éstos se marcarán como “TestMethod” y llevarán un nombre también explicativo de tal manera que reflejen la funcionalidad a la que llaman, qué va a hacer y la comprobación de lo que debería haber realizado, por ejemplo: “Public Sub CreaFamilia_AñadeNuevaFamilia_ListaContieneFamilia()”. Dentro de cada uno de estos métodos se reparte mediante el patrón “Arrange-Act-Assert”.

- La sección “Arrange” del método inicializa y configura los valores que se pasarán al método a probar.
- La sección “Act” invoca el método a probar con los parámetros de la sección anterior.
- La sección “Assert” verifica que el método llamado se comporta según lo esperado.

Una vez codificadas así todas las clases que se quieran comprobar, se pueden ejecutar todas las pruebas a la vez, o independientemente, desde el explorador de pruebas del propio Visual Studio, obteniendo la verificación, o el mensaje de error si no ha pasado la prueba, para cada test incluido.

Para las pruebas de la aplicación al completo, así como de la interfaz de usuario, se ha recurrido a pruebas manuales, tanto por mí mismo, como por terceros, entre los que se encuentran usuarios con el perfil objetivo de la aplicación (técnicos-comerciales de una empresa fabricante-integradora de equipos industriales).

5.4.1 Pruebas unitarias

Se realizan pruebas unitarias de caja negra y se asume, en el paradigma orientado a objetos, que la unidad es la clase, por lo que se comprueba si el estado en el que queda la instancia de la clase que se está probando es correcto o incorrecto para los datos que se le pasan como entrada (Polo y Pérez, 2012).

Para cada clase con métodos se crea una clase paralela de prueba, según lo explicado anteriormente, que comprueba cada método pasándole parámetros conocidos y comprobando si la respuesta coincide con la esperada, ya que los datos conocidos incluyen la respuesta esperada.

5.4.2 Pruebas de integración

Se realizan también pruebas entre clases; se observan si los mensajes que fluyen desde los objetos de una clase se envían y reciben en el orden adecuado en la clase receptora y si se producen o no en esta los cambios de estado esperados (Polo y Pérez, 2012).

Se realizan estas pruebas de manera similar a las unitarias pero utilizando llamadas a métodos de más alto nivel que involucren comunicación entre clases.

5.5 CONCLUSIONES

Durante esta fase del proyecto se ha construido “físicamente” la aplicación y su documentación asociada. Para ello se han seguido las guías de transformación “del diseño al código” de la bibliografía utilizada (y referida en cada apartado). Se ha descrito el entorno de desarrollo (equipos y herramientas utilizados). Se ha mostrado la estructura de 4 fases seguida para la implementación de la solución, con descripciones de los problemas resueltos y las decisiones más importantes tomadas en cada una de las fases. Finalmente se ha mencionado el método seguido para la realización de las pruebas utilizadas durante y después de la codificación.

Todo el contenido resultado del proceso recogido en este capítulo puede verse en los anexos de la presente memoria.

Capítulo 6

Conclusiones y trabajos futuros

6.1 INTRODUCCIÓN

Como último capítulo se recogen las conclusiones obtenidas con el trabajo realizado en el presente proyecto, así como ciertos trabajos futuros de especial interés que podrían realizarse como continuación del desarrollo de la herramienta aquí detallada.

6.2 CONCLUSIONES

El trabajo realizado durante este proyecto tiene, como principal objetivo, aplicar los conocimientos adquiridos durante el estudio del grado para resolver un problema real. En este caso se trata de crear la herramienta de dimensionamiento de líneas de producción de la industria conservera que dé respuesta al problema contextualizado al inicio de esta memoria. Para ello se ha seguido un "itinerario" claro y bien documentado en la diversa bibliografía utilizada en esta memoria: obtención del problema, análisis, diseño de la solución, implementación de dicha solución y documentación del proceso.

Recopilando el trabajo realizado y los objetivos logrados en cada una de estas tareas, nos encontramos con:

Obtención del problema.

Lo primero ha sido introducirse en el contexto de la fabricación/implantación de líneas de producción en la industria conservera desde el punto de vista del propio fabricante, los principales problemas existentes a la hora de ofrecer el producto adecuado a un cliente final y cómo se resuelven en la actualidad. Para ello he utilizado la propia experiencia en el diseño de estos equipos en la empresa en la que actualmente trabajo, así como la consulta a los compañeros técnicos comerciales del departamento de ventas. Se ha visto que se presenta aquí el desafío de ofrecer un producto a medida, que quizás no exista todavía, en un tiempo mínimo, con un coste mínimo. Las herramientas utilizadas en la actualidad, aunque muy potentes y útiles para otros ámbitos, no están enfocadas directamente en solucionar este desafío. De esta manera se obtiene el principal objetivo: obtener una herramienta que genere un borrador de presupuesto adecuado a los requerimientos básicos del cliente, flexible a los cambios de dichos requerimientos, con un mínimo consumo de recursos temporales y económicos, que documente el trabajo y ofrezca al cliente final la información básica necesaria para la toma de decisiones.

Análisis y diseño.

Para estas tareas se ha recurrido inicialmente a la documentación utilizada en asignaturas durante el propio grado para, a continuación, ampliarla en base a la propia bibliografía de esa documentación y a diversas consultas en la red. Con ello se han tomado las decisiones necesarias para estructurar el estudio del problema y para el diseño, tanto a nivel de forma y estructural como de fondo, siguiendo pasos pautados por los libros del grado, como por ejemplo el modelo FURPS+ para los requisitos o los tipos de diagramas utilizados para recoger las decisiones tomadas. El resultado del análisis (requisitos y diagramas), realizado a partir de los objetivos planteados en la tarea anterior, ha dado lugar a los resultados finales del diseño (diagramas de secuencia y clases) que se utilizan para la implantación de la solución.

Implementación de la solución.

Partiendo del diseño obtenido se ha procedido a la elección de las herramientas para la realización de la implementación. Se incluyen por tanto las herramientas hardware (en este caso las disponibles al alcance del que escribe y suficientes), la elección del lenguaje (Visual basic.NET por su facilidad de aprendizaje, interfaces de usuario y documentación disponible) y el software a utilizar (aquellas herramientas más adecuadas para cada tarea que, además, no incurrieran en un coste extra). Se ha buscado la documentación necesaria, y recogida en la bibliografía, para utilizar estas herramientas, así como recursos online para la realización del trabajo que se ha dividido en cuatro fases:

- Fase 1: Desarrollo de la lógica de negocio: Donde se ha creado el núcleo de la aplicación, con la implementación de las clases y secuencias de trabajo obtenidas de la tarea de diseño.
- Fase 2: Diseño y lógica de la interfaz de usuario: Donde se prestó especial atención al uso de la aplicación por parte del usuario.
- Fase 3: Unión, depuración y ajustes de lógica de negocio y de interfaz: Con la aplicación ya en un estado funcional desde el punto de vista de negocio y de interfaz, se realizaron en esta fase los cambios necesarios para que funcionase según lo esperado, así como una mejora en el orden de código y clases. También se completó la documentación de código en forma de comentarios con intención de que sea lo más auto explicativo posible.
- Fase 4: Empaquetado y generación de documentación: En esta fase, se obtuvo el resultado final de la implantación: el archivo instalador de la aplicación y la documentación de clases partiendo de los comentarios de código anteriormente mencionados.

Documentación del proceso.

Utilizando los resultados obtenidos en cada una de las tareas anteriores, así como la documentación de código mencionada, se redacta la presente memoria del proyecto que recoge la información suficientemente detallada de la herramienta desarrollada para facilitar futuras modificaciones, actualizaciones y nuevas funcionalidades que puedan surgir.

Finalmente, se puede comprobar que la herramienta desarrollada cumple los objetivos iniciales:

- Genera un borrador de presupuesto en función de los requisitos introducidos, con un coste de tiempo despreciable.
- Permite cambios sobre el mismo borrador en base a modificaciones de los requisitos inicialmente introducidos.
- Permite la exportación del trabajo realizado sobre la herramienta, tanto del catálogo de máquinas disponibles (vía ficheros XML) como de los propios borradores de presupuesto (vía impresión).
- Recoge la información requerida tanto por el usuario como por el cliente en el borrador de presupuesto generado.

6.3 TRABAJOS FUTUROS

Aunque la aplicación desarrollada cumple con los requerimientos expuestos inicialmente, éstos estaban limitados al ámbito de este PFG, y se pueden ver rápidamente ampliados al comenzar a trabajar con la herramienta. El descubrimiento de nuevas necesidades es una característica común en las puestas en marcha de nuevas herramientas.

Se recogen a continuación algunas propuestas que podrían servir para aumentar el potencial de uso de la aplicación, mejorar su utilización y su integración en los procedimientos de posibles empresas usuarias.

Ampliación de la cantidad de características de las máquinas

Durante el desarrollo de la aplicación se limitaron las características de las máquinas a seleccionar a 5 variables básicas (producción, operarios, potencia eléctrica, valor coste y valor venta), por ser las más comunes en casi cualquier tipo de máquina, y suficientes para la funcionalidad requerida, pero es posible que, para ciertos nichos de mercado en la industria conservera, aparezcan otras características comunes que interesase tener en cuenta.

La mayor dificultad en esta ampliación recae más en el estudio de campo necesario y a medida sobre las características que interese incluir, ya que, desde el punto de vista del software, es un problema trivial.

Catálogo de clientes

Al igual que se dispone de un catálogo de máquinas en la aplicación, sería interesante poder disponer de un catálogo de clientes donde poder escoger el cliente al que se le va a ofrecer la línea que se está dimensionando. Incluso se podría añadir una nueva funcionalidad, relacionada con el ajuste de los valores de venta en función del cliente seleccionado.

Esta solución bien se podría implementar de la misma manera que el catálogo de máquinas, lo cual resultaría trivial, añadiendo una base de datos más desarrollada o implementando las clases y métodos necesarios para realizar consultas a una base de datos propiedad del usuario.

Ampliación de apartados del informe – Condiciones particulares del presupuesto

Aunque la aplicación ya dispone de un campo de texto en el informe generado, que el usuario puede utilizar para añadir cualquier otro dato o información que necesite incluir en el borrador de presupuesto, es posible que algún usuario desee añadir otro campo de texto, por ejemplo, para las condiciones particulares del presupuesto.

Este añadido tampoco tendría mayor dificultad que modificar el formulario de datos de la línea para añadir un nuevo campo, o crear un nuevo formulario para tal cumplido, además de añadir este campo a las clases que finalmente lo muestran en el informe.

Bibliografía

(Anfaco, 2024)

Disponible en: <https://anfaco.es/> (Consultado: 2024).

(Astah, 2024)

Disponible en: www.astah.net (Consultado: 2024)

(Autodesk, 2024)

Disponible en: <https://www.autodesk.com/es> (Consultado: 2024)

(Autodesk Factory Design, 2024)

Disponible en:

https://help.autodesk.com/view/FDU/2024/ENU/?guid=FDU_About_Factory_Design_Uilities_html

(Consultado: 2024)

(Ceballos, 2021)

Ceballos Sierra, F.J., (2021), Visual Basic. Aplicaciones de escritorio con Windows Forms. Autoedición.

(Dassault, 2024)

Disponible en: <https://discover.3ds.com/es/delmia-digital-manufacturing-software> (Consultado: 2024).

(DesignPrinciples, WEB, 2023)

Conceptos básicos de diseño para aplicaciones de escritorio, (2023), Microsoft.com. Disponible en:

<https://learn.microsoft.com/es-es/windows/win32/uxguide/designprinciples> (Consultado: 2024).

(Emerito, 2024)

Disponible en: <https://emeritofoodtechnology.com/> (Consultado: 2024).

(Gaictech, 2024)

Disponible en: <https://www.gaictech.com/> (Consultado: 2024).

(Grady, 1992)

Grady, R., (1992). Practical Software Metrics for Project Management and Process Improvement.

Englewood Cliffs, NJ.: Prentice-Hall.

(Groover, 2013)

Groover, M.P., (2013). Fundamentals of Modern Manufacturing: Materials, Processes, and Systems. Wiley.

(Harold y Means, 2004)

Harold, E.R. y Means, W.S., (2004). XML in a Nutshell. O'Reilly Media, Inc.

(Hermasa, 2024)

Disponible en: <https://hermasa.com/> (Consultado: 2024).

(JBT Corporation, 2024)

Disponible en: <https://www.jbtc.com/es/> (Consultado: 2024).

(Larman, 2002)

Larman, C., (2002). UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson.

(Maconse, 2024)

Disponible en: <https://maconse.com/> (Consultado: 2024).

(Olmsted, 2023)

Olmsted, A., (2023), Visual Basic Quickstart Guide, Packt Publishing.

(Palomo y Moraleda, 2014)

Gómez Palomo, S.R. y Moraleda Gil, E.A., (2014). Aproximación a la Ingeniería del Software. Editorial Universitaria Ramón Areces.

(Pérez y Teheran, 2024)

Pérez Rojas, H.U., y Teheran García, M.A., (2024), Hands-On Visual Studio 2022. Packt Publishing.

(Polo y Lamancha, 2012)

Polo Usaola, M., Pérez Lamancha, B., Reales Mateo, P., (2012). Técnicas combinatorias y de mutación para testing de sistemas de software. RA-MA.

(Pressman y Maxim, 2021)

Pressman, R.S. y Maxim, B.R., (2021). Ingeniería de Software. Un enfoque práctico. McGraw Hill.

(Requisitos, WEB, 2024)

Requisitos de sistema de .NET Framework, (2024), Microsoft.com. Disponible en:

<https://learn.microsoft.com/es-es/dotnet/framework/get-started/system-requirements> (Consultado: 2024).

(Sánchez, 2003)

Sánchez Pineda de las Infantas, M.T., (2003). Procesos de elaboración de alimentos y bebidas. Mundi-Prensa.

(Serialización, WEB, 2024)

Serialización (Visual Basic), (2024), Microsoft.com. Disponible en: <https://learn.microsoft.com/es-es/dotnet/visual-basic/programming-guide/concepts/serialization/> (Consultado: 2024).

(Siemens, 2024)

Disponible en: <https://plm.sw.siemens.com/en-US/tecnomatix/products/plant-simulation-software/> (Consultado: 2024).

(Sipper y Bultin, 1998)

Sipper, D. y Bultin, R.L., (1998). Planeación y control de la producción. McGraw Hill.

(tacore, 2024)

Disponible en: <https://tacore.com/> (Consultado: 2024).

(Weiss, 2013)

Weiss, M.A., (2013). Estructuras de datos en Java. Pearson Educación, S.A.

(Wirfs.Brock, 1993)

Wirfs-Brock, R., (1993). Designing Scenarios: Making the Case for a Use Case Framework. Smalltalk Report SIGS Publications. Disponible en: <https://wirfs-brock.com/PDFs/Designing%20Scenarios.pdf>.

Glosario

Fabricante-integrador: Empresa o entidad dedicada al diseño, fabricación y ensamblaje de maquinaria industrial, así como de líneas de producción formadas por las propias máquinas que fabrica y/u otras máquinas compradas a un tercero. Este tipo de empresas ofrecen un enfoque integral, desde la creación a medida del equipo hasta la implementación y puesta en marcha en la planta del cliente final.

DraftLine: Nombre dado a la aplicación cuyo proceso de desarrollo se detalla en el presente documento.

Presupuesto ágil: Presupuesto rápido (en tiempo), flexible y dinámico (en su capacidad de adaptarse a cambios en los requerimientos), y mínimamente detallado para aportar la suficiente información al cliente en la toma de decisiones antes de solicitar un presupuesto tradicional completamente detallado.

Línea de producción: Conjunto de máquinas o equipos que, conectados entre sí, realizan una generación de valor en una industria, bien sea transformando un producto, ensamblándolo u otros, de manera secuencial.

Máquina: Conjunto de partes o componentes vinculados entre sí, asociados para una aplicación determinada, o agrupación de los mismos, que conforman un nodo que realiza una funcionalidad. Por ejemplo: “Lavadora de Envases LAV1000 Marca Tacore”: Máquina que lava envases, con sus características particulares.

Familia de máquinas: Clasificación de máquinas. Agrupa las máquinas que compartan una misma funcionalidad. Por ejemplo: Familia de máquinas: “Lavadora de envases” – Máquinas: Todas aquellas máquinas cuya funcionalidad sea lavar envases.

Modelo FURPS+: Marco de referencia para clasificar y analizar los requisitos de un sistema, desglosándolos en cinco categorías principales: funcionalidad (Functional), facilidad de uso (Usability), fiabilidad (Reliability), rendimiento (Performance) y soporte (Supportability), además de otros requisitos adicionales (+) (Larman, 2002).



**Herramienta para el
dimensionamiento de
líneas de producción en la
industria conservera**

Miguel Fariña Cores - 2024

Versión de la aplicación: 1.0

Versión del Manual: 1.0

DRAFTLINE

Manual de usuario

Índice del manual de usuario

Índice del manual de usuario	97
A.1 Introducción	98
A.2 Requerimientos del sistema	98
A.3 Instalación	99
A.4 Interfaz del usuario	100
A.5 Crear y gestionar borradores de presupuestos de líneas de producción	113
A.6 Crear y gestionar catálogo de máquinas	116
A.7 Solución de problemas comunes	119
A.8 Soporte técnico y actualizaciones	120
A.9 Glosario de términos del manual de usuario	120

A.1 Introducción

DraftLine es una aplicación que permite al usuario, generalmente un integrante de un departamento de ventas de una empresa de fabricación e integración de líneas de producción en la industria conservera, generar borradores de presupuestos rápidos, claros y ágiles con el fin de entregar una respuesta inmediata a las solicitudes de los clientes; todo ello de manera muy visual, con un entorno gráfico de usuario sencillo y claro y un catálogo de datos para almacenar la información sobre las familias de máquinas que escogerá el usuario y cuya selección será optimizada por la propia aplicación según las condiciones exigidas.

Permite una gran flexibilidad y rapidez para adaptarse a todas las variaciones que los comunes cambios en los requerimientos del cliente requieren en estas primeras etapas de un proyecto de integración de una, o varias, líneas de producción.

El resultado generado por la aplicación será un borrador de presupuesto previo de un listado de maquinaria que la empresa usuaria de la aplicación fabricará o comprará para integrar en la planta industrial del cliente, ofreciéndole a éste una pronta lista de las características y requerimientos de esta línea. Esta información le será muy útil al cliente para integrarla en un nivel superior de su proyecto general.

A.2 Requerimientos del sistema

- Sistema operativo: Microsoft Windows 11, 10, 8.1 o 7 SP1 (Compatibilidad con .NET Framework 4.7.2)
- Procesador Intel Pentium / AMD Athlon o equivalente a 2Ghz o superior
- Memoria RAM: 1GB
- Espacio en disco: 50MB libres
- Resolución de pantalla: 1027x768 píxeles

A.3 Instalación

Para comenzar la instalación de la aplicación, haga doble click en el archivo DraftLine_Instalador.exe.

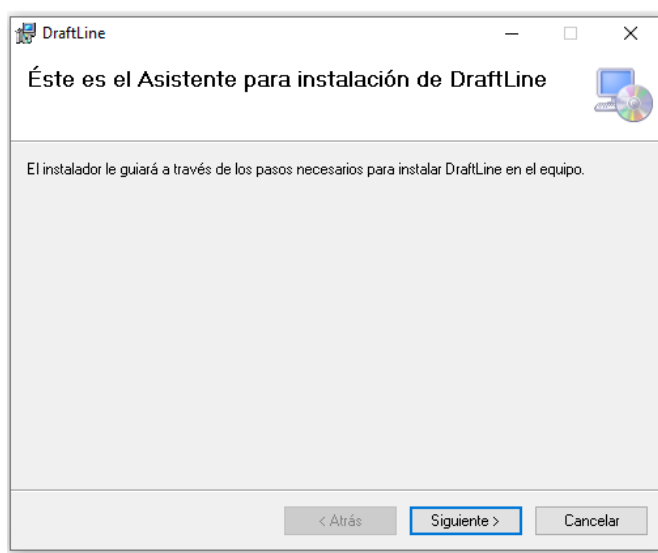


Figura A.1. Pantalla bienvenida de la instalación.

Haga click en Siguiente e indique una dirección para la carpeta de instalación, así como si quiere instalar la herramienta para el usuario actual o para todos los usuarios.

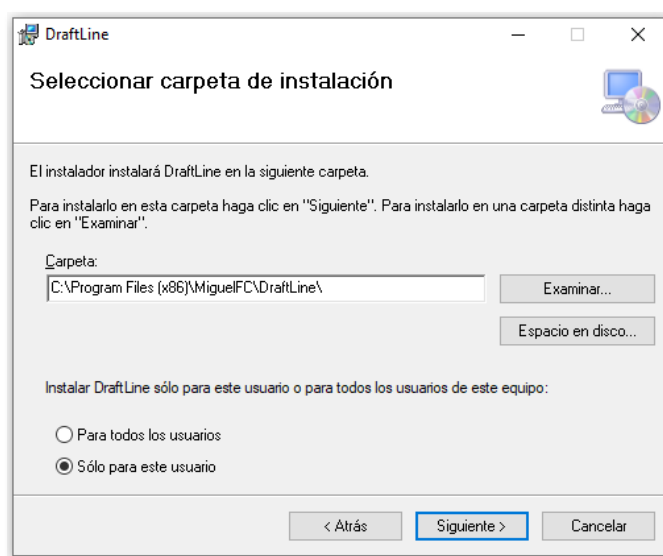


Figura A.2. Selección carpeta de instalación.

Finalmente haga click en siguiente y espere unos momentos mientras se realiza la instalación

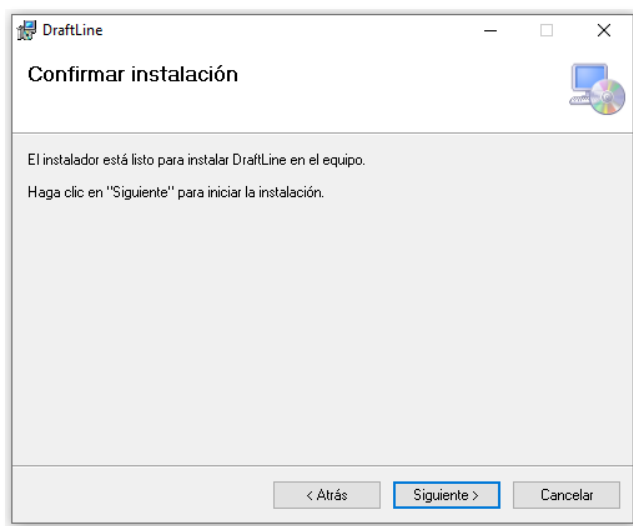


Figura A.3. Confirmación de la instalación.

Windows le preguntará si desea permitir que la aplicación realice cambios en su equipo. Acepte y, finalmente, cierre la ventana que le indica que la instalación ha sido completada.

Encontrará un acceso directo a la aplicación DraftLine en su escritorio

A.4 Interfaz del usuario

A.4.1 Pantalla principal

Cuando inicie DraftLine se encontrará con la pantalla principal que contiene un menú principal, una sección de trabajo y una barra de estado.

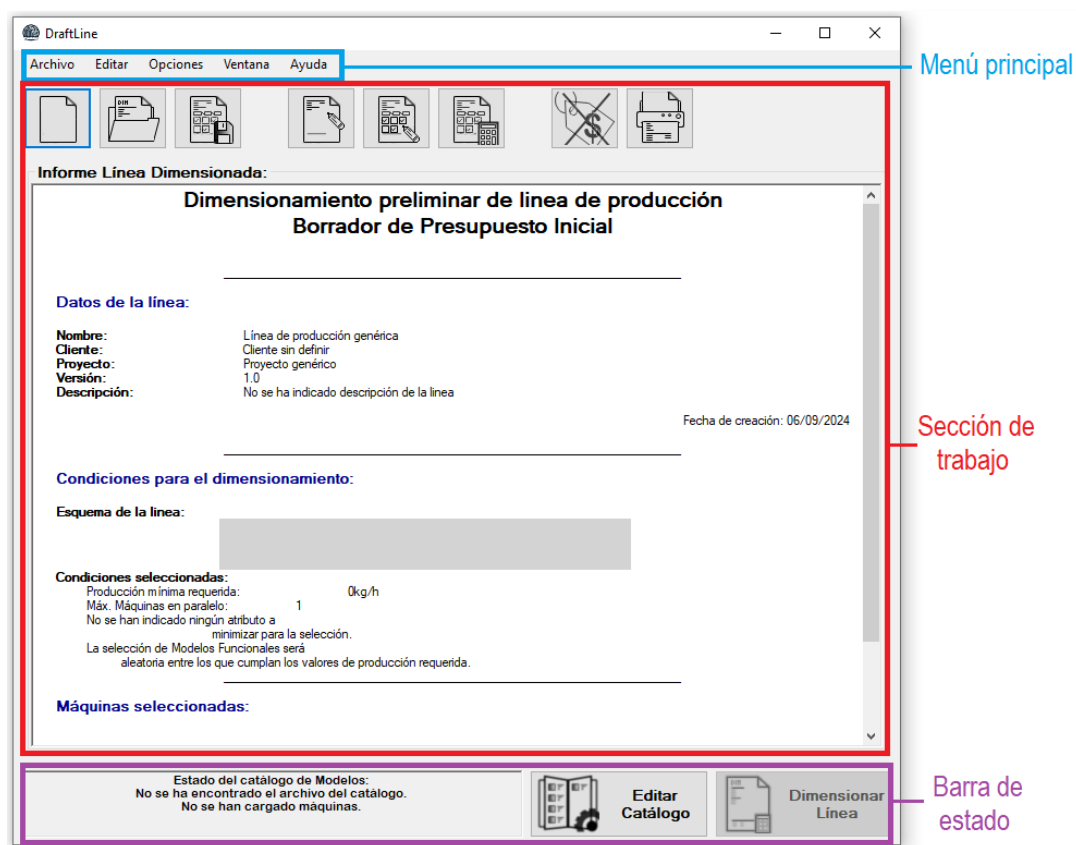


Figura A.4. Componentes pantalla principal.

A.4.1.1 Menú principal

Este menú contiene 5 elementos de menú con las siguientes opciones clasificadas según:



Figura A.5. Menú principal.

- **Archivo:**
 - Nueva línea a dimensionar: Crea una nueva línea vacía con los valores por defecto.
 - Importar línea dimensionada: Abrir una línea previamente guardada en fichero
 - Exportar línea dimensionada: Guardar en fichero la actual línea dimensionada
 - Guardar catálogo: Guardar los cambios en el catálogo actual
 - Importar catálogo: Abrir un catálogo de máquinas guardado en fichero.
 - Exportar catálogo: Guardar el catálogo de máquinas actual en un fichero.
 - Imprimir informe: Muestra la ventana de configuración de impresión del sistema para imprimir el informe de la línea dimensionada actual.

- Salir: Cerrar la aplicación. Si hay cambios pendientes de guardado en el catálogo de máquinas, solicitará confirmación antes de salir.
- **Editar:**
 - Editar datos Informe: Abre ventana para la introducción de los datos del informe.
 - Editar selección familias de máquinas línea: Abre ventana para seleccionar y disponer las familias de máquinas que conformarán la línea a dimensionar.
 - Editar condiciones de dimensionamiento: Abre ventana para la introducción de las condiciones de dimensionamiento de la línea actual.
 - Nueva familia de máquinas: Abre ventana para crear una nueva familia de máquinas en el catálogo actual.
 - Nueva máquina: Si se ha seleccionado una familia de máquinas, abre ventana para crear una nueva máquina que pertenecerá a esa familia de máquinas.
 - Editar familia o máquina seleccionada: Muestra la ventana “Modificar familia de máquinas” o “Modificar máquina”, según el elemento a modificar sea uno u otro, para permitir editar sus propiedades.
 - Eliminar familia o máquina seleccionada: Si se ha seleccionado una familia de máquinas o una máquina, la elimina del catálogo actual.
 - Eliminar todas las familias y máquinas: Vacía el catálogo actual, eliminando todas las familias de máquinas, y sus máquinas, presentes.
- **Opciones:**
 - Visualización del valor de coste: Permite seleccionar la opción de que se muestre, o no, la propiedad valor de coste tanto en el visor del catálogo como en el informe de la línea dimensionada.
 - Mostrar.
 - Ocultar.
- **Ventana:**
 - Configuración catálogo: Muestra el modo “Configurar catálogo” en la sección de trabajo de la pantalla principal.
 - Dimensionamiento de línea: Muestra el modo “Dimensionar Línea” en la sección de trabajo de la pantalla principal.

- **Ayuda:**

- **Manuel de Usuario:** Muestra este manual de usuario en la pantalla principal.
- **Acerca de...:** Muestra la ventana Acerca de... con información de la aplicación.

A.4.1.2 Sección de trabajo

En esta parte de la interfaz se muestra la información de trabajo y los principales botones para realizarlo, en función del modo activado. Puede mostrar tres modos diferentes: “Modo dimensionar línea”, “Modo editar catálogo” y “Modo ayuda”.

Modo “Dimensionar Línea”

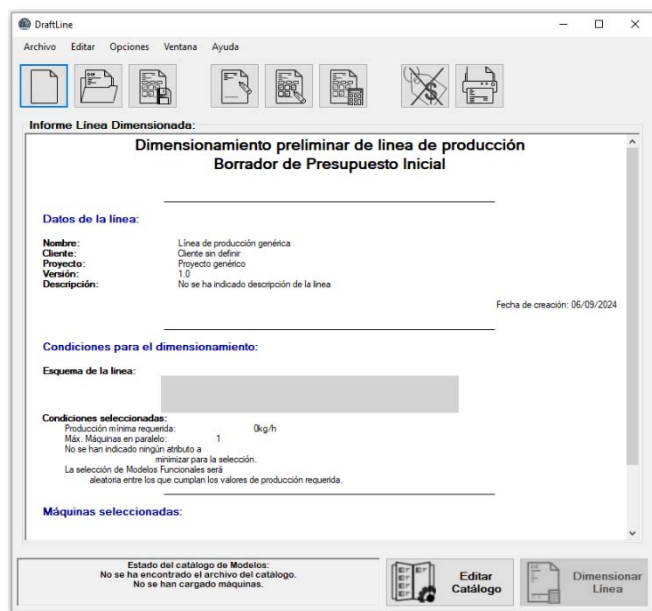


Figura A.6. Pantalla principal en modo dimensionar línea.

Modo de trabajo principal, donde se configurará la línea de producción a dimensionar y se mostrará el informe con los datos finales para su guardado o impresión. Ver apartado 5.- Crear y gestionar borradores de presupuestos de líneas de producción.

En este modo la interfaz muestra los elementos “Botones acceso rápido a herramientas” y “Visor Informe de línea dimensionada”.

Botones acceso rápido a herramientas para este modo:



Figura A.7. Botones acceso rápido en modo dimensionar línea.

- Nueva línea a dimensionar: Crea una nueva línea vacía con los valores por defecto.
- Importar línea dimensionada: Abrir una línea previamente guardada en fichero.
- Exportar línea dimensionada: Guardar en fichero la actual línea dimensionada.
- Editar datos informe: Abre ventana para la introducción de los datos del informe.
- Editar selección familias de máquinas línea: Abre ventana para seleccionar y disponer las familias de máquinas que conformarán la línea a dimensionar.
- Editar condiciones de dimensionamiento: Abre ventana para la introducción de las condiciones de dimensionamiento de la línea actual.
- Visualización del valor de coste: Permite seleccionar la opción de que se muestre, o no, la propiedad valor de coste tanto en el visor del catálogo como en el informe de la Línea dimensionada.
- Imprimir informe: Muestra la ventana de configuración de impresión del sistema para imprimir el informe de la línea dimensionada actual.

Visor informe línea dimensionada

Se muestra la información de la línea que se está dimensionada, en formato informe con la siguiente información:

- Título del informe.
- Datos de la línea: Editables en “Editar datos informe”.
- Condiciones para el dimensionamiento:
 - Esquema de la línea: Editable en “Editar selección familias de máquinas línea”.
 - Condiciones seleccionadas: Editable en “Editar condiciones de dimensionamiento”.
- Máquinas seleccionadas: Aquí se mostrará el listado de máquinas seleccionadas por la aplicación en función del esquema de familias de máquinas y de las condiciones especificadas por el usuario.

- Resumen características de línea dimensionada. Aquí se mostrarán las características de la línea conformada por las máquinas seleccionadas por la aplicación.

Informe Línea Dimensionada:

Dimensionamiento preliminar de línea de producción
Borrador de Presupuesto Inicial

Datos de la línea:

Nombre: Línea de producción genérica
 Cliente: Cliente sin definir
 Proyecto: Proyecto genérico
 Versión: 1.0
 Descripción: No se ha indicado descripción de la línea

Fecha de creación: 06/06/2024

Condiciones para el dimensionamiento:

Esquema de la línea:

Condiciones seleccionadas:
 Producción mínima requerida: 0 kg/h
 Mx. Modelos Funcionales en paralelo: 0
 No se han indicado ningún atributo a minimizar para la selección.
 La selección de Modelos Funcionales será aleatoria entre los que cumplan los valores de producción requerida.

Modelos seleccionados:

Resumen características de línea dimensionada:

Producción máxima: 0 kg/h
 Potencia eléctrica total: 0 kw
 Operarios necesarios: 0 px

Valor de Coste de la línea: 0 €
 Valor de Venta de la línea: 0 €

Informe generado por la aplicación DimLinPro. 1.0.0.0

Figura A.8. Visor informe línea dimensionada.

Modo “Editar catálogo”

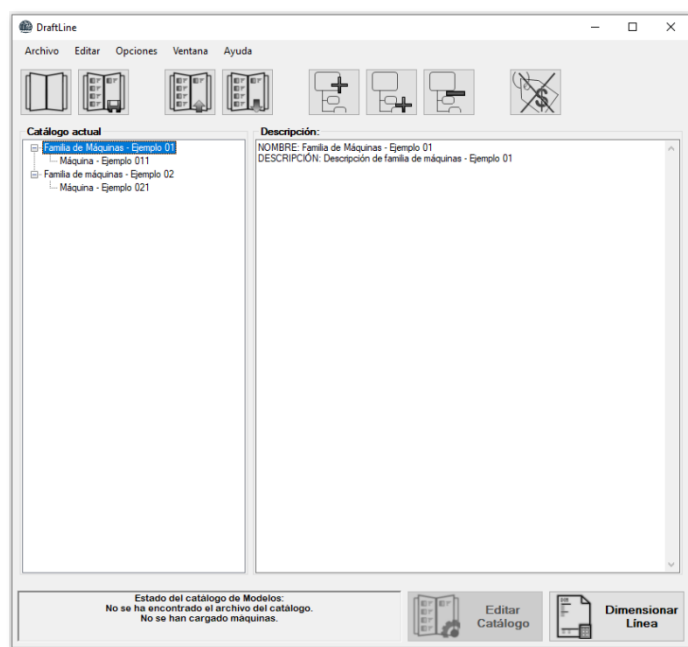


Figura A.9. Pantalla principal en modo editar catálogo.

Modo que permite el trabajo previo necesario al inicio del uso de la aplicación. Desde aquí se podrán ver, editar, importar y exportar las familias de máquinas que posteriormente se utilizarán para el diseño y dimensionamiento de las líneas de producción.

En este modo la interfaz muestra los siguientes elementos:

Botones acceso rápido a herramientas para este modo:



Figura A.10. Botones acceso rápido modo editar catálogo.

- Nuevo catálogo: Borra todos los elementos del catálogo actual.
- Guardar catálogo: Guarda el estado del catálogo actual en el catálogo local.
- Importar catálogo: Carga en el catálogo actual un catálogo previamente exportado a un fichero de datos.
- Exportar catálogo: Exporta el catálogo actual al fichero de datos especificado por el usuario.
- Nueva familia de máquinas: Muestra la ventana “Nueva familia de máquinas” para permitir añadir una nueva familia de máquinas al catálogo actual.
- Nueva máquina: Muestra la ventana “Nueva máquina” para permitir añadir una nueva máquina a la familia de máquinas seleccionada del catálogo actual.
- Eliminar: Elimina del catálogo actual la familia de máquinas o máquina seleccionada.
- Visualización del valor de coste: Permite seleccionar si se desea mostrar u ocultar el valor de coste, tanto de cada máquina como del total de la línea dimensionada, en la aplicación y en el informe impreso.

Visor catálogo

Formado por dos ventanas: “Catálogo” y “Descripción”. En la ventana izquierda se muestran, en forma de árbol, las familias de máquinas y, para cada una de ellas, sus máquinas, existentes en el catálogo actual. En la ventana derecha se mostrará la información del elemento seleccionado en la ventana izquierda.

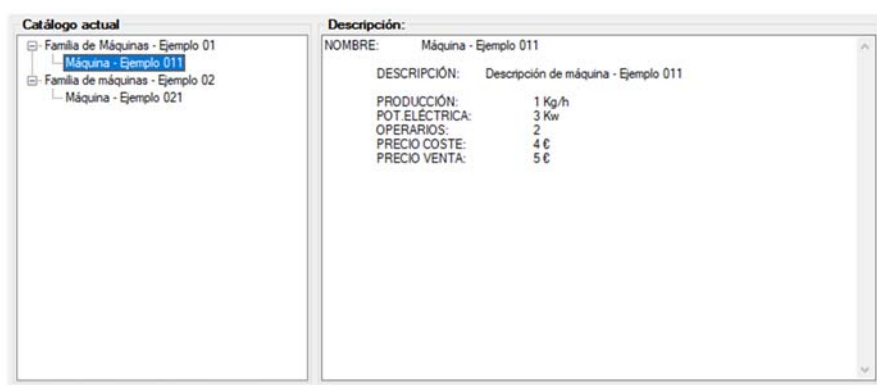


Figura A.11. Visor catálogo.

Modo “Ayuda”

Visor que muestra este manual de usuario

Barra de estado



Figura A.12. Barra de estado de la pantalla principal.

En esta parte de la pantalla principal se muestra una caja de texto informando del estado del catálogo actual y dos botones de acceso rápido a los modos de trabajo “editar catálogo” y “dimensionar línea”.

A.4.2 Ventana “Datos de la línea a dimensionar”

En esta ventana el usuario podrá editar los campos de datos de la línea que está dimensionando. Estos datos se utilizan únicamente como información para mostrar en el informe de la línea.

Datos de la línea a Dimensionar

Nombre de la Línea:
Línea de producción genérica

Cliente: Proyecto: Versión:
Cliente sin definir Proyecto genérico 1.0

Descripción:
No se ha indicado descripción de la línea

✓ ✕

Figura A.13. Ventana “Datos de la línea a dimensionar”.

A.4.3 Ventana “Selección de familias de máquinas de la línea a dimensionar”

En esta ventana, el usuario podrá diseñar la línea, a nivel familias de máquinas, añadiendo familias, de las disponibles en el catálogo actual, al panel del esquema. Allí podrá reordenarlas arrastrándolas con el ratón a su nueva posición, o eliminarlas del esquema con el menú *eliminar* que aparecerá al hacer click derecho sobre la representación de la familia de máquinas a eliminar.

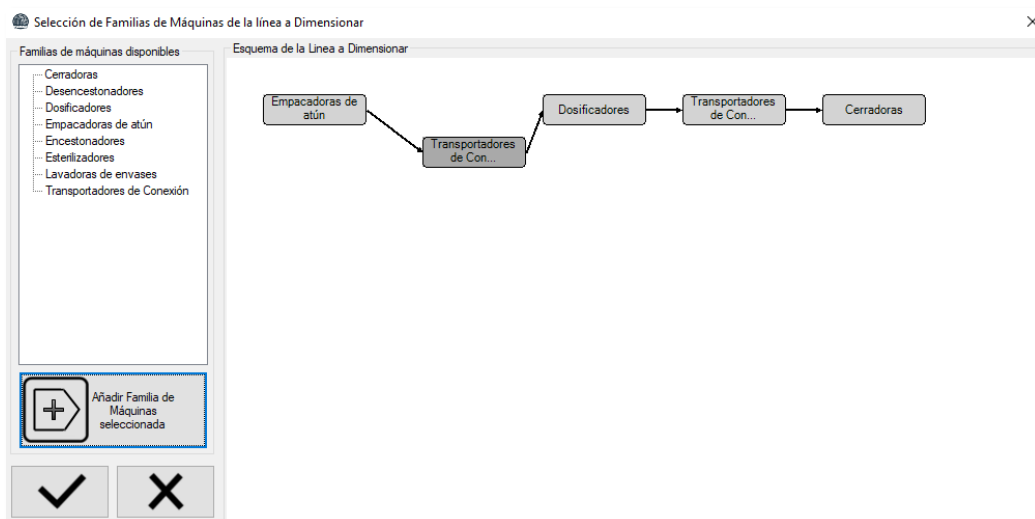


Figura A.14. Ventana “Selección de familias de máquinas de la línea a dimensionar.

A.4.4 Ventana "Condiciones para el dimensionamiento"

En esta ventana se definirán las condiciones para que la aplicación seleccione las Máquinas adecuadas. Se debe indicar:

- Producción requerida: La producción mínima que debe cumplir la línea una vez dimensionada.
- Optimizar por: Seleccionar qué parámetro será el que se deba minimizar a la hora del dimensionamiento como objetivo primario. El objetivo secundario se tendrá en cuenta en caso de que varias opciones cumplan con el primario.
- Paralelismo de máquinas. En el caso de que ninguna de las máquinas disponibles, para una familia de máquinas seleccionada, cumpla con la producción requerida mínima, la aplicación tomará varias unidades cuya suma de producción sí cumpla la condición. La cantidad de unidades máximas a tomar será este valor.

Figura A.15. Ventana "Condiciones para el dimensionamiento".

A.4.5 Ventana "Nueva familia de máquinas"

El usuario podrá introducir los datos para la creación de una nueva familia de máquinas en esta ventana.

Figura A.16. Ventana "Nueva familia de máquinas".

A.4.6 Ventana “Nueva máquina”

El usuario podrá introducir los datos para la creación de una nueva máquina en esta ventana. Previamente deberá seleccionar una familia de máquinas a la que pertenecerá esta nueva máquina.

Figura A.17. Ventana “Nueva máquina”.

A.4.7 Ventana “Editar”

El usuario podrá editar los campos de características de una familia de máquinas, o una máquina, en particular, desde esta ventana, que mostrará los campos necesarios según el elemento que se trate de editar.

Figura A.18. Ventanas “Editar”.

A.4.8 Ventana "Importar"

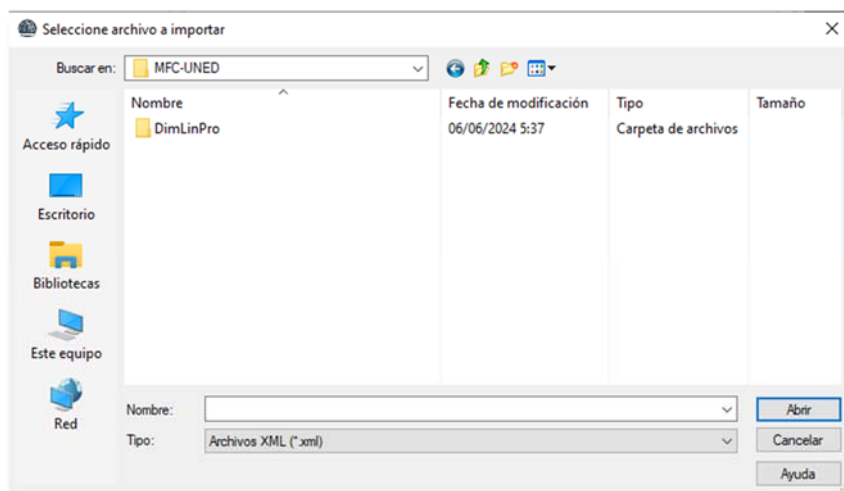


Figura A.19. Ventana "Importar".

Permite la selección del fichero de datos a importar.

A.4.9 Ventana "Exportar"

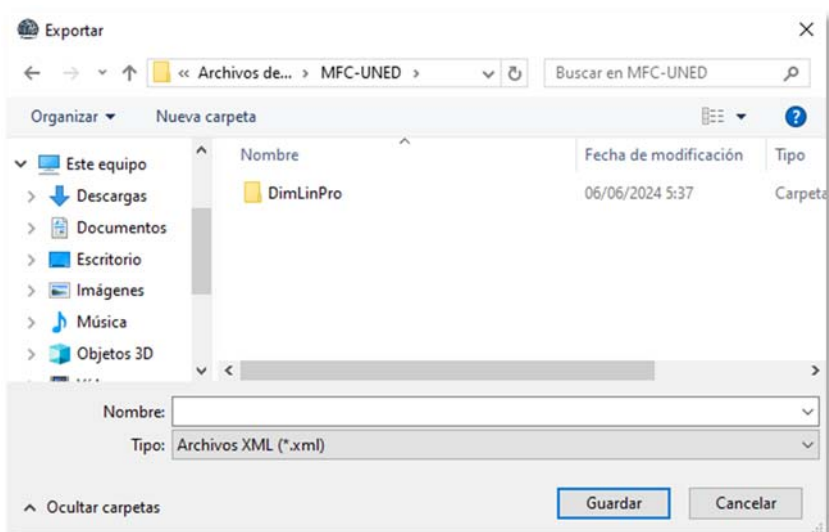


Figura A.20. Ventana "Exportar".

Permite indicar la ubicación y nombre del fichero a exportar

A.4.10 Ventana “Imprimir”

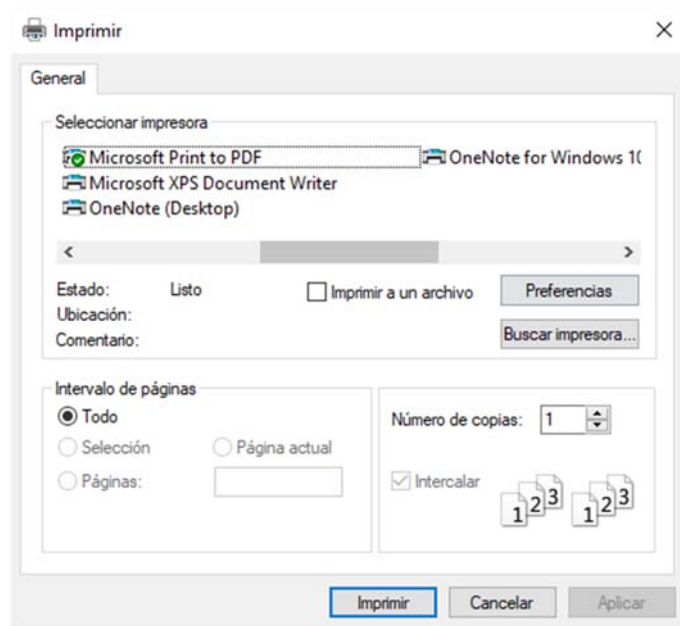


Figura A.21. Ventana “Imprimir”.

Permite seleccionar y configurar la impresora donde se imprimirá el informe generado.

A.4.11 Ventana “Acerca de...”



Figura A.22. Ventana “Acerca de...”

Muestra información de la aplicación.

A.5 Crear y gestionar borradores de presupuestos de líneas de producción

DraftLine permite diseñar y dimensionar líneas de producción, en función del catálogo de máquinas disponibles y de las condiciones especificadas, generando finalmente informes de borradores de presupuesto. Es posible crear una nueva línea, exportarla, importar líneas creadas anteriormente e incluso modificarlas, desde el modo “dimensionar línea” de la pantalla principal.

A.5.1 Crear nueva Línea

Al iniciarse la aplicación, se comienza con una nueva línea preparada para su diseño y dimensionamiento. También se puede iniciar una **nueva línea** al pulsar sobre el botón correspondiente () o en el menú “Archivo>Nueva línea a dimensionar”. Tenga en cuenta que al iniciar una nueva línea se perderá la configuración de la línea actual si no se ha exportado.

A continuación, se deberán editar los **datos de la línea** que se está dimensionando () o “Editar>Editar datos informe” en el menú principal. Ver: Ventana “Datos de la línea a dimensionar”.

Estos datos incluyen:

- **Nombre** de la línea.
- **Cliente** destinatario del presente borrador de presupuesto.
- **Proyecto** al que pertenece la línea a dimensionar.
- **Versión**.
- **Descripción** y otros datos de importancia sobre el informe y la línea que se está dimensionando.

Estos datos se mostrarán al inicio del informe sobre la línea a dimensionar.

El siguiente paso será diseñar el esquema de familias de máquinas que conformará la línea deseada. Para ello pulse sobre el botón correspondiente () o en el menú “Editar>Editar selección familia de máquinas línea”. Se abrirá la Ventana correspondiente Ver: Ventana “Selección de familias de máquinas de la línea a dimensionar”. En esta ventana se podrá elegir, en el visor izquierdo, la familia de máquinas que se desea añadir a la línea, seleccionándola y pulsando en el botón añadir: (). En el panel derecho se irá construyendo un esquema con representaciones de las familias seleccionadas y su orden, indicado por flechas que las conectan. Podrá modificar el esquema mediante la actuación sobre las representaciones directamente:

- Cambiar posición de un elemento:

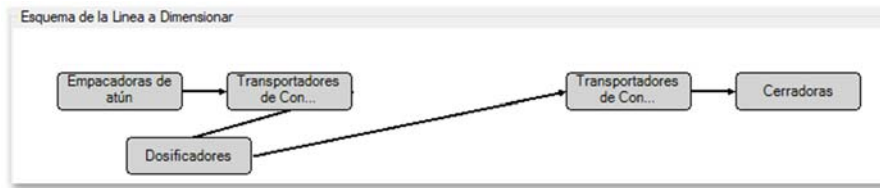


Figura A.23. Cambio orden de familias de máquinas seleccionadas al arrastrar una de ellas.

Pulse con el botón izquierdo del ratón sobre el elemento que desea reubicar y, sin soltar, arrástrelo a la posición deseada, soltando allí el botón. Automáticamente los demás elementos se reordenarán.

- Eliminar una familia del esquema

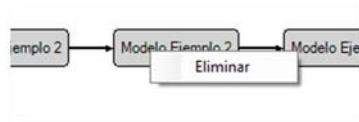


Figura A.24. Botón eliminar (botón derecho de ratón sobre representación).

Pulse con el botón derecho del ratón sobre la representación de una familia y se mostrará el menú eliminar de ese elemento. Púlselo y se eliminará de la selección. Además, al pasar el ratón por encima de una representación, se mostrará un mensaje con información sobre esa familia de máquinas.

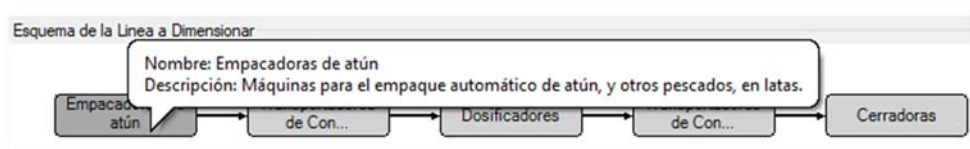


Figura A.25. Información de una familia de máquinas seleccionada al pasar el ratón por encima.

Este esquema así generado se mostrará en el Informe bajo el título “Esquema de la línea”, en la sección “Condiciones para el dimensionamiento”.

El último paso, aunque no necesariamente en este orden, será la indicación de las condiciones de dimensionamiento de la línea, que permitirán a la aplicación realizar la selección de las máquinas, pertenecientes a las familias de máquinas escogidas, que cumplan dichas condiciones. Para ello, pulse sobre el botón correspondiente () o en el menú “Editar>Editar condiciones de dimensionamiento”.

En la ventana que se muestra (Ver: Ventana “Condiciones para el dimensionamiento”) deberá indicar:

- Producción requerida: La producción mínima que debe cumplir la línea una vez dimensionada.
- Optimizar por: Seleccionar qué parámetro será el que se deba minimizar a la hora del dimensionamiento como objetivo primario. El objetivo secundario se tendrá en cuenta en el caso de que varias opciones cumplan con el primario.
- Paralelismo de máquinas: En caso de que ninguna máquina disponible para una familia de máquinas seleccionada cumpla con la producción requerida mínima, la aplicación tomará varias unidades cuya suma de producción sí cumpla la condición. La cantidad de unidades máximas a tomar será este valor.

Estas condiciones seleccionadas se mostrarán en el informe de la línea, bajo el título “Condiciones seleccionadas” de la sección “Condiciones para el dimensionamiento”.

El informe sobre la línea (el borrador de presupuesto inicial) irá autocompletándose a medida que los tres pasos anteriores se vayan realizando. En cualquier momento puede volver a uno de ellos y editar los datos para modificar la línea dimensionada. Finalmente, puede guardar la línea dimensionada (ver apartado siguiente: Exportar línea) o imprimirla en una de las impresoras (pdf, físicas, etc.) de su sistema (ver apartado A.5.5.- Imprimir borrador de presupuesto de línea).

A.5.2 Exportar Línea

Para exportar una línea dimensionada, pulse el botón correspondiente () o en el menú “Archivo>Exportar línea dimensionada”. En la ventana que se abrirá a continuación (Ver: Ventana “Exportar”) indique un nombre y una ubicación para guardar allí la línea actual y pulse aceptar.

Nota: El archivo generado contiene una copia de las familias de máquinas utilizadas por la línea, para permitir recuperar dicha línea en un futuro aun cuando esas máquinas ya no existan en el catálogo de la aplicación en ese momento.

A.5.3 Importar Línea

Para importar una línea dimensionada, pulse el botón correspondiente () o en el menú “Archivo>Importar línea dimensionada”. En la ventana que se abrirá a continuación (Ver: Ventana “Importar”) seleccione el fichero que contiene la línea a importar y pulse aceptar. Si la línea a importar contiene familias de máquinas que ya no existen en el catálogo actual de la aplicación, se le pedirá confirmación para añadirlos al catálogo

actual. Esta confirmación será necesaria para la importación de la línea. Si no se acepta, no se importará la línea.

Nota: Tenga en cuenta que al importar una línea se perderá la configuración de la línea actual si no se ha exportado previamente.

A.5.4 Modificar línea

Tanto si se ha dimensionado una línea nueva, como si se ha importado una, es posible modificarla editando los valores que se indicaron para su creación:

- Editar datos de la línea .
- Editar selección de familias de máquinas .
- Editar condiciones de dimensionamiento .

A.5.5 Imprimir borrador de presupuesto de línea

Es posible imprimir el contenido del informe en cualquier momento, pulsando en el botón correspondiente () o en el menú: “Archivo>Imprimir”. Se abrirá una nueva ventana para seleccionar y configurar la impresora destino (Ver: Ventana “Imprimir”).

A.6 Crear y gestionar catálogo de máquinas

Para el funcionamiento de la aplicación será necesario disponer de las familias de máquinas y sus respectivas máquinas en el catálogo actual. Para ello se dispone del “Modo editar catálogo” de la pantalla principal. Para entrar en este modo, pulse el botón correspondiente ( Editar Catálogo) o en el menú “Ventana>Configuración catálogo”.

A.6.1 Crear un nuevo catálogo

Para crear un catálogo nuevo, eliminando cualquier familia de máquinas que pudiese estar en el catálogo actual, pulse el botón correspondiente () o en el menú “Editar>Eliminar todos los modelos”.

A.6.2 Nueva familia de máquinas

Tanto si el catálogo actual está vacío como si ya tiene máquinas, se podrá añadir una nueva familia de máquinas pulsando el botón correspondiente () o en el menú “Editar>Nueva familia de máquinas” o en el elemento “Añadir nueva familia de máquinas” del menú contextual que aparece al pulsar con el botón derecho del ratón sobre cualquier parte del visor izquierdo (“Catálogo actual”). Introduzca el nombre y descripción de la familia de máquinas a crear en la ventana que se abre (Ver: Ventana “Nueva familia de máquinas”) y pulse aceptar. La nueva familia de máquinas aparecerá en el visor izquierdo del modo editar catálogo.

Nota: Tenga en cuenta que el nombre de la familia de máquinas debe ser único, la aplicación no permite que dos familias de máquinas distintas tengan el mismo nombre.

A.6.3 Nueva máquina

Las máquinas deben pertenecer a una familia de máquinas, por lo que para crear una nueva máquina deberá previamente seleccionar una familia de máquinas en el visor izquierdo (Catálogo actual) y, a continuación, pulsar el botón correspondiente () o en el menú “Editar>Nuevo máquina”, o en el botón “Añadir máquina” del menú contextual que aparecerá al pulsar con el botón derecho del ratón sobre una familia de máquinas seleccionada. Introduzca el nombre, descripción y valores de las propiedades de la máquina a crear en la ventana que se abre (Ver: Ventana “Nueva máquina”) y pulse aceptar. La nueva máquina aparecerá en el visor izquierdo del modo Editar catálogo, bajo la familia de máquinas correspondiente.

A.6.4 Editar familia o máquina seleccionada

Es posible editar cualquier elemento existente en el catálogo mediante su selección y, a continuación, seleccionar en el menú “Editar>Editar familia o máquina seleccionada” o en el botón “Editar elemento seleccionado” del menú contextual que aparecerá al pulsar con el botón derecho del ratón sobre un elemento seleccionado. Se abrirá la ventana de edición (de familia o máquina según corresponda) donde podrá editar sus propiedades y guardarlas pulsando aceptar.

A.6.5 Eliminar familia o máquina seleccionada

Es posible eliminar cualquier elemento del catálogo actual. Para ello seleccione la familia de máquinas o la máquina que desee eliminar y pulse el botón correspondiente () , o en el menú “Editar>Eliminar familia o máquina seleccionada” o en el botón “Eliminar elemento seleccionado” del menú contextual que aparecerá al pulsar con el botón derecho del ratón sobre un elemento seleccionado.

Nota: Tenga en cuenta que, si elimina una familia de máquinas, automáticamente se eliminarán todas las máquinas que le pertenecen.

A.6.6 Guardar catálogo

La aplicación dispone de un mismo catálogo (llamado “catálogo actual” en este manual) que es el que carga en memoria al iniciarse, y sobre el que se trabaja durante el uso de la aplicación. Si se realizan cambios en este catálogo actual, la aplicación lo mostrará en la barra de estado, y consultará al usuario si desea guardar los cambios realizados al salir de la aplicación. Igualmente, se pueden guardar los cambios en cualquier momento mediante el botón correspondiente () o en el menú “Archivo>Guardar catálogo”.

A.6.7 Exportar catálogo

Es posible guardar una copia del catálogo actual en un fichero a elección del usuario, permitiendo así crear copias de respaldo o incluso diferentes catálogos según necesidades. Para ello pulse el botón correspondiente () o en el menú “Archivo>Exportar catálogo”. En la ventana que se abrirá a continuación (Ver: Ventana “Exportar”) indique un nombre y una ubicación para guardar allí el catálogo y pulse aceptar.

A.6.8 Importar catálogo

Si dispone de una copia de respaldo o de varios catálogos almacenados en ficheros de datos diferentes, es posible cargarlos en el catálogo actual. Para ello pulse el botón correspondiente () o en el menú “Archivo>Importar catálogo”. En la ventana que se abrirá a continuación (Ver: Ventana “Importar”) seleccione el fichero donde está alojado el catálogo y pulse aceptar.

Nota: Tenga en cuenta que la carga de un catálogo importado eliminará automáticamente cualquier familia y máquina anteriormente en el catálogo actual.

A.7 Solución de problemas comunes

A.7.1 Mensajes en el cuadro de estado

Error al cargar el catálogo de máquinas: Error en el documento XML

No ha sido posible importar el catálogo desde el fichero seleccionado. El fichero no es correcto o está dañado. Seleccione otro fichero para importar.

No se ha encontrado el archivo del catálogo. No se han cargado máquinas

El fichero que almacena el catálogo actual no se encuentra en el directorio de la aplicación. Pulse guardar catálogo actual () para generar un nuevo fichero.

A.7.2 Otros errores

Error al crear una nueva máquina.

No se ha seleccionado una familia de máquinas al que pertenecerá la nueva máquina. Seleccione una familia de máquinas y vuelva a intentarlo.

Error al eliminar una familia o máquina

No se ha seleccionado ninguna familia de máquinas o máquina para eliminar. Seleccione un elemento y vuelva a intentarlo.

Error al editar una familia o máquina

No se ha seleccionado ninguna familia de máquinas o máquina para editar. Seleccione una y vuelva a intentarlo.

Error al seleccionar familia de máquinas para esquema de línea. No hay ningún elemento para seleccionar

No hay familias de máquinas disponibles en el catálogo actual. Cree nuevas familias de máquinas o importe un catálogo nuevo y vuelva a intentarlo.

Error al añadir familia de máquinas para esquema de línea

No ha seleccionado ninguna familia de máquinas. Seleccione una familia de máquinas y vuelva a intentarlo.

A.8 Soporte técnico y actualizaciones

Para solicitar soporte técnico o consultar posibles actualizaciones, póngase en contacto con:

draftline_soporte@outlook.es

A.9 Glosario de términos del manual de usuario

Máquina: Conjunto de partes o componentes vinculados entre sí, asociados para una aplicación determinada, o agrupación de los mismos, que conforman un nodo que realiza una funcionalidad. Por ejemplo: “Lavadora de Envases LAV1000 Marca Tacore”: Máquina que lava envases, con sus características particulares.

Familia de máquinas: Clasificación de máquinas. Agrupa las máquinas que compartan una misma funcionalidad. Por ejemplo: Familia de máquinas: “Lavadora de envases” – Máquinas: Todas aquellas máquinas cuya funcionalidad sea lavar envases.

Línea de producción: Conjunto de máquinas o equipos que, conectados entre sí, realizan una generación de valor en una industria, bien sea transformando un producto, ensamblándolo u otros, de manera secuencial.

ANEXO B CÓDIGO FUENTE

B.1 Clase FormPrincipal

```
Imports System.IO

''' <summary>
''' Esta clase es el formulario base de la aplicación. Se carga al iniciar la aplicación y muestra los
''' modos Editar catálogo, Dimensionar Línea y ayuda, desde donde trabajará el usuario. Contiene los
''' controles (botones, menús, etc.) para acceder a todas las funcionalidades de la aplicación. Desde
''' ella se crean los demás formularios que se le presentarán al usuario. Es este formulario el que
''' tomará los datos de esos formularios para luego utilizar el resto de clases y cumplir los casos
''' de uso definidos.
''' </summary>
Public Class FormPrincipal

    '''<summary> Centinela para saber si se han guardado los cambios en el catálogo actual</summary>
    Private catalogoModificado As Boolean = False
    '''<summary> Gestor de Máquinas (catálogo)</summary>
    Private sMaquinas As ServicioMaquinas
    '''<summary> Gestor de Líneas</summary>
    Private sLineas As ServicioLineas
    '''<summary> Gestor de impresión</summary>
    Private imprimir As Imprimir
    '''<summary> Clase que realiza los cálculos</summary>
    Private dimensionar As Dimensionar
    ''' <summary>Instancia de la clase donde se generará el informe </summary>
    Private informe As Informe
    ''' <summary>Dirección del archivo del catálogo de modelos actual</summary>
    Private catalogoLocal As String = Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"catalogoLocal.xml")
    ''' <summary> Booleano para definir si se quiere mostrar o no el valor de coste </summary>
    Private valorCosteVisible As Boolean = True
    ''' <summary> Ruta del manual de usuario </summary>
    Private rutaPDF As String = Path.Combine(Application.StartupPath, "ManualUsuario.pdf")
    ''' <summary> Almacena el Formulario para introducir la info de la nueva familia</summary>
    Private formFam As FormFamilia
    ''' <summary> Almacena el Formulario para introducir la info de la nueva máquina</summary>
    Private formMaq As FormMaquina
    ''' <summary> Almacena el Formulario para introducir la info de los datos de la línea</summary>
    Private formDatosLinea As FormDatosLinea
    ''' <summary> Almacena el Formulario para introducir la selección de familias para la línea</summary>
    Private formSelFam As FormSelecFam
    ''' <summary> Almacena el Formulario para introducir las condiciones para el
dimensionamiento</summary>
    Private formCond As FormCondiciones
    ''' <summary> Almacena el Formulario con info sobre esta aplicación</summary>
    Private formAcercaDe As FormAcercaDe

    ''' <summary>
    ''' Método Constructor del formulario principal
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub FormPrincipal_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        sMaquinas = ServicioMaquinas.Instance 'La instancia (única) del servicioMáquinas para gestionar
las máquinas
        sLineas = ServicioLineas.Instance 'La instancia (única) del servicioLineas para gestionar
las líneas
        dimensionar = Dimensionar.Instance 'La instancia (única) de la clase dimensionar, para los
cálculos
        informe = New Informe() 'Inicializo el informe, por defecto
        RichTextBoxEstado.Clear() 'Preparo el RichTextBox donde muestro el estado del catálogo
actual
        RichTextBoxEstado.Font = New Font("Arial Rounded MT Bold", 8.25F)
        RichTextBoxEstado.SelectionAlignment = (HorizontalAlignment.Center)
        CargaCatalogo(catalogoLocal) 'Cargo inicialmente el catálogo persistente propio
        informe.CreaInforme() 'Creo el informe
        RichTextBoxInforme.Rtf = informe.Contenido.Rtf 'Cargo el texto del informe en el richTextBox

    End Sub

    ''' <summary>
    ''' Método que llama a la carga de catálogo desde servicioMaquinas, actualizando, además, el estado
devuelto
    ''' </summary>
    ''' <param name="dirArchivo">Dirección del archivo que contiene el catálogo a cargar</param>
```

```

Private Sub CargaCatalogo(dirArchivo As String)
    RichTextBoxEstado.Clear() 'Limpio el richtextbox de Estado
    RichTextBoxEstado.AppendText("Estado del catálogo de Modelos: " & vbCrLf)
    'Cargo el catálogo desde la dirección y muestro el estado
    RichTextBoxEstado.AppendText(sMaquinas.CargaDesde(dirArchivo))
    CargarDatosEnTreeView() 'Cargo los datos del catálogo en el treeview
End Sub

''' <summary>
''' Método que cambia el Modo de la interfaz a Editar Catálogo, capturando el botón pulsado
''' </summary>
''' <param name="sender">El botón de modo pulsado</param>
''' <param name="e">El click sobre el botón. No usado</param>
Private Sub ModoCatalogo(sender As Object, e As EventArgs) Handles BttEditarCat.Click,
TSMItemEditarCat.Click
    CambiaPanel("modoEditarCatalogo")
End Sub

''' <summary>
''' Método que cambia el Modo de la interfaz a Dimensionar, capturando el botón pulsado
''' </summary>
''' <param name="sender">El botón de modo pulsado</param>
''' <param name="e">El click sobre el botón. No usado</param>
Private Sub ModoDimensionar(sender As Object, e As EventArgs) Handles BttDimens.Click,
TSMItemDimens.Click
    CambiaPanel("modoDimensionar")
End Sub

''' <summary>
''' Método que cambia el Modo de la interfaz a Ayuda, capturando el botón pulsado
''' </summary>
''' <param name="sender">El botón del menú Manual de Usuario</param>
''' <param name="e">El click sobre el botón. No usado</param>
Private Sub ModoAyuda(sender As Object, e As EventArgs) Handles TSMItemManualUsuario.Click
    CambiaPanel("modoAyuda")
End Sub

''' <summary>
''' Método que cambia el panel central en función del modo en que se encuentre la aplicación
''' </summary>
''' <param name="modo">El Modo de la interfaz al que hay que cambiar.</param>
Private Sub CambiaPanel(modo As String)
    PanelEditarCat.Visible = False 'Primero, quito la visibilidad a todo en el panel central:
    PanelDimensionar.Visible = False
    PanelAyuda.Visible = False
    BttEditarCat.Enabled = True 'Activo todos los botones para luego desactivar sólo el que
corresponde
    BttDimens.Enabled = True
    Select Case modo
        Case "modoEditarCatalogo"
            PanelEditarCat.Visible = True 'Hago visible el panel del modo configurar
            BttEditarCat.Enabled = False 'Deshabilito el botón a este panel
            CargarDatosEnTreeView() 'Cargo los datos de la catálogo en el treeview
        Case "modoDimensionar"
            PanelDimensionar.Visible = True 'Hago visible el panel del modo dimensionar
            BttDimens.Enabled = False 'Deshabilito el botón a este panel
        Case "modoAyuda"
            PanelAyuda.Visible = True 'Hago visible el panel del modo Ayuda
            Try
                WebBrowserAyuda.Navigate(rutaPDF) 'Cargo el documento en el navegador
            Catch ex As Exception
                MessageBox.Show("Error al cargar el documento: " & ex.Message)
            End Try
    End Select
End Sub

''' <summary>
''' Método que maneja el evento de cierre del formulario
''' Para controlar si hay cambios en la catálogo y si se desean guardar o no
''' </summary>
''' <param name="sender">Este formulario</param>
''' <param name="e">El cierre del formulario</param>
Private Sub Form_FormClosing(sender As Object, e As FormClosingEventArgs) Handles
MyBase.FormClosing
    If catalogoModificado Then 'Si hay cambios pendientes en el catálogo actual, muestro mensaje
        Dim respuesta As DialogResult = MessageBox.Show("¿Guardar los cambios en el catálogo de
modelos actual?",
"Guardar cambios", MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Warning)
        Select Case respuesta
            Case DialogResult.Yes 'Guardo cambios y cierro
                sMaquinas.GuardarEn(catalogoLocal) 'Guardo el catálogo en el archivo local

```

```

        Case DialogResult.No      'No guardo cambios y cierro
        Case Else                 'No guardo ni cierro (abort, cancel, ...)
            e.Cancel = True       'Cancelo el formClosing
        End Select
    End If
End Sub

'-----
'| Métodos del Modo Editar catálogo |
'-----

''' <summary>
''' Método que actualiza el TreeView para mostrar el listado de máquinas del catálogo
''' </summary>
Private Sub CargarDatosEnTreeView()
    Dim famMaqList As List(Of FamiliaMaquina) = sMaquinas.GetFamilias() 'Creo y cargo la lista de
Familias
    TreeViewCatalogo.Nodes.Clear() 'Limpio el TreeView antes de cargar
los datos
    For Each fam As FamiliaMaquina In famMaqList 'Recorro la lista de familias
        Dim nodoFamilia As New TreeNode(fam.Nombre) 'Creo un nodo con cada familia
        nodoFamilia.Tag = fam 'Asigno los Tag para identificar el tipo de
elemento del nodo
        TreeViewCatalogo.Nodes.Add(nodoFamilia) 'Añado el nodo al TreeView
        For Each maq As Maquina In fam.Maquinas 'Recorro las máquinas de la familia
            Dim nodoMaquina As New TreeNode(maq.Nombre) 'Creo un nodo con la máquina
            nodoMaquina.Tag = maq 'Asigno los Tag para identificar el tipo de
elemento del nodo
            nodoFamilia.Nodes.Add(nodoMaquina) 'Añado el nodo al como hijo del nodo familia al que
pertenece
        Next
    Next
    TreeViewCatalogo.Sort() 'Ordeno alfabéticamente
    TextBoxDescripcion.Clear() 'Limpio el textBox de descripción
End Sub

''' <summary>
''' Método para manejar el menú contextual del botón derecho en el TreeView
''' </summary>
''' <param name="sender">El TreeView</param>
''' <param name="e">El Evento Click del ratón</param>
Private Sub TreeViewCatalogo_MouseClick(sender As Object, e As MouseEventArgs) Handles
TreeViewCatalogo.MouseClick
    If e.Button = MouseButton.Right Then 'Si se pulsó con el
botón derecho
        Dim nodoSeleccionado As TreeNode = TreeViewCatalogo.GetNodeAt(e.Location) 'Tomo el nodo
bajo el cursor
        If nodoSeleccionado Is Nothing Then 'Si no se ha seleccionado
ningún nodo
            TSMenuEliminar.Enabled = False 'Deshabilito eliminar nodo
seleccionado
            TSMenuNuevaMaquina.Enabled = False 'Deshabilito añadir máquina
            TSMenuEditar.Enabled = False 'Deshabilito editar
            ElseIf TypeOf nodoSeleccionado.Tag Is FamiliaMaquina Then 'Si se ha seleccionado una
familia
                TSMenuEliminar.Enabled = True 'Habilito eliminar
                TSMenuNuevaMaquina.Enabled = True 'Habilito añadir máquina
                TSMenuEditar.Enabled = True 'Habilito editar
            ElseIf TypeOf nodoSeleccionado.Tag Is Maquina Then 'Si se ha seleccionando una máquina
                TSMenuEliminar.Enabled = True 'Habilito eliminar
                TSMenuNuevaMaquina.Enabled = False 'Deshabilito añadir maquina
                TSMenuEditar.Enabled = True 'Habilito editar
            End If
            ContextMenuStrip1.Show(TreeViewCatalogo, e.Location) 'Muestro el menú contextual
        End If
    End Sub

''' <summary>
''' Método que captura la selección en el treeview para actualizar correspondientemente
''' la ventana de descripción
''' </summary>
''' <param name="sender">El TreeView</param>
''' <param name="e">La selección de un nodo</param>
Private Sub TreeViewCatalogo_AfterSelected(sender As Object,
e As TreeViewEventArgs) Handles
TreeViewCatalogo.AfterSelect
    Dim nodoSeleccionado As TreeNode = e.Node 'Tomo el nodo seleccionado
    If nodoSeleccionado IsNot Nothing Then 'Si se ha seleccionado algún
nodo
        TextBoxDescripcion.Clear() 'Limpio el texto del textBox

```

```

        If TypeOf nodoSeleccionado.Tag Is FamiliaMaquina Then 'Si es una familia de máquinas
            'llamo al método que muestra los datos de Familia
            TextBoxDescripcion.Text = sMaquinas.GetDatosFamilia(nodoSeleccionado.Text)
        ElseIf TypeOf nodoSeleccionado.Tag Is Maquina Then 'Si es una máquina
            'llamo al método que muestra los datos de maquinas añadiendo el nombre de la fam. a
            la que pertenece
            TextBoxDescripcion.Text = sMaquinas.GetDatosMaquina(nodoSeleccionado.Text,
                                                                nodoSeleccionado.Parent.Text,
                                                                valorCosteVisible)
        End If
    End If
End Sub

''' <summary>
''' Método que abre un nuevo formulario para pedir la información de la nueva familia de máquinas,
con la que
''' podrá crearla
''' </summary>
''' <param name="sender">El botón de crear nueva Familia o la selección correspondiente en el
''' ToolStripMenu</param>
''' <param name="e">El click sobre el botón</param>
Private Sub NuevaFamilia_Click(sender As Object, e As EventArgs) Handles TSMenuNuevaFamilia.Click,
    BttNuevaFamilia.Click,
    TSMItemNuevaFamilia.Click
    formFam = New FormFamilia() 'Creo un nuevo formulario para introducir los datos de la nueva
familia
    If formFam.ShowDialog() = DialogResult.OK Then 'Muestro el formulario y compruebo si el usuario
aceptó
        sMaquinas.CreaFamilia(formFam.Nombre, formFam.Descripcion) 'Creo la nueva Familia
        CargarDatosEnTreeView() 'Vuelvo a cargar el TreeView con los datos
actuales
        catalogoModificado = True 'Establezco el centinela de catálogo
modificado
    End If
End Sub

''' <summary>
''' Método que elimina la familia de máquinas o la máquina seleccionada en el TreeView, capturando
el evento
''' desde el menú contextual del treeview, desde el elemento correspondiente del menú principal,
o desde
''' el botón borrar
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub Borrar_Click(sender As Object, e As EventArgs) Handles TSMenuEliminar.Click,
    BttBorrar.Click, TSMItemBorrar.Click
    catalogoModificado = True 'Establezco el centinela de catálogo
modificado
    Dim nodoSeleccionado As TreeNode = TreeViewCatalogo.SelectedNode 'Tomo el nodo seleccionado
    If nodoSeleccionado IsNot Nothing Then 'Compruebo que hay algo seleccionado
        Dim nombreModelo As String = nodoSeleccionado.Text 'Tomo el Nombre del elemento
seleccionado
        'Si es una Familia, se realiza la consulta correspondiente
        If TypeOf nodoSeleccionado.Tag Is FamiliaMaquina Then
            Dim confirmacion As DialogResult = MessageBox.Show("¿Borrar la Familia de Máquinas "
& nombreModelo &
" y todas sus Máquinas?", "Confirmación", MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
            If confirmacion = DialogResult.Yes Then 'Se confirma la aceptación del borrado y se
llama al método
                'Intento borrar la fam. seleccionada del catálogo en ServicioModelos
                If sMaquinas.DelFamilia(nombreModelo) Then
                    'Mensaje de aviso de que se ha borrado la familia
                    MessageBox.Show("Familia de Máquinas: " & nodoSeleccionado.Text & " y todas
sus máquinas,
borrada.",
"Aviso")
                Else 'Si no se ha podido borrar
                    MessageBox.Show("Error al borrar la Familia de máquinas: " &
nodoSeleccionado.Text, "Error")
                End If
            End If
            'Si es una máquina se realiza la consulta correspondiente
        ElseIf TypeOf nodoSeleccionado.Tag Is Maquina Then
            Dim confirmacion As DialogResult = MessageBox.Show("¿Borrar la Máquina " & nombreModelo
&
vbCrLf & "?", "Confirmación", MessageBoxButtons.YesNo,
MessageBoxIcon.Question)

```

```

        If confirmacion = DialogResult.Yes Then 'Se confirma la aceptación del borrado y se
        llama al método
        Familia
            'Intento borrar la maq. seleccionada de la lista de máquinas en su correspondiente
            'en ServicioModelos
            Dim identificador As Guid = sMaquinas.GetIDFamilia(nodoSeleccionado.Parent.Text)
            Dim fam As FamiliaMaquina = sMaquinas.GetFamiliaID(identificador)
            If sMaquinas.DelMaquina(nombreModelo, fam) Then
                'Mensaje de aviso de que se ha borrado el modelo
                MessageBox.Show("Máquina: " & nodoSeleccionado.Text & " borrada.", "Aviso")
            Else 'Si no se ha podido borrar
                MessageBox.Show("Error al borrar la Máquina: " & nodoSeleccionado.Text,
                "Error")
            End If
        End If
    End If
    CargarDatosEnTreeView() 'Actualizo el TreeView con los datos actuales en ServicioModelos
    sLineas.NuevaLinea() 'Creo línea nueva por si la anterior se ve afectada por los cambios
en el catálogo
End If
End Sub

''' <summary>
''' Abre un nuevo formulario para crear una nueva máquinas y añadirla a la Familia seleccionada
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub NuevaMaquina_Click(sender As Object, e As EventArgs) Handles TSMenuNuevaMaquina.Click,
    BttNuevaMaq.Click,
TSMItemNuevaMaq.Click
    Dim nodoSeleccionado As TreeNode = TreeViewCatalogo.SelectedNode 'Tomo el nodo seleccionado
    'Si hay un nodo familia seleccionado, lo casteo a familia
    If nodoSeleccionado IsNot Nothing AndAlso TypeOf nodoSeleccionado.Tag Is FamiliaMaquina Then
        Dim fam As FamiliaMaquina = DirectCast(nodoSeleccionado.Tag, FamiliaMaquina)
        formMaq = New FormMaquina(fam) 'Abro el formulario de crear nueva Máquina, con la familia
seleccionada
        If formMaq.ShowDialog() = DialogResult.OK Then 'Muestro el formulario y compruebo si el
usuario aceptó
            'Creo el nuevo Modelo de Clase
            sMaquinas.CrearMaquina(formMaq.Nombre, formMaq.Descripcion, formMaq.Familia,
formMaq.Listaprop)
            CargarDatosEnTreeView() 'Vuelvo a cargar el TreeView con los
datos actuales
            catalogoModificado = True 'Establezco el centinela de catálogo
modificado
        End If
    Else
        MessageBox.Show("Seleccione primero una Familia de máquinas a la que pertenecerá la nueva
Máquina")
    End If
End Sub

''' <summary>
''' Método para permitir borrar desde el TreeView con el botón Supr del teclado
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub TreeViewCatalogo_KeyPress(sender As Object, e As KeyEventArgs) Handles
TreeViewCatalogo.KeyUp
    Dim nodoSeleccionado As TreeNode = TreeViewCatalogo.SelectedNode 'Tomo el nodo seleccionado
    'Si se presiona Supr y hay algo seleccionado en el TreeView
    If e.KeyCode = Keys.Delete AndAlso nodoSeleccionado IsNot Nothing Then
        Borrar_Click(Nothing, Nothing) 'Llama al método para borrar el modelo seleccionado
    End If
End Sub

''' <summary>
''' Método que captura el evento de clicar para editar un módulo
''' Comprueba si hay un elemento seleccionado
''' Si es una familia, muestra el formulario de editar familia
''' Si es una máquina, muestra el formulario de editar máquina
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub Editar_Click(sender As Object, e As EventArgs) Handles TSMenuEditar.Click,
TSMItemEditar.Click
    catalogoModificado = True 'Establezco el centinela de catálogo
modificado
    Dim nodoSeleccionado As TreeNode = TreeViewCatalogo.SelectedNode 'Tomo el nodo seleccionado
    'Si hay un nodo familia seleccionado, lo casteo a familia

```

```

        If nodoSeleccionado IsNot Nothing AndAlso TypeOf nodoSeleccionado.Tag Is FamiliaMaquina Then
            formFam = New FormFamilia(nodoSeleccionado.Text) 'Form para editar la info de la familia
            formFam.ShowDialog() 'Muestro el Formulario
            sMaquinas.ActualizaFamilia(formFam.Nombre, formFam.Descripcion) 'Edito la familia de
máquinas
            CargarDatosEnTreeView() 'Vuelvo a cargar el TreeView con los datos actuales
        ElseIf nodoSeleccionado IsNot Nothing AndAlso TypeOf nodoSeleccionado.Tag Is Maquina Then
            'Obtengo el ID de la familia a la que pertenece la máquina a editar
            Dim IDFam As Guid = sMaquinas.GetIDFamilia(nodoSeleccionado.Parent.Text)
            'Formulario para editar la info de la Maquina
            formMaq = New FormMaquina(sMaquinas.GetFamiliaID(IDFam), nodoSeleccionado.Text)
            formMaq.ShowDialog() 'Muestro el Formulario
            'Actualizo la máquina
            sMaquinas.ActualizaMaquina(formMaq.ID, formMaq.Nombre, formMaq.Descripcion,
formMaq.Familia,
formMaq.Listaprop)
            CargarDatosEnTreeView() 'Vuelvo a cargar el TreeView con los datos actuales
            sLineas.NuevaLinea() 'Creo linea nueva por si la anterior se ve afectada por los cambios
en el catálogo
        Else
            MessageBox.Show("Seleccione una Familia o una Máquina para editar")
        End If
    End Sub

''' <summary>
''' Método que captura la pulsación sobre el botón de importar catálogo de máquinas
''' </summary>
''' <param name="sender">El botón importar catálogo</param>
''' <param name="e">El clickado sobre el botón</param>
Private Sub ImportarCatalogo_Click(sender As Object, e As EventArgs) Handles
BttImportarCatalogo.Click,
TSMItemImportarCatalogo.Click
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then 'Muestro el diálogo de abrir archivo
        CargaCatalogo(OpenFileDialog1.FileName) 'Llamo a cargar el catálogo con esa dirección
seleccionada
        catalogoModificado = True
        CargarDatosEnTreeView() 'Vuelvo a cargar el TreeView con los datos actuales
        sLineas.NuevaLinea() 'Creo linea nueva por si la anterior se ve afectada por los cambios
en el catálogo
    End If
End Sub

''' <summary>
''' Método que captura la pulsación sobre el botón de exportar catálogo de máquinas
''' </summary>
''' <param name="sender">El botón exportar catálogo</param>
''' <param name="e">El clickado sobre el botón</param>
Private Sub ExportarCatalogo_Click(sender As Object, e As EventArgs) Handles
BttExportarCatalogo.Click,
TSMItemExportarCatalogo.Click
    If SaveFileDialog1.ShowDialog() = DialogResult.OK Then 'Muestro el diálogo de elegir archivo
donde exportar
        'Llamo a guardar el catálogo con esa dirección seleccionada
        sMaquinas.GuardarEn(SaveFileDialog1.FileName)
    End If
End Sub

''' <summary>
''' Método que captula la pulsación sobre el botón de guardar catálogo
''' Guarda el estado de del catálogo actual en el archivo local a la aplicación
''' </summary>
''' <param name="sender">El botón guardar catálogo</param>
''' <param name="e">El clickado sobre el botón</param>
Private Sub GuardarCatalogo_Click(sender As Object, e As EventArgs) Handles BttGuardar.Click,
TSMItemGuardar.Click
    Dim respuesta As DialogResult = MessageBox.Show("¿Guardar los cambios en el catálogo actual?",
"Guardar", "catálogo", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning)
    Select Case respuesta
        Case DialogResult.Yes
            sMaquinas.GuardarEn(catalogoLocal) 'Guardo el catálogo en el fichero local
            RichTextBoxEstado.Clear() 'Actualizo la información en el richtextbox de
estado
            RichTextBoxEstado.AppendText("No hay cambios pendientes en el catálogo de máquinas
local")
            catalogoModificado = False 'Cambio el centinela del catálogo
        Case Else
            'No hago nada
    End Select

```

```

End Sub

''' <summary>
''' Método que captura la pulsación sobre el botón de olvidar catálogo
''' Elimina todos los modelos del catálogo actual tras confirmación del usuario
''' Pero no guarda los cambios en el fichero local
''' </summary>
''' <param name="sender">El botón olvidar catálogo</param>
''' <param name="e">El clickado sobre el botón</param>
Private Sub NuevoCatalogo_Click(sender As Object, e As EventArgs) Handles BttNuevoCatalogo.Click,
    TSMItemNuevoCatalogo.Click
    If MessageBox.Show("¿Confirmar borrado de todas las Familias y Máquinas del catálogo actual?",
        "Borrar catálogo", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) = DialogResult.Yes
Then
    sMaquinas.BorraCatalogo() 'Llama al método de servicioModelos que vacía el catálogo
    CargarDatosEnTreeView() 'Actualizo el treeview
    sLineas.NuevaLinea() 'Creo línea nueva por si la anterior se ve afectada por los cambios
en el catálogo
End If
End Sub

'-----
'| Métodos del Modo Dimensionar |
'-----

''' <summary>
''' Método que captura la pulsación sobre el botón para editar los datos del informe
''' </summary>
''' <param name="sender">El botón Datos</param>
''' <param name="e">El clickado sobre el botón</param>
Private Sub DatosLinea_Click(sender As Object, e As EventArgs) Handles BttDatosLinea.Click,
    TSMItemDatosLinea.Click
    formDatosLinea = New FormDatosLinea(sLineas.Linea.Nombre,
        sLineas.Linea.Cliente,
        sLineas.Linea.Proyecto,
        sLineas.Linea.Version,
        sLineas.Linea.Descripcion) 'Formulario para tomar
los datos
If formDatosLinea.ShowDialog() = DialogResult.OK Then 'Compruebo si el usuario aceptó el
nuevo form
    sLineas.Linea.Nombre = formDatosLinea.Nombre
    sLineas.Linea.Cliente = formDatosLinea.Cliente
    sLineas.Linea.Proyecto = formDatosLinea.Proyecto
    sLineas.Linea.Version = formDatosLinea.Version
    sLineas.Linea.Descripcion = formDatosLinea.Descripcion 'Paso los datos al informe
    informe.CreaInforme() 'Regenero Informe
    'Actualizo contenido del informe en el richtextbox de este form
    RichTextBoxInforme.Rtf = informe.Contenido.Rtf
End If
End Sub

''' <summary>
''' Método que captura la pulsación sobre el botón para seleccionar las Familias de máquinas,
''' que conformarán la línea, a tener en cuenta en el dimensionamiento
''' </summary>
''' <param name="sender">El botón Selección de Familias de Máquinas</param>
''' <param name="e">El clickado sobre el botón</param>
Private Sub SeleccionFamilias_Click(sender As Object, e As EventArgs) Handles
BttSeleccionFamilias.Click,
TSMItemSeleccionFamilias.Click
    formSelFam = New FormSelecFam() 'Formulario para seleccionar las Familias
    formSelFam.ListaPrevía = sLineas.Linea.SeleccionFam 'Le envío la selección actual
    If formSelFam.ShowDialog() = DialogResult.OK Then 'Compruebo si el usuario aceptó el nuevo
form
        sLineas.Linea.Eschema = formSelFam.Eschema
        sLineas.Linea.SeleccionFam = formSelFam.SeleccionModClase
        dimensionar.Dimensiona()
        informe.CreaInforme() 'Regenero Informe
        'Actualizo contenido del informe en el richtextbox de este form
        RichTextBoxInforme.Rtf = informe.Contenido.Rtf
    End If
End Sub

''' <summary>
''' Método que muestra el formulario para seleccionar las condiciones de dimensionamiento y
''' realizar el dimensionamiento
''' </summary>
''' <param name="sender">Botón Condiciones</param>
''' <param name="e">El clickado sobre el botón</param>

```

```

Private Sub Condiciones_Click(sender As Object, e As EventArgs) Handles BttCondiciones.Click,
    TSMItemCondiciones.Click
    'Formulario para seleccionar las condiciones para el dimensionamiento
    'Le paso los datos para que mantenga consistencia con la ultima seleccion hecha
    formCond = New FormCondiciones(sLineas.Linea.Condiciones.Paralelismo,
        sLineas.Linea.Condiciones.ProduccionMin,
        sLineas.Linea.Condiciones.MinimizarPrimaria,
        sLineas.Linea.Condiciones.MinimizarSecundaria)
    If formCond.ShowDialog() = DialogResult.OK Then 'Compruebo si el usuario aceptó el nuevo form
        'Y en tal caso, asigno los valores a las condiciones de la línea
        sLineas.Linea.Condiciones.Paralelismo = formCond.paralelismo
        sLineas.Linea.Condiciones.ProduccionMin = formCond.produccionMin
        sLineas.Linea.Condiciones.MinimizarPrimaria = formCond.minimizarPrimaria
        sLineas.Linea.Condiciones.MinimizarSecundaria = formCond.minimizarSecundaria
        'Llamo al método para dimensionar la línea con esas condiciones y paso el resultado al
informe
        dimensionar.Dimensiona()
        informe.CreaInforme() 'Regenero Informe
        'Actualizo el contenido del informe en el richtextbox de este form
        RichTextBoxInforme.Rtf = informe.Contenido.Rtf

    End If
End Sub

''' <summary>
''' Método que captura la pulsación sobre el botón para imprimir el informe. Abre el diálogo para
''' configurar la impresora y llama al método imprime de la clase que encapsula el tratamiento de
''' impresión, pasándole el informe y la configuración de impresora
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub Imprimir_Click(sender As Object, e As EventArgs) Handles BttImprimir.Click,
TSMItemImprimir.Click
    'Llamo a la ventana de configuración de impresión, y si es aceptada:
    If PrintDialog1.ShowDialog() = DialogResult.OK Then
        imprimir = New Imprimir(RichTextBoxInforme)
        'Tomo la config.de impresión y la envío a mi gestor de impresion
        imprimir.Imprime(PrintDialog1.PrinterSettings)
    End If
End Sub

''' <summary>
''' Método que captura el evento de pulsar sobre el botón de Mostrar y ocultar el valor de coste.
''' Cambia booleano que define si se mostrará o no y el estado de los botones
''' </summary>
''' <param name="sender">El botón del menú Mostrar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub MostrarValorCoste_Click(sender As Object, e As EventArgs) Handles TSMItemMostrar.Click,
    BttVistaValorCoste.Click, TSMItemOcultar.Click,
    BttVistaValorCoste2.Click
    If valorCosteVisible Then
        valorCosteVisible = False
        TSMItemMostrar.Enabled = True
        TSMItemOcultar.Enabled = False
        BttVistaValorCoste.Image = My.Resources.MostrarCoste
    Else
        valorCosteVisible = True
        TSMItemMostrar.Enabled = False
        TSMItemOcultar.Enabled = True
        BttVistaValorCoste.Image = My.Resources.OcultarCoste
    End If
    CargarDatosEnTreeView() 'Actualizo, para, en caso de que se esté mostrando o no un valor coste,
ya no lo haga
    informe.valorCosteVisible = valorCosteVisible 'Actualizo el informe también
    informe.CreaInforme()
    'Actualizo el contenido del informe en el richtextbox de este form
    RichTextBoxInforme.Rtf = informe.Contenido.Rtf
End Sub

''' <summary>
''' Método que captura el evento para exportar la línea actual a un fichero.
''' Abre el diálogo para seleccionar el fichero destino y con esa información llama al método
correspondiente
''' del servicioLineas.
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>

```

```

Private Sub ExportarLinea_Click(sender As Object, e As EventArgs) Handles
TSMItemExportarLinea.Click,
    BttExportarLinea.Click
    If SaveFileDialog1.ShowDialog() = DialogResult.OK Then 'Muestro el diálogo de elegir archivo
    donde exportar
        sLineas.GuardarEn(SaveFileDialog1.FileName) 'Llamo a guardar la línea con esa dirección
    seleccionada
    End If
End Sub

''' <summary>
''' Método que captura el evento para importar una línea guardada en un fichero.
''' Abre el diálogo para seleccionar el fichero a importar. Llama al servicioLineas
''' para que compruebe si existen en catálogo actual las máquinas necesarias para la línea a
importar.
''' Si existen, le indica al servicioLinea que acepte la línea importada, si no, consulta al
usuario
''' si quiere añadir las máquinas de la línea importada al catálogo. Si no acepta, no se carga la
línea.
''' Si se ha cargado la línea, actualiza el informe que se muestra de la nueva línea y abre el
formulario que
''' muestra el esquema de la nueva línea para que el usuario la revise y acepte o modifique.
''' </summary>
''' <param name="sender">El botón de importar o el elemento correspondiente del menú
principal</param>
''' <param name="e">El clicado sobre el botón o elemento de menú</param>
Private Sub ImportarLinea_Click(sender As Object, e As EventArgs) Handles
TSMItemImportarLinea.Click,
    BttImportarLinea.Click
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then 'Muestro el diálogo de abrir archivo
    sLineas.CargarDesde(OpenFileDialog1.FileName) 'Llamo a cargar la línea con esa dirección
    seleccionada
    If sLineas.CompruebaLinea() Then
        sLineas.AceptaLinea()
    Else
        If MessageBox.Show("Una o más Máquinas utilizadas en la Línea a importar no se
    encuentran en el
        catálogo de Máquinas actual. ¿Añadir estas máquinas al catálogo?", "Comprobación
    Máquinas",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) =
    DialogResult.OK Then
        sLineas.AceptaLinea(True)
        catalogoModificado = True 'Marco que se ha modificado el catálogo
        RichTextBoxEstado.Clear() 'Actualizo la información en el richtextbox
    de estado
        RichTextBoxEstado.AppendText("Se han añadido nuevas Familias de Máquinas al
    catálogo")
    Else
        Return
    End If
    End If
    informe.CreaInforme() 'Actualizo el contenido del informe para la nueva linea
    RichTextBoxInforme.Rtf = informe.Contenido.Rtf 'Muestro el contenido del informe en el
richtextbox
    'Llamo al método para mostrar el formulario de selección de Familias, y que así el usuario
    revise
    'el esquema de la linea y lo acepte, además así se regenera el dibujo a incluir en el
    informe de la linea
    SeleccionFamilias_Click(Nothing, Nothing)
End If
End Sub

''' <summary>
''' Método que captura la pulsación sobre los botones para crear una nueva linea.
''' Avisa de que se perderán los datos de la linea anterior y, en caso de aceptar,
''' crea una nueva linea por defecto y actualiza el informe y lo muestra en el
''' richtextbox del formulario
''' </summary>
''' <param name="sender">Los botones nueva linea</param>
''' <param name="e">El click sobre el botón</param>
Private Sub NuevaLinea_Click(sender As Object, e As EventArgs) Handles BttNuevaLinea.Click,
TSMItemNuevaLinea.Click
    If MessageBox.Show("Se perderán los datos de la linea actual. ¿Continuar?",
    "Nueva Linea a dimensionar", MessageBoxButtons.YesNo,
    MessageBoxIcon.Warning) = DialogResult.Yes Then
        sLineas.NuevaLinea() 'Creo una nueva linea, con los valores por defecto
        informe.CreaInforme() 'Actualizo el contenido del informe para la nueva linea

```

```

        RichTextBoxInforme.Rtf = informe.Contenido.Rtf 'Muestro el contenido del informe en el
richtextbox
    End If
End Sub

''' <summary>
''' Método que captura la pulsación sobre el botón salir del menú archivo
''' </summary>
''' <param name="sender">ToolStripMenuItem Salir</param>
''' <param name="e">Click sobre el elemento</param>
Private Sub Salir_Click(sender As Object, e As EventArgs) Handles TSMItemSalir.Click
    Me.Close()
End Sub

''' <summary>
''' Método que captura el evento de pulsar sobre el botón de Acerca de... y muestra la ventana de
información
''' </summary>
''' <param name="sender">El botón del menú de Acerca de...</param>
''' <param name="e">El click sobre el botón</param>
Private Sub AcercaDe_Click(sender As Object, e As EventArgs) Handles TSMItemAcercaDe.Click
    formAcercaDe = New FormAcercaDe()
    formAcercaDe.Show()
End Sub
End Class

```

B.2 Clase ServicioMaquinas

```

Imports System.IO
Imports System.Xml.Serialization

''' <summary>
''' Esta clase es el gestor de las familias de máquinas y las propias máquinas
''' Alojara el catálogo de Máquinas (listado de familias de máquinas) y todos los
''' métodos para crearlos, editarlos, añadirlos o eliminarlos del catálogo y demás
''' interacciones con ellos.
''' Se tratará como Singleton para que sea siempre la misma instancia a la que se llame.
''' </summary>
''' <remarks>
''' Sólo permito una instancia (Singleton), obligando así a mantener un
''' estado consistente de la gestión de la biblioteca
''' </remarks>
Public Class ServicioMaquinas
    ''' <summary>Almacena su propia y única instancia </summary>
    Private Shared instanciaSMaquinas As ServicioMaquinas
    '''<summary>Almacena las familias de máquinas del catálogo</summary>
    Private catalogo As List(Of FamiliaMaquina)
    '''<summary>Tipo enumerado para que se pueda informar de qué tipo se trata,
    '''antes de crearlo incluso</summary>
    Enum TipoElemento
        TipoFamilia
        TipoMaquina
    End Enum

    ''' <summary>
    ''' Constructor privado. Sólo se genera la instancia desde la propia clase
    ''' Inicializa el catálogo
    ''' </summary>
    Private Sub New()
        catalogo = New List(Of FamiliaMaquina) 'Inicializo la lista familias de máquinas
    End Sub

    ''' <summary>
    ''' Método (propiedad) para acceder a la clase
    ''' </summary>
    ''' <returns>la única instancia de la clase</returns>
    Public Shared ReadOnly Property Instance() As ServicioMaquinas
        Get
            If instanciaSMaquinas Is Nothing Then 'Si aún no se ha instanciado, creo una nueva
                instanciaSMaquinas = New ServicioMaquinas()
            End If
            Return instanciaSMaquinas
        End Get
    End Property

```

```

''' <summary>
''' Método para crear Nuevas Familias de máquinas
''' </summary>
''' <param name="nombre">El nombre de la familia</param>
''' <param name="descripcion">La descripción de la familia</param>
Public Sub CreaFamilia(nombre As String, descripcion As String)
    Try
        Dim fam As New FamiliaMaquina(nombre, descripcion) 'Creo una familia nueva
        catalogo.Add(fam) 'Lo añado a la lista
    Catch ex As Exception
        MessageBox.Show("Error al añadir la familia de máquinas: " & ex.Message, "Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error)
    End Try
End Sub

''' <summary>
''' Método para crear nuevas máquinas y añadirlas a la familia correspondiente
''' </summary>
''' <param name="nombre">El nombre de la máquina</param>
''' <param name="descripcion">La descripción de la máquina</param>
''' <param name="familia">La clase a la que añadir la nueva máquina</param>
''' <param name="listaPropiedades">Los valores de los atributos de la máquina</param>
Public Sub CrearMaquina(nombre As String, descripcion As String, familia As FamiliaMaquina,
        listaPropiedades As List(Of Single))
    Try
        Dim maq As New Maquina(nombre, descripcion) 'Creo nueva instancia de Maquina
        maq.AsignaPropiedades(listaPropiedades) 'Le asigno los valores de sus atributos de propiedades
        familia.Maquinas.Add(maq) 'Añado la máquina a la lista de máquinas de la familia correspondiente
    Catch ex As Exception
        MessageBox.Show("Error al añadir la máquina: " & ex.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
    End Try
End Sub

''' <summary>
''' Guarda el catálogo actual.
''' Guarda la lista actual de familias de máquinas (con sus máquinas) en un archivo.
''' </summary>
''' <param name="archivo">El archivo destino</param>
Public Function GuardarEn(archivo As String)
    Try
        Dim serializer As New XmlSerializer(GetType(List(Of FamiliaMaquina))) 'Nueva inst. del
XmlSerializer
        'para el tipo de datos
        Using writer As New StreamWriter(archivo) 'Abre el archivo para escritura
            serializer.Serialize(writer, catalogo) 'Serializa y escribe los modelos en el archivo XML
        End Using
        Return "Se ha guardado el catálogo de máquinas."
    Catch ex As Exception
        Return "Error al guardar el catálogo de máquinas: " & ex.Message 'Error al guardar
    End Try
End Function

''' <summary>
''' Lee el archivo de guardado de Familias de máquinas (catálogo) y los carga en la aplicación
''' Devolviendo texto sobre el estado en que queda el catálogo
''' </summary>
''' <param name="archivo">El archivo origen</param>
''' <returns>Texto a mostrar en estado catálogo</returns>
Public Function CargaDesde(archivo As String) As String
    Try
        If File.Exists(archivo) Then 'Verifico que existe el archivo antes de cargar
            Dim serializer As New XmlSerializer(GetType(List(Of FamiliaMaquina))) 'Nueva inst. del
            'XmlSerializer para el tipo de datos
            Using reader As New StreamReader(archivo) 'Abro el archivo para lectura
                'Deserializo y leo los modelos desde el XML
                Me.catalogo = CType(serializer.Deserialize(reader), List(Of FamiliaMaquina))
                'Devuelvo, el mensaje de confirmación
                Return "Catálogo de máquinas cargado desde el archivo: " & vbCrLf & vbTab & archivo
            End Using
        Else
            'Error si no hay archivo de biblioteca
            Return "No se ha encontrado el archivo del catálogo." & vbCrLf &
            "No se han cargado máquinas."
        End If
    Catch ex As Exception
        Return "Error al cargar el catálogo de máquinas: " & ex.Message 'Error al cargar
    End Try
End Function

```

```

''' <summary>
''' Función que devuelve la lista Familias de máquinas
''' </summary>
''' <returns>Lista de Familias de máquinas</returns>
Public Function GetFamilias() As List(Of FamiliaMaquina)
    Return catalogo 'Devuelvo la lista de familias
End Function

''' <summary>
''' Elimina la familia de máquinas de la lista (catálogo) que coincida con el nombre
''' </summary>
''' <param name="nombreFamilia">El nombre de la familia a eliminar</param>
''' <returns></returns>
Public Function DelFamilia(nombreFamilia As String) As Boolean
    ' Nota: Gestionaré todo para que no se pueda crear una familia con el mismo nombre que otra,
    ' pero de esta forma me aseguro de mantenerlo así.
    ' Uso RemoveAll para eliminar todas las familias que coincidan con el nombre.
    Dim cantidadEliminados As Integer = catalogo.RemoveAll(Function(fam) fam.Nombre = nombreFamilia)
    If cantidadEliminados > 0 Then ' Si se eliminan una o más Familias, entonces OK.
        Return True ' Éxito
    Else
        Return False ' No se encontró ninguna familia para eliminar.
    End If
End Function

''' <summary>
''' Comprueba si alguna familia o máquina ya existe con un nombre dado
''' </summary>
''' <param name="nombre">Nombre a comprobar</param>
''' <param name="familia">La familia a la que pertenece si es una máquina a comprobar</param>
''' <returns>True si ya existe, False si no se ha encontrado coincidencia</returns>
Public Function CompruebaNombre(nombre As String, Optional familia As FamiliaMaquina = Nothing)
    If familia Is Nothing Then
        For Each fam As FamiliaMaquina In catalogo
            If fam.Nombre = nombre Then
                Return True 'Alguna familia ya tiene ese nombre
            End If
        Next
    Else
        For Each maq As Maquina In familia.Maquinas
            If maq.Nombre = nombre Then
                Return True 'Alguna máquina de esta familia ya tiene ese nombre
            End If
        Next
    End If
    Return False 'El nombre es válido
End Function

''' <summary>
''' Elimina la máquina de la lista en la familia a la que pertenece
''' </summary>
''' <param name="nombre">Nombre de la máquina a eliminar</param>
''' <param name="familia">Familia de Máquinas a la que pertenece</param>
''' <returns></returns>
Public Function DelMaquina(nombre As String, familia As FamiliaMaquina) As Boolean
    If CompruebaNombre(nombre, familia) Then 'Si existe un Modelo de Clase con ese nombre en la lista
        For Each fam As FamiliaMaquina In catalogo
            If fam.Nombre = familia.Nombre Then 'Tomo la familia en particular
                ' Uso RemoveAll para eliminar todas las máquinas que coincidan con el nombre.
                Dim cantidadEliminados As Integer =
                    fam.Maquinas.RemoveAll(Function(maq) maq.Nombre = nombre)
                If cantidadEliminados > 0 Then ' Si se eliminan una o más Máquinas, entonces OK.
                    Return True ' Éxito
                End If
            End If
        Next
    End If
    Return False ' No se encontró ninguna máquina para eliminar.
End Function

''' <summary>
''' Devuelve un texto ya formateado con las propiedades de la familia indicada
''' </summary>
''' <param name="nombre">Nombre de la familia de la que se quieren las propiedades</param>
''' <returns>Texto formateado, con las propiedades de la familia indicada</returns>
Public Function GetDatosFamilia(nombre As String) As String
    Dim textoFam As String 'Declaro el texto a devolver
    textoFam = "ERROR: " & vbCrLf & "No se han encontrado datos" 'Inicialmente, devuelvo mensaje de error

```

```

'Compruebo que existe una familia con ese nombre en la lista
If CompruebaNombre(nombre) Then
    For Each fam As FamiliaMaquina In catalogo
        If fam.Nombre = nombre Then
            textoFam = "NOMBRE: " & fam.Nombre & vbCrLf & "DESCRIPCIÓN: " & fam.Descripcion
        End If
    Next
End If
Return textoFam
End Function

''' <summary>
''' Devuelve un texto ya formateado con las propiedades de la máquina indicada,
''' perteneciente a la familia también indicada. Muestra o no el valor de coste
''' según se indique
''' </summary>
''' <param name="nombre">Nombre de la máquina de la que se quieren las propiedades</param>
''' <param name="nombreFam">Nombre de la familia a la que pertenece la máquina</param>
''' <param name="valorCosteVista">Booleano que indica si mostrar o no el valor de coste</param>
''' <returns>Texto formateado, con las propiedades de la máquina indicada</returns>
Public Function GetDatosMaquina(nombre As String, nombreFam As String, valorCosteVista As Boolean) As
String
    Dim textoMaq As String 'Declaro el texto a devolver
    textoMaq = "ERROR: " & vbCrLf & "No se han encontrado datos"
    'If CompruebaNombre(nombre, familia) Then 'Ya lo comprueba el formulario antes de llamar aquí
    For Each fam As FamiliaMaquina In catalogo
        If fam.Nombre = nombreFam Then
            For Each maq As Maquina In fam.Maquinas
                If maq.Nombre = nombre Then
                    If valorCosteVista Then
                        textoMaq = maq.ToString() 'Tomo directamente el texto desde la máquina
ToStringNoCoste()
                    Else
                        textoMaq = maq.ToStringNoCoste() 'Tomo el texto desde la máquina-Sin valor de
coste!
                    End If
                End If
            Next
        End If
    Next
    Return textoMaq
End Function

''' <summary>
''' Devuelve el atributo descripción de la familia o máquina correspondiente al nombre pasado como
argumento
''' Si se incluye el argumento nombreMaq, se refiere al atributo de una máquina
''' </summary>
''' <param name="nombre">Nombre del cual se requiere el atributo</param>
''' <param name="nombreMaq">En caso de incluir éste parámetro, éste será la máquina del que se requiere
el atributo y "nombre" será la familia al que pertenece</param>
''' <returns></returns>
Public Function GetDescripcion(nombre As String, Optional nombreMaq As String = Nothing) As String
    Dim texto As String 'Declaro el texto a devolver
    texto = "ERROR: " & vbCrLf & "No se han encontrado datos"
    For Each fam As FamiliaMaquina In catalogo
        If fam.Nombre = nombre Then
            If nombreMaq Is Nothing Then 'Si se busca el atributo de una Familia
                Return fam.Descripcion 'Devuelvo el atributo Descripción
            Else 'Si se busca el atributo de una Máquina
                For Each maq As Maquina In fam.Maquinas
                    If maq.Nombre = nombreMaq Then
                        texto = maq.Descripcion 'Devuelvo el atributo Descripción
                    End If
                Next
            End If
        End If
    Next
    Return texto
End Function

''' <summary>
''' Método para actualizar los atributos de una familia
''' </summary>
''' <param name="nombre">La familia a actualizar</param>
''' <param name="descripcion">El Atributo Descripción a actualizar</param>
Public Sub ActualizaFamilia(nombre As String, descripcion As String)
    For Each fam As FamiliaMaquina In catalogo
        If fam.Nombre = nombre Then 'Busco la familia a modificar

```

```

        fam.Descripcion = descripcion 'Le piso el atributo con el nuevo
    End If
Next
End Sub

''' <summary>
''' Función que devuelve el nombre correspondiente de una familia, dado su ID
''' </summary>
''' <param name="identificador">El ID del que se requiere su nombre</param>
''' <returns>El nombre de la familia</returns>
Public Function GetNombrePorID(identificador As Guid) As String
    For Each fam As FamiliaMaquina In catalogo 'Busco el identificador en la lista de familias
        If fam.ID = identificador Then
            Return fam.Nombre
        End If
    Next
    Return String.Empty 'Si no se encuentra, devuelvo un string vacío
End Function

''' <summary>
''' Método para borrar el listado de familias
''' Borra el catálogo
''' </summary>
''' <remarks>Atención - Se perderá todo!</remarks>
Public Sub BorraCatalogo()
    catalogo.Clear()
End Sub

''' <summary>
''' Función que devuelve una familia, dado su ID
''' </summary>
''' <param name="identificador">El ID que identifica a la familia</param>
''' <returns>la familia</returns>
Public Function GetFamiliaID(identificador As Guid) As FamiliaMaquina
    For Each fam As FamiliaMaquina In catalogo 'Busco el identificador en la lista de familias
        If fam.ID = identificador Then
            Return fam
        End If
    Next
    Return Nothing 'Si no se encuentra, devuelvo Nothing
End Function

''' <summary>
''' Función que buscará la máquina dada perteneciente a la familia dada
''' y devolverá sus atributos de descripción, operadores, pot.Elect., Valor Coste y Valor Venta
''' </summary>
''' <param name="nombreFamilia">El nombre de la familia a la que pertenece la máquina que se busca</param>
''' <param name="nombreMaquina">El nombre de la familia dela que se buscan sus atributos</param>
''' <returns>Lista de Strings conteniendo los atributos descripción, operadores, pot.Elect., Valor Coste
''' y Valor Venta de la máquina</returns>
Public Function GetAtributosMaquina(nombreFamilia As String, nombreMaquina As String) As List(Of String)
    Dim listaAtributos As New List(Of String) From {"", "", "", "", "", ""}'Declaro la lista a devolver
    For Each fam As FamiliaMaquina In catalogo 'Compruebo que existe una Familia con ese nombre
        If fam.Nombre = nombreFamilia Then
            For Each maq As Maquina In fam.Maquinas
                If maq.Nombre = nombreMaquina Then
                    listaAtributos(0) = maq.Descripcion
                    listaAtributos(1) = maq.Operarios
                    listaAtributos(2) = maq.PotenciaElectrica
                    listaAtributos(3) = maq.Produccion
                    listaAtributos(4) = maq.ValorCoste
                    listaAtributos(5) = maq.ValorVenta
                End If
            Next
        End If
    Next
    Return listaAtributos
End Function

''' <summary>
''' Método que actualiza los atributos de una Maquina
''' </summary>
''' <param name="ID">El Identificador de la Maquina que se va a editar</param>
''' <param name="nombre">El nuevo nombre de la Maquina</param>
''' <param name="descripcion">La nueva descripción de la Maquina</param>
''' <param name="familia">La familia a la que pertenece la Maquina</param>
''' <param name="listaPropiedades">Los nuevos valores de los atributos de la Maquina</param>
Public Sub ActualizaMaquina(ID As Guid, nombre As String, descripcion As String, familia As
FamiliaMaquina,
                                listaPropiedades As List(Of Single))

```

```

    For Each fam As FamiliaMaquina In catalogo 'Busco la familia a la que pertenece
    If fam.ID = familia.ID Then
        For Each maq As Maquina In fam.Maquinas 'Busco la máquina con igual ID
        If maq.ID = ID Then 'Si coincide el ID, entonces
            maq.Nombre = nombre 'Actualizo todos los atributos
            maq.Descripcion = descripcion
            maq.AsignaPropiedades(listaPropiedades)
        End If
        Next
    End If
Next
End Sub

''' <summary>
''' Función que devuelve el ID de una máquina en particular
''' </summary>
''' <param name="nombreFamilia">Nombre de la familia a la que pertenece la máquina</param>
''' <param name="nombreMaquina">Nombre de la máquina</param>
''' <returns>Devuelve el ID de la máquina requerida</returns>
Public Function GetIDMaquina(nombreFamilia As String, nombreMaquina As String) As Guid
    For Each fam As FamiliaMaquina In catalogo 'Busco la familia a la que pertenece
        If fam.Nombre = nombreFamilia Then
            For Each maq As Maquina In fam.Maquinas 'Busco la maquina con igual nombre
                If maq.Nombre = nombreMaquina Then
                    Return maq.ID 'Devuelvo el ID
                End If
            Next
        End If
    Next
    Return Nothing 'Si no se encuentra, devuelvo nada
End Function

''' <summary>
''' Función que devuelve el ID de una familia en particular
''' </summary>
''' <param name="nombreFamilia">El nombre de la familia de la que se requiere el ID</param>
''' <returns>Devuelve el ID de la familia requerida</returns>
Public Function GetIDFamilia(nombreFamilia As String) As Guid
    For Each fam As FamiliaMaquina In catalogo 'Busco la familia
        If fam.Nombre = nombreFamilia Then
            Return fam.ID 'Devuelvo el ID
        End If
    Next
    Return Nothing 'Si no se encuentra, devuelvo nada
End Function

''' <summary>
''' Método que comprueba si un ID dado existe como Familia en su listado
''' </summary>
''' <param name="identificador">El Identificador de una Familia</param>
''' <returns>True si existe, False si no</returns>
Friend Function ExisteFamiliaPorID(identificador As Guid) As Boolean
    For Each fam In catalogo 'Busco la familia
        If fam.ID = identificador Then
            Return True
        End If
    Next
    Return False
End Function

''' <summary>
''' Método que añade la familia pasada a la lista de familias (catalogo)
''' </summary>
''' <param name="familia">Familia a añadir a la lista de Familias de máquinas</param>
Friend Sub AddFamilia(familia As FamiliaMaquina)
    catalogo.Add(familia)
End Sub
End Class

```

B.3 Clase ServicioLineas

```

Imports System.IO
Imports System.Xml.Serialization

''' <summary>

```

```

''' Esta clase es el gestor de las líneas a dimensionar. Alojara la línea sobre la que se está trabajando
''' en la aplicación y todos los métodos para crearla, editarla, exportarla o importar una existente en fichero.
''' Para la importación, gestionará la carga o no de las familias de máquinas incluidas en la línea si no se
''' encuentran en el catálogo actual. Se tratará como Singleton para que sea siempre la misma instancia a la
''' que se llame
''' </summary>

Public Class ServicioLineas
    ''' <summary>Almacena su propia y única instancia </summary>
    Private Shared instanciaServicioLineas As ServicioLineas
    ''' <summary> Almacena la Línea que se va a dimensionar</summary>
    Private _línea As Línea
    ''' <summary> Almacena temporalmente la línea que se quiere importar pero aún no comprobada</summary>
    Private _líneaImportada As Línea
    '''<summary> Gestor de máquinas (catálogo)</summary>
    Private sMáquinas As ServicioMáquinas

    ''' <summary>
    ''' Constructor privado. Sólo se genera la instancia desde la propia clase
    ''' </summary>
    Private Sub New()
        NuevaLínea()
        sMáquinas = ServicioMáquinas.Instance 'La instancia (única) del servicioMáquinas
    End Sub

    ''' <summary>
    ''' Método que crea una nueva línea. Es llamado por el constructor de esta clase y por el método para generar
    ''' una nueva línea a dimensionar
    ''' Crea una nueva Línea con todos los valores por defecto
    ''' </summary>
    Public Sub NuevaLínea()
        Línea = New Línea()
    End Sub

    ''' <summary>
    ''' Guarda la línea en un archivo
    ''' </summary>
    ''' <param name="archivo">El archivo destino</param>
    Public Sub GuardarEn(archivo As String)
        Try
            'Guardo en la línea las familias de máquinas utilizados por esa línea para poder recuperarlos cuando
            'se importe de nuevo si en ese momento no existen en el catálogo
            For Each identificador In Línea.SeleccionFam 'Para cada ident de familia en la lista de seleccionadas
                Línea.FamiliasLínea.Add(sMáquinas.GetFamiliaID(identificador)) 'Tomo la familia y la guardo en
la línea
            Next
            Dim serializer As New XmlSerializer(GetType(Línea)) 'Nueva inst. del XmlSerializer para el tipo de
datos
            Using writer As New StreamWriter(archivo)
                serializer.Serialize(writer, Línea) 'Abre el archivo para escritura
                'Serializa y escribe los modelos en el archivo
XML
            End Using
            MessageBox.Show("Se ha guardado la línea.", "Guardado correcto", MessageBoxButtons.OK,
                MessageBoxIcon.Information) 'Mensaje confirmación
        Catch ex As Exception 'Error al guardar
            MessageBox.Show("Error al guardar la línea: " & ex.Message, "Error al guardar", MessageBoxButtons.OK,
                MessageBoxIcon.Error)
        End Try
    End Sub

    ''' <summary>
    ''' Lee la línea desde un archivo y lo carga en la aplicación
    ''' </summary>
    ''' <param name="archivo">El archivo origen</param>
    Public Sub CargarDesde(archivo As String)
        Try
            If File.Exists(archivo) Then
                'Verifico que existe el archivo antes de cargar
                Dim serializer As New XmlSerializer(GetType(Línea)) 'Nueva inst. del XmlSerializer para el
                'tipo de datos
                Using reader As New StreamReader(archivo)
                    'Abro el archivo para lectura
                    _líneaImportada = New Línea 'Inicializo la línea donde se guardará la que se importa
                    _líneaImportada = CType(serializer.Deserialize(reader), Línea) 'Deserializo y leo desde el
XML
                End Using
            Else 'Error si no hay archivo de biblioteca
                MessageBox.Show("No se ha encontrado el archivo de la línea.", "Error al cargar",
                    MessageBoxButtons.OK, MessageBoxIcon.Error)
            End If
        Catch ex As Exception 'Error al cargar
            MessageBox.Show("Error al cargar la línea: " & ex.Message, "Error al cargar", MessageBoxButtons.OK,

```

```

        MessageBoxIcon.Error)
    End Try
End Sub

''' <summary>
''' Método que comprueba si las máquinas de la línea a importar existen en el catálogo actual
''' </summary>
''' <returns>True si todos existen y False si alguno no</returns>
Friend Function CompruebaLinea() As Boolean
    If _lineaImportada.SeleccionFam IsNot Nothing Then
        For Each famID In _lineaImportada.SeleccionFam
            If Not sMaquinas.ExisteFamiliaPorID(famID) Then
                Return False
            End If
        Next
    End If
    Return True
End Function

''' <summary>
''' Método que acepta la línea recién importada, y la convierte en la Línea a trabajar
''' Si así se le indica, se copiarán sus familias de máquinas en el catálogo actual
''' </summary>
''' <param name="cargarFamilias">Boolean que indica si se desea cargar las familias de máquinas
''' de la línea en el catálogo actual</param>
''' <returns></returns>
Public Function AceptaLinea(Optional cargarFamilias As Boolean = False)
    If cargarFamilias Then
        For Each fam In _lineaImportada.FamiliasLinea
            If Not sMaquinas.ExisteFamiliaPorID(fam.ID) Then
                sMaquinas.AddFamilia(fam)
            End If
        Next
    End If
    _linea = _lineaImportada
End Function

''' <summary>
''' Método (propiedad) para acceder a la clase
''' </summary>
''' <returns>la única instancia de la clase</returns>
Public Shared ReadOnly Property Instance() As ServicioLineas
    Get
        If instanciaServicioLineas Is Nothing Then 'Si aún no se ha instanciado, creo una nueva
            instanciaServicioLineas = New ServicioLineas()
        End If
        Return instanciaServicioLineas
    End Get
End Property

''' <summary>La línea actual</summary>
Public Property Linea As Linea
    Get
        Return _linea
    End Get
    Set
        _linea = Value
    End Set
End Property
End Class

```

B.4 Clase Dimensionar

```

''' <summary>
''' Esta clase encapsula todo lo relacionado con los cálculos de selección según las condiciones del usuario
''' Se tratará como Singleton para que sea siempre la misma instancia a la que se llame.
''' Tomará cada una de las Familias de máquinas, seleccionadas por el usuario para la línea,
''' y seleccionará la máquina, de cada Familia, que mejor se adapte a las condiciones indicadas por el usuario.
''' El resultado será una lista de Máquinas que conformarán la línea dimensionada.
''' </summary>
''' <remarks>
''' Sólo permito una instancia (Singleton), ya que sólo se necesita un objeto calculador
''' </remarks>
Public Class Dimensionar

    ''' <summary>Almacena su propia y única instancia </summary>

```

```

Private Shared ReadOnly instanciaDimensionar As New Dimensionar()
'''<summary> Gestor de Informes</summary>
Private sLineas As ServicioLineas
'''<summary> Gestor de Máquinas (catálogo)</summary>
Private sMaquinas As ServicioMaquinas
''' <summary>Variable para almacenar temporalmente la familia en la que escoger Máquinas</summary>
Private familia As FamiliaMaquina
''' <summary>Lista para almacenar temporalmente las máquinas que cumplen prod>condiciones(0) </summary>
Private listaCumplenProduccion As List(Of Maquina)
''' <summary>'Lista para almacenar temporalmente las máquinas ordenadas según condiciones(1) y (2)</summary>
Private listaOrdenados As List(Of Maquina)
''' <summary>Lista para almacenar las máquinas finalmente seleccionados y devolverla </summary>
Private lineaDimensionada As List(Of Maquina)

''' <summary>
''' Constructor de la clase. Inicializa las listas a utilizar durante los cálculos
''' Es privado para evitar crear nuevas instancias desde fuera de la propia clase (Singleton)
''' </summary>
Private Sub New()
sMaquinas = ServicioMaquinas.Instance 'La instancia (única) del servicioMaquinas para gestionar los
modelos
sLineas = ServicioLineas.Instance 'La instancia (única) del servicioInformes para gestionar los Informes
listaCumplenProduccion = New List(Of Maquina) 'Inicio las listas
listaOrdenados = New List(Of Maquina)
lineaDimensionada = New List(Of Maquina)
End Sub

''' <summary>
''' Método para obtener la instancia de este Singleton
''' </summary>
''' <returns></returns>
Public Shared ReadOnly Property Instance As Dimensionar
Get
Return instanciaDimensionar
End Get
End Property

''' <summary>
''' Función que realiza la selección de Máquinas, una por cada Familia demáquina requerida
''' y cumpliendo las condiciones impuestas
''' Devolviendo un listado con las máquinas resultado
''' </summary>
Public Sub Dimensiona()
lineaDimensionada.Clear() 'Limpio la lista
For Each id In sLineas.Linea.SeleccionFam 'Para cada familia, representado por su ID en seleccionFam
listaCumplenProduccion.Clear() 'Vacío las listas para el siguiente ciclo
listaOrdenados.Clear()
familia = sMaquinas.GetFamiliaID(id) 'Tomo la familia de máquinas
'De su lista de máquinas, tomo aquellas que cumplen prod>condiciones.produccionMin
'y los copio a la lista local listaCumplenProduccion
For Each maq In familia.Maquinas 'Recorro todas las máquinas de la familia
'Compruebo la producción con 1 ud.a N°paralelismo dado
For i As Integer = 1 To sLineas.Linea.Condiciones.Paralelismo
'y tomo el primero que cumpla
If maq.Produccion * i >= sLineas.Linea.Condiciones.ProduccionMin Then
If i = 1 Then 'Si con paralelismo=1 cumple,
listaCumplenProduccion.Add(maq) 'añado esa máquina
Else 'Si se necesitan más máquinas para cumplir (paralelismo >1)
listaCumplenProduccion.Add(CreaNuevaMaquinaParalela(maq, i)) 'añado nueva maq
End If
End If
Next
Next
If listaCumplenProduccion.Count = 0 Then 'Si ninguna cumplió la condición de producción:
'Añado una Máquina virtual informando de que no se ha encontrado.
listaCumplenProduccion.Add(New Maquina("No se ha encontrado Modelo Funcional que cumpla la
producción requerida", "Clase: " & familia.Nombre))
End If
'Obtengo los nombres de los atributos para condicion1 y condicion2 según los valores de
'minimizarPrimaria y minimizarSecundaria de condiciones
'Si se han seleccionado dos condiciones
If sLineas.Linea.Condiciones.MinimizarPrimaria IsNot Nothing AndAlso
sLineas.Linea.Condiciones.MinimizarSecundaria IsNot Nothing Then
Dim condicion1 As String = sLineas.Linea.Condiciones.MinimizarPrimaria 'Asigno las condiciones
Dim condicion2 As String = sLineas.Linea.Condiciones.MinimizarSecundaria
'meto en listaOrdenados los objetos de listaCumplenProduccion ordenados por las condiciones
listaOrdenados = listaCumplenProduccion _
.OrderBy(Function(obj) obj.GetType().GetProperty(condicion1).GetValue(obj)) _
.ThenBy(Function(obj) obj.GetType().GetProperty(condicion2).GetValue(obj)).ToList()

```

```

        'Tomo la primera máquina de esa lista (listaOrdenados(0)) y la añado a la lista
        'lineaDimensionada de máquinas seleccionadas finalmente
        lineaDimensionada.Add(listaOrdenados(0))
        'Si se ha seleccionado sólo una 1ª condición
    ElseIf sLineas.Linea.Condiciones.MinimizarPrimaria IsNot Nothing Then
        Dim condicion1 As String = sLineas.Linea.Condiciones.MinimizarPrimaria 'Asigno la condicion
        'meto en listaOrdenados los objetos de listaCumplenProduccion ordenados por la condición
        listaOrdenados = listaCumplenProduccion _
            .OrderBy(Function(obj) obj.GetType().GetProperty(condicion1).GetValue(obj)).ToList()
        'Tomo la primera máquina de esa lista (listaOrdenados(0)) y la añado a la lista
        'lineaDimensionada de máquinas seleccionadas finalmente
        lineaDimensionada.Add(listaOrdenados(0))
    Else
        'Si las condiciones eran nothing, no se ha seleccionado condición.
        'Tomo el primero de la lista tal cual
        lineaDimensionada.Add(listaCumplenProduccion(0))
    End If
Next
sLineas.Linea.seleccionMq = lineaDimensionada 'Anoto la lista lineaDimensionada en la Linea
End Sub

''' <summary>
''' Función que, dado una máquina y un nº de paralelismo, devuelve una nueva Máquina
''' con sus atributos multiplicados por ese número
''' </summary>
''' <param name="maquina">La máquina que se va a utilizar i veces en paralelo</param>
''' <param name="i">El número de veces que se utilizará la máquina en paralelo</param>
''' <returns>Una nueva Máquina representando el conjunto de i máquinas originales en paralelo</returns>
Private Function CreaNuevaMaquinaParalela(maquina As Maquina, i As Integer) As Maquina
    'Creo la nueva máquina con el nombre y descripción modificados y sus atributos multiplicados por i
    Dim nuevoMq As New Maquina($"[{i} unidades de: ] {maquina.Nombre}",
        $"Conjunto de {i} unidades instaladas en paralelo.
        Se muestran valores para el conjunto. {Environment.NewLine}
        {maquina.Descripcion}") With {
        .Produccion = i * maquina.Produccion,
        .PotenciaElectrica = i * maquina.PotenciaElectrica,
        .Operarios = i * maquina.Operarios,
        .ValorCoste = i * maquina.ValorCoste,
        .ValorVenta = i * maquina.ValorVenta
    }
    Return nuevoMq 'Devuelvo el nuevo Modelo Funcional
End Function
End Class

```

B.5 Clase Informe

```

Imports System.Reflection

''' <summary>
''' Clase para instanciar los informes que la aplicación irá generando
''' Servirá para almacenar los datos que se incluirán en el informe
''' y para formatear el texto antes de mostrarlo y enviarlo a impresión.
''' La fuente de los datos será la Linea actual en la aplicación.
''' </summary>

Public Class Informe
    '''<summary> Gestor de Lineas</summary>
    Private sLineas As ServicioLineas
    ''' <summary>RichTextBox donde construyo el texto del informe para darle formato</summary>
    Private _contenido As RichTextBox
    ''' <summary>Titulo del informe </summary>
    Private Const titulo As String = "Dimensionamiento preliminar de linea de producción" & vbCrLf &
        "Borrador de Presupuesto Inicial"
    ''' <summary> Producción total de la linea dimensionada</summary>
    Private prodTotal As Single
    ''' <summary> Potencia Eléctrica Total de la linea dimensionada</summary>
    Private PETotal As Single
    ''' <summary> Operarios Totales necesarios</summary>
    Private OperTotal As Single
    ''' <summary> Valor de Coste Total de la linea dimensionada</summary>
    Private VCTotal As Single
    ''' <summary> Valor de Venta total de la linea de la linea dimensionada</summary>
    Private VVTotAs Single
    ''' <summary>Guarda la Versión de esta propia aplicación </summary>
    Private versionApp As String = Assembly.GetExecutingAssembly().GetName().Version.ToString()
    ''' <summary> String con el texto para el pie de página del informe </summary>

```

```

Private Const pieDePagina As String = "Informe generado por la aplicación DraftLine. "
'''<summary> Márgen izquierdo del texto </summary>
Private Const margenIzq As Integer = 20
'''<summary> Márgen derecho del texto </summary>
Private Const margenDch As Integer = 10
'''<summary> Booleano para definir si se quiere mostrar o no el valor de coste </summary>
Private _valorCosteVisible As Boolean = True

'''<summary>
''' Constructor de la clase.
''' Inicializa las variables.
'''</summary>
Public Sub New()
    Contenido = New RichTextBox
    sLineas = ServicioLineas.Instance 'La instancia (única) del servicioInformes para gestionar los Informes
End Sub

'''<summary>
''' Método que genera el contenido del richtextbox, añadiendo paso a paso los datos con formato.
'''</summary>
Public Sub CreaInforme()
    ' NOTA: Ver
    'https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.richtextbox?view=windowsdesktop-8.0
    '----- Encabezado (título etc) -----
    Contenido.Clear() 'Limpio el contenido del richTextBox donde formateo el informe
    Contenido.SelectionFont = New Font(Contenido.Font.FontFamily, 14, FontStyle.Bold) 'Fuente para el título
    Contenido.SelectionAlignment = HorizontalAlignment.Center 'Texto Centrado
    Contenido.AppendText(título & vbCrLf & vbCrLf) 'Título y dos saltos de línea para separar
    Contenido.SelectionFont = New Font(Contenido.Font, FontStyle.Regular) 'Fuente normal
    Contenido.SelectionAlignment = HorizontalAlignment.Left 'Vuelvo a alineación normal
    Separador() 'Separador
    '----- Sección datos línea -----
    TituloSeccion("Datos de la línea:" & vbCrLf) 'Encabezado de los datos de la línea y salto de página
    TextoEnNegrita("Nombre: " & vbCrLf)
    Contenido.AppendText(vbTab & sLineas.Linea.Nombre & vbCrLf)
    TextoEnNegrita("Cliente: " & vbCrLf)
    Contenido.AppendText(vbTab & sLineas.Linea.Cliente & vbCrLf)
    TextoEnNegrita("Proyecto: " & vbCrLf)
    Contenido.AppendText(vbTab & sLineas.Linea.Proyecto & vbCrLf)
    TextoEnNegrita("Versión: " & vbCrLf)
    Contenido.AppendText(vbTab & sLineas.Linea.Version & vbCrLf)
    TextoEnNegrita("Descripción: " & vbCrLf)
    Contenido.AppendText(vbTab & sLineas.Linea.Descripcion & vbCrLf)
    Contenido.AppendText(vbCrLf) 'Salto de línea para separar
    Contenido.SelectionAlignment = HorizontalAlignment.Right 'Texto a la derecha
    Contenido.AppendText("Fecha de creación: " & sLineas.Linea.Fecha & vbCrLf) 'Fecha
    Contenido.SelectionAlignment = HorizontalAlignment.Left 'Vuelvo a alineación normal
    Contenido.AppendText(vbCrLf)
    Separador() 'Separador
    '----- Sección condiciones -----
    TituloSeccion("Condiciones para el dimensionamiento:" & vbCrLf) 'Encabezado de las condiciones
    TextoEnNegrita("Esquema de la línea: " & vbCrLf)
    Contenido.SelectionAlignment = HorizontalAlignment.Center 'Texto Centrado
    Contenido.AppendText(" " & vbCrLf) 'Relleno a los lados
    Clipboard.SetImage(sLineas.Linea.Eschema) 'Copio el esquema al portapapeles
    Contenido.Paste() 'Lo pego en el texto
    Contenido.AppendText(" " & vbCrLf) 'Relleno a los lados
    Contenido.SelectionAlignment = HorizontalAlignment.Left 'Vuelvo a alineación normal
    Contenido.AppendText(vbCrLf)
    TextoEnNegrita("Condiciones seleccionadas: " & vbCrLf) 'Encabezado condiciones
    If sLineas.Linea.Condiciones IsNot Nothing Then
        Contenido.AppendText(vbTab & "Producción mínima requerida: " & vbCrLf & vbCrLf &
            sLineas.Linea.Condiciones.ProduccionMin.ToString() & "kg/h" & vbCrLf)
        Contenido.AppendText(vbTab & "Máx. Máquinas en paralelo: " & vbCrLf & vbCrLf &
            sLineas.Linea.Condiciones.Paralelismo.ToString() & vbCrLf)
        If sLineas.Linea.Condiciones.MinimizarPrimaria IsNot Nothing Then
            Contenido.AppendText(vbTab & "Optimizar minimizando: " & vbCrLf & vbCrLf &
                sLineas.Linea.Condiciones.MinimizarPrimaria & vbCrLf)
            If sLineas.Linea.Condiciones.MinimizarSecundaria IsNot Nothing Then
                Contenido.AppendText(vbTab & "y, de manera secundaria, minimizar: " & vbCrLf &
                    sLineas.Linea.Condiciones.MinimizarSecundaria & vbCrLf)
            End If
        End If
    Else
        Contenido.AppendText(vbTab & "No se han indicado ningún atributo a minimizar para la selección." & vbCrLf)
        Contenido.AppendText(vbTab & "La selección de Modelos Funcionales será aleatoria entre los que cumplan " & vbCrLf &
            "los valores de producción requerida." & vbCrLf)
    End If
    Else
        Contenido.AppendText("Aún no se han indicado condiciones." & vbCrLf)
    End Sub

```

```

End If
Separador()
'----- Sección máquinas seleccionadas -----
TituloSeccion("Máquinas seleccionadas:" & vbCrLf) 'Encabezado Máquinas seleccionadas
For Each modelo In sLineas.Linea.seleccionMq 'Muestro los datos de cada máquina seleccionada
    If valorCosteVisible Then
        Contenido.AppendText(modelo.ToString() & vbCrLf & vbCrLf) 'Muestra el valor de coste
    Else
        Contenido.AppendText(modelo.ToStringNoCoste() & vbCrLf & vbCrLf) 'No muestra el valor de coste
    End If
Next
Contenido.AppendText(vbCrLf)
Separador() 'Separador
'----- Sección Resumen selección -----
TituloSeccion("Resumen características de línea dimensionada:" & vbCrLf) 'Encabezado de Resumen
If sLineas.Linea.seleccionMq.Count <= 0 Then
    TextoEnNegrita("No hay resultados")
Else
    CalculaTotales()
    TextoEnNegrita("Producción máxima: ".PadRight(30))
    Contenido.AppendText(vbTab & prodTotal & " kg/h" & vbCrLf)
    TextoEnNegrita("Potencia eléctrica total: ".PadRight(30))
    Contenido.AppendText(vbTab & PETotal & " kw" & vbCrLf)
    TextoEnNegrita("Operarios necesarios: ".PadRight(30))
    Contenido.AppendText(vbTab & OperTotal & " px" & vbCrLf)
    Contenido.AppendText(vbCrLf)
    Contenido.SelectionAlignment = HorizontalAlignment.Right 'Texto a la derecha
    If valorCosteVisible Then 'Si se debe mostrar el valor de coste
        TextoEnNegrita("Valor de Coste de la línea: ".PadRight(30))
        Contenido.AppendText(vbTab & VCTotal & " €" & vbCrLf)
    End If
    TextoEnNegrita("Valor de Venta de la línea: ".PadRight(30))
    Contenido.AppendText(vbTab & VVTotal & " €" & vbCrLf)
    Contenido.SelectionAlignment = HorizontalAlignment.Left 'Vuelvo a alineación normal
End If
Separador() 'Separador
'----- Pie de página informe -----
Contenido.SelectionFont = New Font(Contenido.Font.FontFamily, 8, FontStyle.Bold) 'Fuente negrita pequeña
Contenido.SelectionAlignment = HorizontalAlignment.Right 'Texto a la derecha
Contenido.AppendText(pieDePagina & versionApp) 'Pie de página
'----- Ediciones generales del texto -----
Margenes() 'Edito márgenes del texto
End Sub

''' <summary>
''' Método que aplica los márgenes al texto del informe
''' </summary>
Private Sub Margenes()
    Contenido.SelectAll() 'Selecciono todo el texto del richTextBox
    Contenido.SelectionIndent = margenIzq 'Aplico el margen izquierdo a todo el texto
    Contenido.SelectionRightIndent = margenDch 'Aplico el margen derecho
End Sub

''' <summary>
''' Método que añade un separador al texto del informe
''' </summary>
Private Sub Separador()
    Contenido.SelectionFont = New Font(Contenido.Font, FontStyle.Regular) 'Fuente normal
    Contenido.SelectionColor = Color.Black 'Color negro
    Contenido.SelectionAlignment = HorizontalAlignment.Center 'Texto Centrado
    Contenido.AppendText(New String("_", 70) & vbCrLf & vbCrLf) 'Separador en forma de guiones bajos
    Contenido.SelectionAlignment = HorizontalAlignment.Left 'Vuelvo a alineación normal
End Sub

''' <summary>
''' Método que encapsula el formato del título de sección, para usar en el texto del informe
''' </summary>
''' <param name="tituloSeccion">El texto a poner en formato título de sección</param>
Private Sub TituloSeccion(tituloSeccion As String)
    Contenido.SelectionFont = New Font(Contenido.Font.FontFamily, 10, FontStyle.Bold) 'Cambio fuente
    Contenido.SelectionColor = Color.DarkBlue 'Cambio color
    Contenido.AppendText(tituloSeccion & vbCrLf) 'Añado título
    Contenido.SelectionFont = New Font(Contenido.Font, FontStyle.Regular) 'Vuelvo a fuente normal
    Contenido.SelectionColor = Color.Black 'Vuelvo a color negro
    Contenido.SelectionAlignment = HorizontalAlignment.Left 'Vuelvo a alineación normal
End Sub

''' <summary>
''' Método que encapsula el formato en negrita, para usar en el texto del informe
''' </summary>

```

```

''' <param name="texto">El texto a poner en negrita</param>
Private Sub TextoEnNegrita(texto As String)
    Contenido.SelectionFont = New Font(Contenido.Font, FontStyle.Bold) 'Cambio a negrita
    Contenido.AppendText(texto) 'Añado texto que irá en negrita
    Contenido.SelectionFont = New Font(Contenido.Font, FontStyle.Regular) 'Vuelvo a fuente normal
End Sub

''' <summary>
''' Método que realiza el cálculo de los valores totales resumen para la línea dimensionada
''' La producción total de la línea será la del componente (máquina o máquinas en paralelo) que
''' tenga la menor producción.
''' El resto de propiedades, serán la suma total
''' </summary>
Private Sub CalculaTotales()
    'Pongo todo a cero para hacer el cálculo completo
    prodTotal = 0 'Producción de la línea dimensionada
    PETotal = 0 'Potencia Eléctrica Total
    OperTotal = 0 'Operarios Totales necesarios
    VCTotal = 0 'Valor de Coste Total
    VVTotal = 0 'Valor de Venta total de la línea
    For Each modelo In sLineas.Linea.seleccionMq 'Recorro cada Máquina de las seleccionadas
        'Tomo la producción menor de todas
        If prodTotal = 0 Or prodTotal >= modelo.Produccion Then prodTotal = modelo.Produccion
        PETotal = PETotal + modelo.PotenciaElectrica 'La suma de cada característica de todos los modelos
        OperTotal = OperTotal + modelo.Operarios
        VCTotal = VCTotal + modelo.ValorCoste
        VVTotal = VVTotal + modelo.ValorVenta
    Next
End Sub

''' <summary>RichTextBox donde construyo el texto del informe para darle formato</summary>
Public Property Contenido As RichTextBox
    Get
        Return _contenido
    End Get
    Set(value As RichTextBox)
        _contenido = value
    End Set
End Property

''' <summary> Booleano para definir si se quiere mostrar o no el valor de coste </summary>
Public Property valorCosteVisible As Boolean
    Get
        Return _valorCosteVisible
    End Get
    Set
        _valorCosteVisible = Value
    End Set
End Property

End Class

```

B.6 Clase FamiliaMaquina

```

''' <summary>
''' Clase que representa las Familias de máquinas. Posee los atributos que reflejan las
''' características de cada Familia de Máquinas, así como una lista para albergar las
''' Máquinas que pertenezcan a esta Familia.
''' </summary>
''' <remarks>Serializable para poder realizar los guardados y cargados de archivo.
''' </remarks>

<Serializable> 'Para poder cargar y guardar
Public Class FamiliaMaquina
    ''' <summary>Identificador de la familia</summary>
    Private _ID As Guid
    ''' <summary>Nombre de la familia</summary>
    Private _nombre As String
    ''' <summary>Descripción de la familia</summary>
    Private _descripcion As String
    ''' <summary>Lista de Máquinas asociadas a est Familia</summary>
    Private _maquinas As List(Of Maquina)

    ''' <summary>
    ''' Constructor de la clase
    ''' </summary>

```

```

''' <remarks>Sin parámetros para permitir la serialización XML</remarks>
Public Sub New()
End Sub

''' <summary>
''' Constructor. Creo una nueva Instancia de la clase, con el nombre dado
''' </summary>
''' <param name="nombre">El nombre de la Familia de máquinas</param>
''' <param name="descripcion">Descripción de la Familia de máquinas</param>
Public Sub New(nombre As String, descripcion As String)
    _ID = Guid.NewGuid() 'Creo un nuevo GUID único
    _nombre = nombre 'Asigno nombre
    _descripcion = descripcion 'Asigno descripción
    _maquinas = New List(Of Maquina) 'Creo la lista de Máquinas, inicialmente vacía.
End Sub

''' <summary>
''' Método para añadir máquinas a la lista de esta familia
''' </summary>
''' <param name="maquina">La máquina a añadir</param>
Public Sub AddMaquina(maquina As Maquina)
    _maquinas.Add(maquina) 'La máquina pasada por argumento se introduce en la lista
End Sub

''' <summary>
''' Función ToString sobrescrita, para obtener el Nombre en texto
''' </summary>
''' <returns> Nombre </returns>
Public Overrides Function ToString() As String
    Return _nombre
End Function

''' <summary>Muestra o modifica el identificador de la familia</summary>
''' <returns>El identificador de la familia</returns>
Public Property ID As Guid
    Get
        Return _ID
    End Get
    Set(value As Guid) 'Debo mantener el setter ya que si no no funciona correctamente el guardado y cargado
        _ID = value
    End Set
End Property

''' <summary>Muestra o modifica el nombre de la familia</summary>
''' <returns>El nombre de la familia</returns>
Public Property Nombre As String
    Get
        Return _nombre
    End Get
    Set(value As String)
        _nombre = value
    End Set
End Property

''' <summary>Muestra o modifica la descripción de la familia</summary>
''' <returns>La descripción de la familia</returns>
Public Property Descripcion As String
    Get
        Return _descripcion
    End Get
    Set(value As String)
        _descripcion = value
    End Set
End Property

''' <summary>Muestra o modifica la lista de Máquinas asociadas a este familia</summary>
''' <returns>La lista de Máquinas</returns>
Public Property Maquinas As List(Of Maquina)
    Get
        Return _maquinas
    End Get
    Set(value As List(Of Maquina))
        _maquinas = value
    End Set
End Property
End Class

```

B.7 Clase Máquina

```

''' <summary>
''' Clase que representa las Máquinas. Posee los atributos que reflejan las características
''' de cada Máquina (nombre, descripción, Producción, Operarios, Potencia Eléctrica, Valor de
''' Coste y Valor de Venta.
''' También dispone de los métodos para actualizar dichos atributos así como para mostrarlos en texto,
''' con o sin el Valor de Coste
''' </summary>
''' <remarks>Serializable para poder realizar los guardados y cargados de archivo
''' </remarks>

<Serializable> 'Para poder cargar y guardar
Public Class Máquina
    ''' <summary>Identificador de la Máquina</summary>
    Private _ID As Guid
    ''' <summary>Nombre de la Máquina</summary>
    Private _nombre As String
    '''<summary>Descripción de la Máquina</summary>
    Private _descripcion As String
    '''<summary>Valor de Producción</summary>
    Private _produccion As Single
    '''<summary>Valor Operarios necesarios</summary>
    Private _operarios As Single
    '''<summary>Valor de Potencia eléctrica de la Máquina</summary>
    Private _potenciaElectrica As Single
    '''<summary>Valor de Coste económico</summary>
    Private _valorCoste As Single
    '''<summary>Valor de Venta de la Máquina </summary>
    Private _valorVenta As Single

    ''' <summary>
    ''' Constructor de la clase
    ''' </summary>
    ''' <remarks>Sin parámetros para permitir la serialización XML</remarks>
    Public Sub New()
    End Sub

    ''' <summary>
    ''' Constructor, Crea una nueva Instancia de la clase, con el nombre dado
    ''' </summary>
    ''' <param name="nombre">El nombre de la Máquina</param>
    ''' <param name="descripcion">La descripción de la Máquina</param>
    Public Sub New(nombre As String, descripcion As String)
        _ID = Guid.NewGuid() ' Creo un nuevo GUID único
        _nombre = nombre 'Asigno el nombre
        _descripcion = descripcion 'Asigno la descripción
    End Sub

    ''' <summary>El identificador de la Máquina</summary>
    Public Property ID As Guid
    Get
        Return _ID
    End Get
    Set(value As Guid)
        _ID = value
    End Set
End Property

    ''' <summary>El nombre de la Máquina</summary>
    Public Property Nombre As String
    Get
        Return _nombre
    End Get
    Set(value As String)
        _nombre = value
    End Set
End Property

    ''' <summary>La descripción de la Máquina</summary>
    Public Property Description As String
    Get
        Return _descripcion
    End Get
    Set(value As String)
        _descripcion = value
    End Set
End Property

```

```

'''<summary>La producción de la Máquina </summary>
Public Property Produccion As Single
    Get
        Return _produccion
    End Get
    Set(value As Single)
        _produccion = value
    End Set
End Property

'''<summary>Los operarios necesarios para hacer funcionar la Máquina </summary>
Public Property Operarios As Single
    Get
        Return _operarios
    End Get
    Set(value As Single)
        _operarios = value
    End Set
End Property

'''<summary>La potencia eléctrica de la Máquina </summary>
Public Property PotenciaElectrica As Single
    Get
        Return _potenciaElectrica
    End Get
    Set(value As Single)
        _potenciaElectrica = value
    End Set
End Property

'''<summary>El coste económico, de fabricación o compra, de la Máquina </summary>
Public Property ValorCoste As Single
    Get
        Return _valorCoste
    End Get
    Set(value As Single)
        _valorCoste = value
    End Set
End Property

'''<summary>El valor de venta de la Máquina </summary>
Public Property ValorVenta As Single
    Get
        Return _valorVenta
    End Get
    Set(value As Single)
        _valorVenta = value
    End Set
End Property

''' <summary>
''' Método que recibe como argumento una lista de valores para asignar a las propiedades de la Máquina
''' </summary>
''' <param name="listaPropiedades">Lista de valores de los atributos de la Máquina,
''' almacenados como Single</param>
Public Sub AsignaPropiedades(listaPropiedades As List(Of Single))
    _operarios = listaPropiedades(0)
    _potenciaElectrica = listaPropiedades(1)
    _produccion = listaPropiedades(2)
    _valorCoste = listaPropiedades(3)
    _valorVenta = listaPropiedades(4)
End Sub

''' <summary>
''' Función ToString para obtener una representación en texto de la Máquina
''' </summary>
''' <returns> Representación en texto de la Máquina </returns>
Public Overrides Function ToString() As String
    Return "NOMBRE:".PadRight(20) & _nombre & vbCrLf & vbCrLf & vbTab &
        "DESCRIPCIÓN:".PadRight(20) & _descripcion & vbCrLf & vbCrLf & vbTab &
        "PRODUCCIÓN:".PadRight(20) & $"{vbTab & _produccion} Kg/h" & vbCrLf & vbTab &
        "POT. ELÉCTRICA:".PadRight(20) & $"{vbTab & _potenciaElectrica} Kw" & vbCrLf & vbTab &
        "OPERARIOS:".PadRight(20) & $"{vbTab & _operarios}" & vbCrLf & vbTab &
        "PRECIO COSTE:".PadRight(20) & $"{vbTab & _valorCoste} €" & vbCrLf & vbTab &
        "PRECIO VENTA:".PadRight(20) & $"{vbTab & _valorVenta} €" & vbCrLf & vbTab
End Function

''' <summary>
''' Función ToString para obtener una representación en texto de la Máquina, pero sin mostrar el valor de
coste

```

```

''' </summary>
''' <returns> Representación en texto de la Máquina sin valor de coste </returns>
Public Function ToStringNoCoste() As String
    Return "NOMBRE:".PadRight(20) & _nombre & vbCrLf & vbCrLf & vbCrLf & vbCrLf &
        "DESCRIPCIÓN:".PadRight(20) & _descripcion & vbCrLf & vbCrLf & vbCrLf & vbCrLf &
        "PRODUCCIÓN:".PadRight(20) & $"{vbTab & _produccion} Kg/h" & vbCrLf & vbCrLf &
        "POT.ELÉCTRICA:".PadRight(20) & $"{vbTab & _potenciaElectrica} Kw" & vbCrLf & vbCrLf &
        "OPERARIOS:".PadRight(20) & $"{vbTab & _operarios}" & vbCrLf & vbCrLf &
        "PRECIO VENTA:".PadRight(20) & $"{vbTab & _valorVenta} €" & vbCrLf & vbCrLf
End Function
End Class

```

B.8 Clase Linea

```

Imports System.Xml.Serialization

''' <summary>
''' Clase que representa la linea a dimensionar. Contendrá los atributos que definen sus
''' propiedades, tales como los datos identificativos a mostrar en el informe de la propia
''' linea (Nombre, cliente, proyecto, version, descripcion), la fecha en que se creó, las
''' condiciones de dimensionamiento seleccionadas para esta línea, la selección de Familias
''' de máquinas que forman su estructura y la selección de Máquinas resultado del
''' dimensionamiento.
''' Además, dispone de una variable para almacenar las familias utilizadas en ella en caso
''' de que se exporte a un fichero.
''' </summary>
''' <remarks>Serializable para poder realizar los guardados y cargados de archivo.
''' </remarks>

<Serializable> 'Para poder cargar y guardar
Public Class Linea
    'Serializables
    ''' <summary> Datos informativos de la linea a dimensionar </summary>
    Private _nombre As String = "Línea de producción genérica"
    Private _cliente As String = "Cliente sin definir"
    Private _proyecto As String = "Proyecto genérico"
    Private _version As String = "1.0"
    Private _descripcion As String = "No se ha indicado descripción de la línea"
    ''' <summary> Las condiciones que se han tomado para la selección </summary>
    Private _condiciones As Condiciones = New Condiciones("nuevo")
    ''' <summary> La fecha de creación del informe </summary>
    Private _fecha As String = Date.Now.ToString("dd/MM/yyyy") 'Asigno la fecha actual
    ''' <summary> Lista de las máquinas seleccionadas </summary>
    Private _seleccionMq As List(Of Maquina) = New List(Of Maquina)
    ''' <summary> Lista que almacenará las ID de las familias seleccionados para dimensionar </summary>
    Private _seleccionFam As List(Of Guid) = New List(Of Guid)
    ''' <summary> Lista que almacenará las familias de máquinas usadas en esta linea en caso de que
    ''' se quiera guardar en fichero para poder recuperarlos en un futuro </summary>
    Private _familiasLinea As List(Of FamiliaMaquina) = New List(Of FamiliaMaquina)

    'No serializable
    ''' <summary> Imagen con el esquema de las familias de máquinas escogidas </summary>
    <XmlIgnore> Private _esquema As Image = New Bitmap(400, 50) 'Inicialmente, imagen por defecto

    ''' <summary>
    ''' Constructor de la clase
    ''' </summary>
    ''' <remarks> Sin parámetros para permitir la serialización XML </remarks>
    Public Sub New()
    End Sub

    ''' <summary> Lista de las máquinas seleccionadas </summary>
    Public Property seleccionMq As List(Of Maquina)
        Get
            Return _seleccionMq
        End Get
        Set(value As List(Of Maquina))
            _seleccionMq = value
        End Set
    End Property

    ''' <summary> El nombre de la línea que se está dimensionando (Dato de línea) </summary>
    Public Property Nombre As String
        Get

```

```

        Return _nombre
    End Get
    Set(value As String)
        _nombre = value
    End Set
End Property

''' <summary>El Cliente de la línea que se está dimensionando (Dato de línea) </summary>
Public Property Cliente As String
    Get
        Return _cliente
    End Get
    Set(value As String)
        _cliente = value
    End Set
End Property

''' <summary>El nombre del proyecto al que pertenece la línea que se está dimensionando
''' (Dato de línea) </summary>
Public Property Proyecto As String
    Get
        Return _proyecto
    End Get
    Set(value As String)
        _proyecto = value
    End Set
End Property

''' <summary>La versión de la línea que se está dimensionando (Dato de línea) </summary>
Public Property Version As String
    Get
        Return _version
    End Get
    Set(value As String)
        _version = value
    End Set
End Property

''' <summary>La descripción de la línea que se está dimensionando (Dato de línea) </summary>
Public Property Descripcion As String
    Get
        Return _descripcion
    End Get
    Set(value As String)
        _descripcion = value
    End Set
End Property

''' <summary> Las condiciones que se han tomado para la selección </summary>
Public Property Condiciones As Condiciones
    Get
        Return _condiciones
    End Get
    Set(value As Condiciones)
        _condiciones = value
    End Set
End Property

''' <summary>La fecha de creación del informe</summary>
Public Property Fecha As String
    Get
        Return _fecha
    End Get
    Set(value As String)
        _fecha = value
    End Set
End Property

''' <summary> Imagen con el esquema de las familias de máquinas escogidas</summary>
<XmlIgnore> Public Property Esquema As Image
    Get
        Return _esquema
    End Get
    Set(value As Image)
        _esquema = value
    End Set
End Property

''' <summary>Lista que almacenará las ID de las familias seleccionados para dimensionar</summary>
Public Property SeleccionFam As List(Of Guid)

```

```

    Get
        Return _seleccionFam
    End Get
    Set
        _seleccionFam = Value
    End Set
End Property

''' <summary>Lista que almacenará las familias de máquinas usadas en esta línea en caso de que
''' se quiera guardar en fichero para poder recuperarlos en un futuro </summary>
Public Property FamiliasLinea As List(Of FamiliaMaquina)
    Get
        Return _familiasLinea
    End Get
    Set
        _familiasLinea = Value
    End Set
End Property
End Class

```

B.9 Clase Condiciones

```

''' <summary>
''' Clase para encapsular las condiciones de dimensionamiento, elegidas por el usuario, para una línea en
''' particular.
''' Formará una característica de esa línea y será utilizada para el cálculo de dimensionamiento.
''' Cuenta con las variables para guardar: La producción mínima requerida, el paralelismo (cant máx de máquinas
''' en paralelo para conseguir la producción mínima) y las características a minimizar en la selección.
''' </summary>
'''
<Serializable>
Public Class Condiciones
    Private _paralelismo As Integer
    Private _produccionMin As Single
    Private _minimizarPrimaria As String
    Private _minimizarSecundaria As String

    ''' <summary>
    ''' Constructor por defecto, necesario para la serialización.
    ''' </summary>
    Public Sub New()
    End Sub

    ''' <summary>
    ''' Constructor para ser llamado desde Linea. Inicializa todo por defecto
    ''' </summary>
    Public Sub New(nombre As String)
        'Inicializo todo por defecto
        _paralelismo = 1
        _produccionMin = 0
        _minimizarPrimaria = Nothing
        _minimizarSecundaria = Nothing
    End Sub

    ''' <summary>
    ''' Cantidad de máquinas de la misma familia que se pueden tomar para cumplir la producción
    ''' </summary>
    ''' <returns>paralelismo</returns>
    Public Property Paralelismo As Integer
        Get
            Return _paralelismo
        End Get
        Set
            _paralelismo = Value
        End Set
    End Property

    ''' <summary>
    ''' Producción mínima requerida
    ''' </summary>
    ''' <returns>produccionMin</returns>
    Public Property ProduccionMin As Single
        Get
            Return _produccionMin
        End Get
        Set
            _produccionMin = Value
        End Set
    End Property

```

```

        End Set
    End Property

    ''' <summary>
    ''' Propiedad que se debe minimizar en la selección (primaria)
    ''' </summary>
    ''' <returns>_minimizarPrimaria</returns>
    Public Property MinimizarPrimaria As String
        Get
            Return _minimizarPrimaria
        End Get
        Set
            _minimizarPrimaria = Value
        End Set
    End Property

    ''' <summary>
    ''' Propiedad que se debe minimizar en la selección (secundaria)
    ''' </summary>
    ''' <returns>_minimizarSecundaria</returns>
    Public Property MinimizarSecundaria As String
        Get
            Return _minimizarSecundaria
        End Get
        Set
            _minimizarSecundaria = Value
        End Set
    End Property
End Class

```

B.10 Clase Imprimir

```

Imports System.Drawing.Printing          'Librería para la impresión
Imports System.Runtime.InteropServices    'Librería para las funciones no gestionadas (interactuar con APIWindows)

''' <summary>
''' Clase donde se encapsulo todo lo necesario para realizar la impresión del informe.
''' Toma el RichTextBox de la pantalla principal tal cual, define el área a imprimir,
''' a partir de la configuración de impresión pasada y la va llenando con el contenido
''' del RichTextBox. Cuando completa un área, pero aún no ha acabado el RichTextBox,
''' informa al sistema de impresión de que aún quedan páginas por imprimir y vuelve a
''' rellenar otro área de impresión, hasta que no queda más contenido en el RichTextBox.
''' </summary>
Public Class Imprimir
    ''' <summary> RichTextBox que contendrá el texto formateado a imprimir</summary>
    Private textoParaImprimir As RichTextBox
    ''' <summary>Índice del caracter que se está imprimiendo</summary>
    Private indiceCaracterActual As Integer = 0
    ''' <summary>Constante para mensajes de Windows</summary>
    Private Const EM_FORMATRANGE As Integer = 1081
    ''' <summary>Almacena el rango de caracteres a imprimir </summary>
    Private Structure RANGOCARACTER
        Public cpMin As Integer 'Índice del 1º carácter
        Public cpMax As Integer 'Índice del último carácter
    End Structure
    ''' <summary>Almacena el rango ya formateado</summary>
    Private Structure RANGOFORMATO
        Public hdc As IntPtr      'Manejador del dispositivo de impr.
        Public hdcTarget As IntPtr 'Manejador del dispositivo objetivo
        Public rc As Rectangle     'Área de impresión (en twips)
        Public rcPage As Rectangle 'Área de la página entera (en twips)
        Public chrg As RANGOCARACTER 'Rango de caracteres a imprimir
    End Structure

    ''' <summary>
    ''' Constructor de la clase
    ''' </summary>
    ''' <param name="RichTextBoxTexto">El RichTextBox cuyo contenido imprimiré</param>
    Public Sub New(RichTextBoxTexto As RichTextBox)
        textoParaImprimir = RichTextBoxTexto 'Paso el richTextBox al mío de esta clase
    End Sub

    ''' <summary>
    ''' Método que crea un nuevo doc de impresión, asignando la configuración de la impresora que proviene de
    ''' un printDialog y establece el manejador del evento PrintPage del documento.

```

```

''' Reinicia el índice actual y comienza a imprimir
''' </summary>
''' <param name="configuracionImpresion">La configuración seleccionada para imprimir, desde el
printDialog</param>
Public Sub Imprime(configuracionImpresion As PrinterSettings)
    Using printDoc As New PrintDocument With {
        .PrinterSettings = configuracionImpresion          'Nuevo doc. de impresión
        .PrinterSettings = configuracionImpresion          'Paso la configuración
    }
        AddHandler printDoc.PrintPage, AddressOf PrintPageHandler Asigno el manejador del evento PrintPage
        indiceCaracterActual = 0                          'Reseteo el índice actual
        printDoc.Print()                                  'Inicio la impresión
    End Using
End Sub

''' <summary>
''' Método que maneja el evento de imprimir el documento. Establece el rango de formato y los rectángulos
''' en los que se imprimirá.
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub PrintPageHandler(sender As Object, e As PrintPageEventArgs)
    Dim rangoFormato As New RANGOFORMATO() 'Estructura para definir el rango del formato (ver estructura
en el código)
    Dim rectAImprimir As New Rectangle(e.MarginBounds.Left * 14, e.MarginBounds.Top * 14,
e.MarginBounds.Width * 14, e.MarginBounds.Height * 14) 'Rectángulos de impresión (*14 para pasar de px a twips)
    'twips ver: https://learn.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-
6.0/aa314264(v=vs.60)
    Dim rectPagina As Rectangle = rectAImprimir          'Rectángulo de página entera
    Dim hdc As IntPtr = e.Graphics.GetHdc()             'Tomo el manejador del dispositivo gráfico
    rangoFormato(hdc) = hdc                             'Establezo el manejador a la estructura
    rangoFormato(hdcTarget) = hdc
    rangoFormato(rc) = rectAImprimir                    'Establezco el rectángulo de impresión
    rangoFormato(rcPage) = rectPagina                   'Establezo rectángulo de página entera
    rangoFormato.chrg.cpMin = indiceCaracterActual      'Indico el caracter inicial
    rangoFormato.chrg.cpMax = textoParaImprimir.TextLength 'Indico caracter final
    'Asigno la estructura RANGOFORMATO a un puntero de memoria
    Dim lParam As IntPtr = Marshal.AllocCoTaskMem(Marshal.SizeOf(rangoFormato))
    Marshal.StructureToPtr(rangoFormato, lParam, False)
    Dim res As Integer = SendMessage(textoParaImprimir.Handle, EM_FORMATRANGE, CType(1, IntPtr), lParam)
'Formateo el rango
    Marshal.FreeCoTaskMem(lParam)                       'Libero la memoria asignada
    e.Graphics.ReleaseHdc(hdc)                          'Libero el manejador del disp.gráfico
    If res < textoParaImprimir.TextLength Then          'Si aún hay más para imprimir
        e.HasMorePages = True                          'Indico que hay más páginas pendientes
        indiceCaracterActual = res                      'Paso el índice al actual
    Else                                                'Si no queda nada para imprimir
        e.HasMorePages = False                          'No hay más páginas pendientes
        SendMessage(textoParaImprimir.Handle, EM_FORMATRANGE, CType(0, IntPtr), IntPtr.Zero) 'Reseteo rango
    End If
End Sub

''' <summary>
''' Función para interactuar con la API de Windows
''' Pasará el mensaje a la ventana indicada. El mensaje irá desde un puntero de memoria a otro (Rango)
''' </summary>
''' <param name="hWnd">A donde se enviará el mensaje (al manejador del contenido a imprimir)</param>
''' <param name="msg">El mensaje que se enviará </param>
''' <param name="wParam">Parámetro inicial</param>
''' <param name="lParam">Parámetro final</param>
''' <returns></returns>
<DllImport("user32.dll", CharSet:=CharSet.Auto)> 'Para importar la función SendMessage desde user32.dll en
tiempo de ejecución
Private Shared Function SendMessage(hWnd As IntPtr, msg As Integer, wParam As IntPtr, lParam As IntPtr) As
Integer
    End Function
End Class

```

B.11 Clase FormFamilia

```

''' <summary>
''' Formulario para introducir los datos de la nueva Familia de máquinas a crear o editar una existente
''' </summary>
Public Class FormFamilia
    '''<summary>Variable que almacena el nombre de la Familia, si se trata de una edición</summary>

```

```

Private nombreFamilia As String
'''<summary>La instancia del gestor de catalogo (ServicioMaquinas)</summary>
Private sMaquinas As ServicioMaquinas
'''<summary> Almacena el nombre introducido</summary>
Private _nombre As String
'''<summary> Almacena la descripción introducida</summary>
Private _descripcion As String

''' <summary>
''' Constructor del formulario. Obtiene la instancia del gestor del catálogo y,
''' si se trata de editar una familia existente, toma los valores.
''' </summary>
''' <param name="nombreFamilia">String opcional con el nombre de la Familia a editar</param>
Public Sub New(Optional nombreFamilia As String = Nothing)
    ' Esta llamada es exigida por el diseñador.
    InitializeComponent()
    sMaquinas = ServicioMaquinas.Instance() 'Tomo la instancia de servicioMaquinas
    If nombreFamilia IsNot Nothing Then 'Si paso el nombre de una Familia entonces pongo en modo "Editar"
        Me.Text = "Editar Familia de Máquinas existente" 'Cambio el nombre de la ventana de este formulario
        TextBoxNombre.Text = nombreFamilia 'Recuperamos el nombre de la familia
        TextBoxNombre.ReadOnly = True 'Bloqueo para evitar cambiar el nombre
        TextBoxDescripcion.Text = sMaquinas.GetDescripcion(nombreFamilia) 'Recuperamos la descripción
    End If
End Sub

''' <summary>
''' Método que captura el evento de que el usuario haga click sobre el botón aceptar
''' Si está en "modo edición", actualiza el campo descripción (el nombre será el mismo).
''' Si está en "modo crear", comprueba que se haya introducido un nombre al menos, y no coincida con
''' el nombre de ninguna familia existente en el catálogo.
''' Marca el resultado del diálogo a OK
''' </summary>
''' <param name="sender">El botón aceptar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub Aceptar_Click(sender As Object, e As EventArgs) Handles BttAceptar.Click
    If nombreFamilia IsNot Nothing Then 'Si está en modo "editar"
        _nombre = nombreFamilia 'Tomo el nombre que se ha introducido
        _descripcion = TextBoxDescripcion.Text 'Toma la descripción introducida
    ElseIf TextBoxNombre.Text = "" Then 'Si no se introduce ningún nombre, muestra error
        MessageBox.Show("Se debe introducir un nombre", "Error, falta un nombre", MessageBoxButtons.OK,
            MessageBoxIcon.Error)

        Return
    ElseIf sMaquinas.CompruebaNombre(TextBoxNombre.Text) Then 'Si el nombre ya está utilizado, muestra error
        MessageBox.Show("Ya existe una Familia de máquinas con ese nombre." & vbCrLf &
            "Introduzca un nombre distinto", "Error, coincidencia de nombre",
            MessageBoxButtons.OK, MessageBoxIcon.Error)

        TextBoxNombre.Clear() 'Limpio el TextBox del nombre para que el usuario introduzca uno nuevo
        Return
    Else
        _nombre = TextBoxNombre.Text 'Tomo el nombre que se ha introducido
        _descripcion = TextBoxDescripcion.Text 'Toma la descripción introducida
    End If
    Me.DialogResult = DialogResult.OK 'Marco el resultado del diálogo
    Me.Close() 'Cierro este formulario
End Sub

''' <summary>
''' Método que captura el evento de pulsar sobre el botón cancelar
''' Se pierden los posibles datos introducidos y se cierra el formulario
''' Marca el resultado del diálogo a Cancel
''' </summary>
''' <param name="sender">El botón cancelar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub Cancelar_Click(sender As Object, e As EventArgs) Handles BttCancelar.Click
    Me.DialogResult = DialogResult.Cancel 'Marco el resultado del diálogo y cierro este formulario
    Me.Close()
End Sub

'''<summary> Almacena el nombre introducido</summary>
Public Property Nombre As String
    Get
        Return _nombre
    End Get
    Set
        _nombre = Value
    End Set
End Property

'''<summary> Almacena la descripción introducida</summary>
Public Property Descripcion As String

```

```

        Get
            Return _descripcion
        End Get
        Set
            _descripcion = Value
        End Set
    End Property
End Class

```

B.12 Clase FormMaquina

```

Imports System.Linq.Expressions

''' <summary>
''' Formulario para introducir los datos de la nueva Máquina a crear o editar una existente
''' </summary>
Public Class FormMaquina
    '''<summary>Variable que almacena el nombre de la máquina, si se trata de una edición</summary>
    Private nombreMaquina As String
    '''<summary>La instancia del gestor de catálogo (ServicioMaquinas)</summary>
    Private sMaquinas As ServicioMaquinas
    '''<summary>Variable que almacena la familia destino de esta máquina</summary>
    Private _familia As FamiliaMaquina
    '''<summary>Variable que almacena el ID de la máquina a editar si se trata de una edición</summary>
    Private _ID As Guid
    ''' <summary>Marca que indica si se trata de una edición de una máquina existente o no</summary>
    Private _esEdicion As Boolean
    ''' <summary>Almacena el nombre de la máquina </summary>
    Private _nombre As String
    ''' <summary>Almacena la descripción de la máquina</summary>
    Private _descripcion As String
    ''' <summary>Almacena la lista de propiedades de la máquina</summary>
    Private _listaprop As List(Of Single)

    ''' <summary>
    ''' Constructor del formulario. Obtiene la instancia del gestor de catálogo y,
    ''' si se trata de editar una máquina existente, toma los valores.
    ''' </summary>
    ''' <param name="nombreMaquina">String opcional con el nombre de la máquina a editar</param>
    Public Sub New(familia As FamiliaMaquina, Optional nombreMaquina As String = Nothing)
        ' Esta llamada es exigida por el diseñador.
        InitializeComponent()
        EsEdicion = False
        sMaquinas = ServicioMaquinas.Instance() 'Tomo la instancia de servicioMaquinas
        Me.Familia = familia 'Asigno la familia
        _listaprop = New List(Of Single) 'Inicializo la lista
        If nombreMaquina IsNot Nothing Then 'Si paso nombre de una máquina entonces nos ponemos en modo "Editar"
            EsEdicion = True
            Me.Text = "Editar Máquina existente" 'Cambio el nombre de la ventana de este formulario
            Me.nombreMaquina = nombreMaquina
            TextBoxNombre.Text = nombreMaquina 'Recupero el nombre y el resto de datos y los muestro
            ID = sMaquinas.GetIDMaquina(familia.Nombre, nombreMaquina)
            Dim listAtributos As New List(Of String)
            listAtributos = sMaquinas.GetAtributosMaquina(familia.Nombre, nombreMaquina)
            TextBoxDescripcion.Text = listAtributos(0)
            TextBoxOperadores.Text = listAtributos(1)
            TextBoxPotElect.Text = listAtributos(2)
            TextBoxProduccion.Text = listAtributos(3)
            TextBoxValorCoste.Text = listAtributos(4)
            TextBoxValorVenta.Text = listAtributos(5)
        End If
    End Sub

    ''' <summary>
    ''' Método para controlar que en textBox sólo se introduzcan números
    ''' </summary>
    ''' <param name="sender">textBoxProduccion y textBoxParalelismo</param>
    ''' <param name="e">Teclas pulsadas del teclado en el campo de texto de esos textbox</param>
    Private Sub TextBox_KeyPress(sender As Object, e As KeyEventArgs) Handles TextBoxProduccion.KeyPress,
        TextBoxOperadores.KeyPress, TextBoxPotElect.KeyPress,
        TextBoxValorCoste.KeyPress, TextBoxValorVenta.KeyPress

        If Char.IsDigit(e.KeyChar) Then 'Si la pulsación es un dígito
            e.Handled = False
        ElseIf Char.IsControl(e.KeyChar) Then 'Si la pulsación es una tecla de control
            e.Handled = False
        Else 'Resto de casos

```

```

        e.Handled = True
    End If
End Sub

''' <summary>
''' Método que captura el evento de que el usuario haga click sobre el botón aceptar
''' Comprueba que se ha introducido al menos un nombre, y, si está en modo editar, que ese nombre no
''' exista ya en la Familia de Máquinas destino.
''' Toma todos los valores introducidos y marca el resultado del diálogo a OK
''' </summary>
''' <param name="sender">El botón aceptar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub Aceptar_Click(sender As Object, e As EventArgs) Handles BttAceptar.Click
    'Si no se introduce ningún nombre, muestra error y vuelve
    If TextBoxNombre.Text = "" Then
        MessageBox.Show("Se debe introducir un nombre", "Error, falta un nombre", MessageBoxButtons.OK,
            MessageBoxIcon.Error)

        Return
    End If
    'Si no está en modo Edición y ya existe una máquina, en esa Familia, con ese nombre, da error y vuelve
    If Not EsEdicion AndAlso sMaquinas.CompruebaNombre(TextBoxNombre.Text, Familia) Then
        MessageBox.Show("Ya existe una Máquina con ese nombre." & vbCrLf & "Introduzca un nombre distinto",
            "Error, coincidencia de nombre", MessageBoxButtons.OK, MessageBoxIcon.Error)
        TextBoxNombre.Clear()
        Return
    End If
    'Si nada de lo anterior nos hizo volver, seguimos
    Nombre = TextBoxNombre.Text 'Tomo el nombre que se ha introducido
    Descripcion = TextBoxDescripcion.Text 'Toma la descripción introducida
    For i As Integer = 1 To 5 'relleno la lista de propiedades de 0 (en single)
        Listaprop.Add(0.0F)
    Next
    Try
        'Capturo la excepción de que los valores introducidos no sean válidos para Single
        Listaprop(0) = TextBoxOperadores.Text 'Asigno los valores introducidos
        Listaprop(1) = TextBoxPotElect.Text
        Listaprop(2) = TextBoxProduccion.Text
        Listaprop(3) = TextBoxValorCoste.Text
        Listaprop(4) = TextBoxValorVenta.Text
    Catch ex As Exception 'Si el valor no es válido, capturo, aviso, pongo todo a 0 y no acepto
        MessageBox.Show(ex.ToString() & vbCrLf & "Introduzca de nuevo los valores",
            "Error en los datos introducidos.", MessageBoxButtons.OK, MessageBoxIcon.Error)
        TextBoxOperadores.Text = 0 'Pongo los valores a 0
        TextBoxPotElect.Text = 0
        TextBoxProduccion.Text = 0
        TextBoxValorCoste.Text = 0
        TextBoxValorVenta.Text = 0
        Return
    End Try

    Me.DialogResult = DialogResult.OK 'Marco el resultado del diálogo
    Me.Close() 'Cierro este formulario
End Sub

''' <summary>
''' Método que captura el evento de pulsar sobre el botón cancelar
''' Se pierden los posibles datos introducidos y se cierra el formulario
''' Marca el resultado del diálogo a Cancel
''' </summary>
''' <param name="sender">El botón cancelar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub Cancelar_Click(sender As Object, e As EventArgs) Handles BttCancelar.Click
    Me.DialogResult = DialogResult.Cancel 'Marco el resultado del diálogo y cierro este formulario
    Me.Close()
End Sub

''' <summary>Marca que indica si se trata de una edición de una máquina existente o no</summary>
Public Property EsEdicion As Boolean
    Get
        Return _esEdicion
    End Get
    Set
        _esEdicion = Value
    End Set
End Property

''' <summary>Almacena el nombre de la máquina </summary>
Public Property Nombre As String
    Get
        Return _nombre
    End Get

```

```

        Set
            _nombre = Value
        End Set
    End Property

    ''' <summary>Almacena la descripción de la máquina</summary>
    Public Property Descripcion As String
        Get
            Return _descripcion
        End Get
        Set
            _descripcion = Value
        End Set
    End Property

    ''' <summary>Almacena la lista de propiedades de la máquina</summary>
    Public Property Listaprop As List(Of Single)
        Get
            Return _listaprop
        End Get
        Set
            _listaprop = Value
        End Set
    End Property

    '''<summary>Variable que almacena el ID de la máquina a editar si se trata de una edición</summary>
    Public Property ID As Guid
        Get
            Return _ID
        End Get
        Set
            _ID = Value
        End Set
    End Property

    '''<summary>Variable que almacena la familia destino de esta máquina</summary>
    Public Property Familia As FamiliaMaquina
        Get
            Return _familia
        End Get
        Set
            _familia = Value
        End Set
    End Property
End Class

```

B.13 Clase FormSelecFam

```

Imports System.Drawing.Drawing2D
Imports System.Reflection

''' <summary>
''' Clase con el formulario para que el usuario seleccione las Familias de máquinas que compondrán la Linea,
''' así como su orden. Esa selección se realiza añadiendo Familias de máquinas representadas gráficamente como
''' una caja con su nombre, a un Panel, donde el usuario podrá reorganizarlos arrastrando cada caja a la posición
''' deseada, eliminando la que desee y añadiendo nuevas.
''' Se obtiene un lista con los identificadores de las Familias de máquinas seleccionadas, así como una imagen
''' del esquema formado por esa lista.
''' </summary>
Public Class FormSelecFam
    '''<summary> Gestor de Maquinas (catalogo)</summary>
    Private sMasquinas As ServicioMaquinas
    ''' <summary>Lista de Familias de maquinas seleccionadas </summary>
    Private listaSeleccionFam As New List(Of FamiliaMaquina)
    ''' <summary> Lista de Representaciones de Familias </summary>
    Private listaRepresentaciones As New List(Of RepresentacionFamilia)
    ''' <summary>Tamaño máximo de la selección defamilias. Impuesto para contar con un límite superior
    real</summary>
    Private limiteSeleccion As Integer = 48
    ''' <summary>La lista para almacenar los ID's de las famiaas seleccionados para dimensionar</summary>
    Private _seleccionModFam As New List(Of Guid)
    ''' <summary>Para almacenar la captura en imagen del esquema dibujado en el panel</summary>
    Private _esquema As Image
    ''' <summary>Almacena la separación de la representación entre modelos </summary>
    Private Const separacion As Integer = 35

```

```

''' <summary>Boolean si la representación se está arrastrando</summary>
Private arrastrandoCaja As Boolean = False
''' <summary> Guarda la posición original de la representación que se está arrastrando</summary>
Private posicionOriginalCaja As Point
''' <summary> La representación seleccionada en el panel</summary>
Private cajaSeleccionada As RepresentacionFamilia = Nothing
''' <summary> Longitud de la caja que representará a la familia seleccionada</summary>
Private longitudCaja As Integer = 100
''' <summary>Altura de la caja que representará la familia seleccionad</summary>
Private alturaCaja As Integer = 30
'''<summary>Almacena la altura de la imagen, que varía según cuantos elementos tenga,
para la captura en imagen. Inicialmente alt.mín es la caja más la separación arriba y abajo</summary>
Private alturaEsquema As Integer = separacion * 2 + alturaCaja
''' <summary>Almacena la lista de Familias seleccionadas previamente si se la pasa el llamante </summary>
Private _listaPrevía As List(Of Guid)

' Ver: https://learning.oreilly.com/library/view/windows-forms-programming/0321125193/ sobre dibujar
' Ver: https://learning.oreilly.com/library/view/windows-forms-programming/0321125193/0321125193\_ch06lev1sec4.html
' y https://learning.oreilly.com/library/view/net-windows-forms/0596003382/ch07.html sobre el DoubleBuffered

''' <summary>
''' Constructor. Inicializa los componentes.También la listaPrevía.
''' Se configura el formulario y, sobre todo, el panel con DoubleBuffered, para evitar el parpadeo en el
''' redibujado
''' </summary>
Public Sub New()
' Esta llamada es exigida por el diseñador.
InitializeComponent()
'Nota: Necesito los doublebuffered para uqe no parpadeen los controles mientras los nuevo.
'Para este tipo de control (panel), la variable DoubleBuffered está protegida (protected)
'por lo que tengo que añadir el módulo ControlExtensions.vb, para forzar el cambio utilizando reflexion.
'Cambiado!, finalmente hago la reflexión aquí, con el método aquí, sin usar otro módulo
Me.DoubleBuffered = True
SetDoubleBuffered(PanelRepresentaciones, True)
SetDoubleBuffered(TreeViewFamDisponibles, True)
ListaPrevía = Nothing 'Inicializo a nada
End Sub

''' <summary>
''' Método que maneja la carga del formulario. Toma la instancia del servicioMaquinas, carga las Familias
''' de Máquinas disponibles para elección en el TreeView y, si hay una selección previa, llama al método
''' para cargarla en el panel y en la lista actual
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Public Sub FormSelecFam_Load(sender As Object, e As EventArgs) Handles MyBase.Load
sMasquinas = ServicioMaquinas.Instance 'La instancia (única) del servicioMoaquinas
CargarDatosEnTreeView() 'Cargo los datos
If ListaPrevía IsNot Nothing Then 'Si hay selección previa
CargaListaPrevía() 'Cargo la selección previa en el panel y en la lista actual
End If
End Sub

''' <summary>
''' Método que carga la selección previa de Familias en el panel y en la selección actual, para mantener
''' la consistencia con el estado actual de la aplicación
''' </summary>
Private Sub CargaListaPrevía()
For Each idFam In ListaPrevía 'Para cada identificador de la lista previa
If listaSeleccionFam.Count = limiteSeleccion Then 'Compruebo si se ha alcanzado el límite
MessageBox.Show("Se ha alcanzado el límite de tamaño de selección", "Error",
MessageBoxButtons.OK,
MessageBoxIcon.Error)
Else 'Si no se ha alcanzado el límite, obtengo la Familia a partir de su identificador
Dim familia As FamiliaMaquina = sMasquinas.GetFamiliaID(idFam)
listaSeleccionFam.Add(familia) 'Añado la Familia a la lista para gestionar el esquema
Dim nuevaCaja As New RepresentacionFamilia() With { 'Creo una nueva caja (representación familia)
.Text = familia.Nombre,
.Tag = familia,
.Width = longitudCaja,
.Height = alturaCaja}
nuevaCaja.ContextMenuStrip = ContextMenuStrip1 'Asigno el contextMenuStrip con el botón de
eliminar
AddHandler nuevaCaja.MouseHover, AddressOf Caja_MouseHover 'Añado los eventos de la caja
AddHandler nuevaCaja.MouseDown, AddressOf Caja_MouseDown
AddHandler nuevaCaja.MouseMove, AddressOf Caja_MouseMove
AddHandler nuevaCaja.MouseUp, AddressOf Caja_MouseUp
PanelRepresentaciones.Controls.Add(nuevaCaja) 'Añado la caja al panel y
listaRepresentaciones.Add(nuevaCaja) 'a la lista de representaciones

```

```

        ReordenarRepresentacion()                'Reordeno el esquema
    End If
Next
End Sub

''' <summary>
''' Método para forzar, mediante reflexión, el cambio de la propiedad protegida "DoubleBuffered" en los
''' controles Y así evitar el parpadeo
''' </summary>
''' <param name="control">El control del cual se quiere modificar el parámetro protegido</param>
''' <param name="valor">El valor a poner en el parámetro (boolean)</param>
Public Sub SetDoubleBuffered(control As Control, valor As Boolean)
    Dim tipo As Type = control.GetType() 'Tomo el Tipo del control que llama al método
    'Busco la información de la propiedad DoubleBuffered, marcada como instancia o "no publica" en el control
    Dim propiedad As PropertyInfo = tipo.GetProperty("DoubleBuffered", BindingFlags.Instance Or
        BindingFlags.NonPublic)
    'Cambio el valor de esa propiedad, en el control pasado, al valor pasado
    propiedad.SetValue(control, valor, Nothing)
End Sub

''' <summary>
''' Método que actualiza el TreeView para mostrar el listado de Familias de máquinas disponibles del catálogo
''' </summary>
Private Sub CargarDatosEnTreeView()
    'Creo y cargo la lista de familias disponibles para su selección, desde el catálogo
    Dim familiasDisponibles As List(Of FamiliaMaquina) = sMasquinas.GetFamilias()
    TreeViewFamDisponibles.Nodes.Clear() 'Limpio el TreeView antes de cargar los datos
    For Each familia As FamiliaMaquina In familiasDisponibles 'Recorro la lista de familias
        Dim nodoFamilia As New TreeNode(familia.Nombre) 'Creo un nodo con el nombre de la familia
        nodoFamilia.Tag = familia 'Asigno los Tag para identificar que es una familia
        TreeViewFamDisponibles.Nodes.Add(nodoFamilia) 'Añado el nodo al TreeView
    Next
    TreeViewFamDisponibles.Sort()                'Treeview ordenado alfabéticamente
End Sub

''' <summary>
''' Método para añadir la familia seleccionada en el TreeView al esquema del panel y a la selección
''' de Familias de máquinas
''' </summary>
''' <param name="sender">El botón de añadir Familia</param>
''' <param name="e">El click sobre el botón</param>
Private Sub AddFamilia_Click(sender As Object, e As EventArgs) Handles BttAddFamilia.Click
    Dim nodoSeleccionado As TreeNode = TreeViewFamDisponibles.SelectedNode 'Tomo el nodo seleccionado
    If nodoSeleccionado IsNot Nothing Then 'Compruebo que hay algo seleccionado
        Dim nombreSeleccionado As String = nodoSeleccionado.Text 'Tomo el Nombre del Modelo seleccionado
        If listaSeleccionFam.Count = limiteSeleccion Then 'Compruebo si se ha alcanzado el límite
            MessageBox.Show("Se ha alcanzado el límite de tamaño de selección", "Error",
                MessageBoxButtons.OK,
                MessageBoxIcon.Error)
        Else 'Si no se ha alcanzado el límite, tomo el ID del elemento seleccionado
            Dim idFam As Guid = sMasquinas.GetIDFamilia(nombreSeleccionado)
            Dim familia As FamiliaMaquina = sMasquinas.GetFamiliaID(idFam) 'Obtengo el modelo por su ID
            listaSeleccionFam.Add(familia) 'Añado el modelo a la lista para gestionar el esquema
            Dim nuevaCaja As New RepresentacionFamilia() With { 'Creo una nueva caja (representación de
familia)
                .Text = familia.Nombre,
                .Tag = familia,
                .Width = longitudCaja,
                .Height = alturaCaja}
            nuevaCaja.ContextMenuStrip = ContextMenuStrip1 'Asigno el contextMenuStrip con el botón de
eliminar
            AddHandler nuevaCaja.MouseHover, AddressOf Caja_MouseHover 'Añado los eventos de la caja
            AddHandler nuevaCaja.MouseDown, AddressOf Caja_MouseDown
            AddHandler nuevaCaja.MouseMove, AddressOf Caja_MouseMove
            AddHandler nuevaCaja.MouseUp, AddressOf Caja_MouseUp
            PanelRepresentaciones.Controls.Add(nuevaCaja) 'Añado la caja al panel y
            listaRepresentaciones.Add(nuevaCaja) 'a la lista de representaciones
            ReordenarRepresentacion() 'Reordeno los botones
        End If
    Else 'Si no se ha seleccionado ningún elemento, mostrar un mensaje de error
        MessageBox.Show("No se ha seleccionado ningún elemento.", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error)
    End If
End Sub

''' <summary>
''' Método que recoloca la posición de las cajas según su orden en la listaRepresentaciones, colocándolas
''' en la siguiente línea si se pasan de ancho del panel donde se dibujan
''' </summary>
Private Sub ReordenarRepresentacion()

```

```

Dim x As Integer = separacion 'El primer punto, x,y será a una dist "separación" del inicio del panel
Dim y As Integer = separacion
alturaEsquema = separacion * 2 + alturaCaja 'Reinicio la altura que tomará el dibujo cuando se capture
For Each caja As RepresentacionFamilia In listaRepresentaciones 'Por cada caja en la lista
    caja.Location = New Point(x, y) 'Pongo la caja en la posición marcada por x e y
    x += caja.Width + separacion 'Por cada caja, aumento x en largo caja + la separación, para la sig.caja
    'Si al sumarle el largo de la sig.caja a la posición actual de x se sale del panel
    If x + caja.Width > PanelRepresentaciones.Width Then
        x = separacion 'Pongo x otra vez al inicio
        y += caja.Height + separacion 'Aumento altura (recuerda, hacia abajo) con una fila entera:
caja+sep.
        'Aumento en una fila la altura que tomará la captura del dibujo
        alturaEsquema += separacion + alturaCaja
    End If
Next
PanelRepresentaciones.Invalidate() 'Invalido para redibujar el panel
End Sub

''' <summary>
''' Método que captura el evento de pasar el ratón por encima de una de las representaciones, para mostrar
''' los datos de la Familia de máquinas que representa
''' </summary>
''' <param name="sender">La representación (caja)</param>
''' <param name="e">Posicionarse encima</param>
Private Sub Caja_MouseHover(sender As Object, e As EventArgs)
    Dim caja As RepresentacionFamilia = CType(sender, RepresentacionFamilia) 'Tomo el tipo de la caja
    Dim familia As FamiliaMaquina = CType(caja.Tag, FamiliaMaquina) 'Tomo el Modelo
    'Muestro mediante el Tooltip los datos del modelo de clase
    ToolTip1.SetToolTip(caja, "Nombre: " & familia.Nombre & vbCrLf & "Descripción: " & familia.Descripcion)
End Sub

''' <summary>
''' Método para eliminar una de las familias Seleccionadas de la lista que se está seleccionando, así como
''' de su lista de representaciones y, por ello, del panel donde se muestran las representaciones
''' </summary>
''' <param name="caja">Representación de Familia que se pretende eliminar</param>
Private Sub EliminarCaja(caja As RepresentacionFamilia)
    'Tomo la familia desde el Tag de la Representación
    Dim familia As FamiliaMaquina = CType(caja.Tag, FamiliaMaquina)
    'No me es necesario comprobar si la lista tiene el objeto porque no llegaríamos aquí si así no fuese así
    listaSeleccionFam.Remove(familia) 'Elimino esa familia de la lista de familias seleccionados hasta el
momento
    PanelRepresentaciones.Controls.Remove(caja) 'Elimino la representación de las incluidas en el Panel
    listaRepresentaciones.Remove(caja) 'Elimino la representación de la lista de representaciones
seleccionadas
    ReordenarRepresentacion() 'Vuelvo a reordenar las restantes representaciones
End Sub

''' <summary>
''' Método que captura cada redibujado del panel para crear las representaciones y las flechas que las
conectan
''' </summary>
''' <param name="sender">El Panel donde se muestran las representaciones de Modelos</param>
''' <param name="e">El evento de redibujado del panel</param>
Private Sub PanelRepresentaciones_Paint(sender As Object, e As PaintEventArgs) Handles
PanelRepresentaciones.Paint
    Dim g As Graphics = e.Graphics
    'Reorro la lista de representaciones hasta la penúltima
    For i As Integer = 0 To listaRepresentaciones.Count - 2
        Dim cajaActual As RepresentacionFamilia = listaRepresentaciones(i) 'Tomo la caja de este ciclo
        Dim cajaSiguiente As RepresentacionFamilia = listaRepresentaciones(i + 1) 'Caja a la que irá la
flecha
        If EsUltimoDeLaFila(cajaActual) Then 'Si la caja actual es la última de la fila
            ' Dibujo la flecha hacia abajo, luego a la izquierda hasta la primera de la sig.fila,
            ' y luego hacia abajo de nuevo
            'Parto de la mitad inf. de la caja
            Dim puntoInicio As New Point(cajaActual.Left + cajaActual.Width / 2, cajaActual.Bottom)
            'Bajo (la altura crece hacia abajo) a la mitad de la separacion
            Dim puntoIntermedio1 As New Point(puntoInicio.X, puntoInicio.Y + separacion / 2)
            'Voy a la izq. a mitad de la 1ª caja de la fila
            Dim puntoIntermedio2 As New Point(cajaSiguiente.Left + cajaSiguiente.Width / 2,
puntoIntermedio1.Y)
            'Finalmente bajo hasta la parte sup. de la 1ª caja de la fila
            Dim puntoFin As New Point(puntoIntermedio2.X, cajaSiguiente.Top)
            DibujarFlecha(g, puntoInicio, puntoIntermedio1) 'Dibujo las flechas en cada tramo
            DibujarFlecha(g, puntoIntermedio1, puntoIntermedio2)
            DibujarFlecha(g, puntoIntermedio2, puntoFin)
        Else 'Si no es la última de la fila entonces la siguiente está en línea
            'Salgo a mitad de altura del extremo dcho de esta caja

```

```

        Dim puntoInicio As New Point(cajaActual.Right, cajaActual.Top + cajaActual.Height / 2)
        'Acabo a mitad de alt. del ext.izq. de la siguiente
        Dim puntoFin As New Point(cajaSiguiente.Left, cajaSiguiente.Top + cajaSiguiente.Height / 2)
        DibujarFlecha(g, puntoInicio, puntoFin) 'Dibujo la flecha del único tramo
    End If
Next
End Sub

''' <summary>
''' Método que comprueba si una caja es la última de la fila en la representación del panel
''' Si la posición de su extremo derecho más el ancho de la que sería la siguiente caja más la separación
''' se pasan del extremo derecho del panel, entonces sí, es la última de la fila
''' </summary>
''' <param name="caja">La caja de la cual se quiere comprobar si es la última de la fila</param>
''' <returns>true si es la última, false si no</returns>
Private Function EsUltimoDeLaFila(caja As RepresentacionFamilia) As Boolean
    Return caja.Right + caja.Width + separacion > PanelRepresentaciones.Width
End Function

''' <summary>
''' Método que dibuja las flechas entre representaciones.
''' Dibuja una línea personalizada con una punta de flecha en su extremo, entre dos puntos dados
''' y en el componente gráfico pasado
''' </summary>
''' <param name="g">El componente gráfico donde se dibujará la flecha</param>
''' <param name="puntoInicio">El punto inicial de donde parte la flecha</param>
''' <param name="puntoFin">El punto final donde acaba la flecha</param>
Private Sub DibujarFlecha(g As Graphics, puntoInicio As Point, puntoFin As Point)
    Dim lapiz As New Pen(Color.Black, 2) 'Creo un nuevo pen en color Negro y grosor 2 (¿parametrizar?)
    Dim punta As New AdjustableArrowCap(3, 3) 'Defino la punta de la línea como una punta de flecha de 3x3
    lapiz.CustomEndCap = punta
    g.DrawLine(lapiz, puntoInicio, puntoFin) 'Asigno la punta de la línea a la línea
    'Dibujo la línea en el componente
End Sub

''' <summary>
''' Método que captura el evento de pulsar un botón del ratón sobre una de las representaciones
''' Si es botón izquierdo, activa booleano que indica que se está arrastrando dicha representación,
''' y guarda la representación seleccionada así como su posición original.
''' Si es botón derecho, muestra el menú con el botón de eliminar ese modelo de los seleccionados
''' </summary>
''' <param name="sender">Los botones del ratón</param>
''' <param name="e">la pulsación</param>
Private Sub Caja_MouseDown(sender As Object, e As MouseEventArgs)
    If e.Button = MouseButtons.Left Then 'Si es botón izq.
        arrastrandoCaja = True 'Marca de que estoy arrastrando una representación
        cajaSeleccionada = CType(sender, RepresentacionFamilia) 'Tomo la representación a arrastrar
        posicionOriginalCaja = e.Location 'Tomo la posición original desde donde voy a arrastrar
    ElseIf e.Button = MouseButtons.Right Then 'Si es botón dcho
        ContextMenuStrip1.Tag = sender 'Asigno el tag del botón al contextmenu
        ContextMenuStrip1.Show(Cursor.Position) 'Muestro el menú bajo el ratón
    End If
End Sub

''' <summary>
''' Método que captura la pulsación sobre el elemento eliminar del menú contextual de las representaciones,
''' para llamar al método que eliminará dicha representación y su familia asociada en los listados de
''' seleccionadas
''' </summary>
''' <param name="sender">El elemento del menú contextual de la representación</param>
''' <param name="e">El click sobre el elemento</param>
Private Sub Eliminar_Click(sender As Object, e As EventArgs) Handles TSMItemEliminar.Click
    'Tomo la caja del tag del contextmenu
    Dim caja As RepresentacionFamilia = CType(ContextMenuStrip1.Tag, RepresentacionFamilia)
    If caja IsNot Nothing Then 'Si se ha seleccionado una caja
        EliminarCaja(caja) 'Elimino la caja del panel y el Modelo que representa del listado
    End If
End Sub

''' <summary>
''' Método para gestionar el movimiento de una representación y las nuevas posiciones de ella y las demás,
''' así como de las familias de máquinas asociadas
''' </summary>
''' <param name="sender">La representación que se está moviendo</param>
''' <param name="e">El movimiento del ratón</param>
Private Sub Caja_MouseMove(sender As Object, e As MouseEventArgs)
    'Si se ha marcado arrastrandoCaja como true (se ha pulsado botón izq sobre una caja) y ésta está
    seleccionada
    If arrastrandoCaja AndAlso cajaSeleccionada IsNot Nothing Then
        Dim nuevaPosicion As Point = cajaSeleccionada.Location 'Tomo la posición actual de la caja
        'Le aplico la dif. = pos. del ratón actual menos la pos. inicial

```

```

nuevaPosicion.Offset(e.Location - posicionOriginalCaja)
cajaSeleccionada.Location = nuevaPosicion 'Asigno a la representación seleccionada, la nueva posición
Dim fam As FamiliaMaquina = CType(cajaSeleccionada.Tag, FamiliaMaquina) 'Tomo la familia seleccionada
'Reordeno las representaciones en "tiempo real". Cambio las posiciones según la nueva posición
'de la caja actual
For Each caja As RepresentacionFamilia In listaRepresentaciones 'Para cada caja en la lista
    'Si se trata de una de las cajas que no es la que tengo pulsada y se están pisando
    If caja IsNot cajaSeleccionada AndAlso caja.Bounds.Intersects(cajaSeleccionada.Bounds) Then
        'Tomo la posición en la lista de la caja pisada
        Dim indiceCaja As Integer = listaRepresentaciones.IndexOf(caja)
        listaRepresentaciones.Remove(cajaSeleccionada) 'Quito la caja que estoy moviendo de la lista
        'Y la vuelvo a introducir en la lista, en la posición de la pisada
        listaRepresentaciones.Insert(indiceCaja, cajaSeleccionada)
        'La pisada, y siguientes, se recolocarán automáticamente a continuación,
        'se moverán todas 1 a la derecha
        'Hago lo mismo con la lista de Modelos seleccionados
        Dim familia As FamiliaMaquina = CType(caja.Tag, FamiliaMaquina) 'Tomo la familia que
representa
        Dim indiceFamilia As Integer = listaSeleccionFam.IndexOf(familia) 'Y hago lo mismo que antes
        listaSeleccionFam.Remove(fam)
        listaSeleccionFam.Insert(indiceFamilia, fam)
        ReordenarRepresentacion() 'Vuelvo a colocar las cajas en el panel
    Exit For
    End If
Next
PanelRepresentaciones.Invalidate() ' Redibujar el panel para actualizar las flechas
End If
End Sub

''' <summary>
''' Método que captura el evento de dejar de pulsar el botón izq del ratón
''' </summary>
''' <param name="sender">La caja que estaba pulsada</param>
''' <param name="e">Soltar el botón</param>
Private Sub Caja_MouseUp(sender As Object, e As MouseEventArgs)
    If e.Button = MouseButton.Left Then 'Si el botón que se suelta es el izq
        arrastrandoCaja = False 'Ya no se está arrastrando ninguna caja
        cajaSeleccionada = Nothing 'Y no hay ninguna seleccionada
        ReordenarRepresentacion() 'Vuelvo a colocar las cajas en el panel
        '(Quizás no es necesario, pero por si acaso)
    End If
End Sub

''' <summary>
''' Método que captura la aceptación del usuario.
''' Captura la imagen del esquema, prepara la lista de id's de familias de máquinas seleccionadas
''' y devuelve el resultado de OK para informar al método llamante
''' </summary>
''' <param name="sender">El botón de aceptar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub Aceptar_Click(sender As Object, e As EventArgs) Handles BttAceptar.Click
    'Inicializo la imagen del esquema con el tamaño del panel
    _esquema = New Bitmap(PanelRepresentaciones.Width, alturaEsquema)
    'Dibujo el contenido del panel en un rectángulo, en el esquema
    PanelRepresentaciones.DrawToBitmap(_esquema, New Rectangle(0, 0, PanelRepresentaciones.Width,
alturaEsquema))
    'Genero la lista de id's a partir de la lista de Familias de máquinas seleccionadas
    _seleccionModFam.Clear()
    For Each familia In listaSeleccionFam
        _seleccionModFam.Add(familia.ID)
    Next
    Me.DialogResult = DialogResult.OK 'Marco el resultado del diálogo
    Me.Close() 'Cierro este formulario
End Sub

''' <summary>
''' Método que captura la cancelación del formulario pro parte del usuario.
''' Se pierden los cambios y se marca el resultado del diálogo = Cancel
''' </summary>
''' <param name="sender">El botón de cancelar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub Cancelar_Click(sender As Object, e As EventArgs) Handles BttCancelar.Click
    Me.DialogResult = DialogResult.Cancel 'Marco el resultado del diálogo y cierro este formulario
    Me.Close()
End Sub

''' <summary>La lista para almacenar los ID's de las familias seleccionados para dimensionar</summary>
Public Property SeleccionModClase As List(Of Guid)
Get
    Return _seleccionModFam

```

```

        End Get
        Set
            _seleccionModFam = Value
        End Set
    End Property

    ''' <summary> La imagen esquema de la linea según sus familias de máquinas seleccionadas</summary>
    Public Property Esquema As Image
        Get
            Return _esquema
        End Get
        Set
            _esquema = Value
        End Set
    End Property

    ''' <summary>Almacena la lista de Familias seleccionadas previamente si se la pasa el llamante </summary>
    Public Property ListaPrevía As List(Of Guid)
        Get
            Return _listaPrevía
        End Get
        Set
            _listaPrevía = Value
        End Set
    End Property
End Class

```

B.14 Clase RepresentacionFamilia

```

''' <summary>
''' Clase para definir un control personalizado que representará a cada Familia
''' en el esquema de selección de Familias de máquinas de la línea a dimensionar
''' Serán "cajas" rectangulares, con las esquinas redondeadas
''' </summary>
''' NOTA: Ver:
''' https://learning.oreilly.com/library/view/windows-forms-programming/0321125193/0321125193\_ch08lev1sec2.html
Public Class RepresentacionFamilia

    Inherits Control 'Hereda de Control para los métodos que usaré en el dibujo del esquema

    ''' <summary>Color del fondo del control</summary>
    Private _colorFondo As Color = Color.LightGray
    ''' <summary>Almaceno el radio del rectángulo a dibujar </summary>
    Private _radioBorde As Integer = 4
    ''' <summary>Color del texto</summary>
    Private _colorTexto As Color = Color.Black
    ''' <summary>Color del rectángulo redondeado</summary>
    Private _colorBorde As Color = Color.Black
    ''' <summary>Color del fondo cuando el ratón pasa por encima</summary>
    Private _colorRatonEncima As Color = Color.DarkGray
    ''' <summary>Color del fondo original antes de pasar por encima</summary>
    Private ColorFondoOriginal As Color

    ''' <summary>
    ''' Constructor para asignar los eventos de cada caja
    ''' </summary>
    Public Sub New()
        InitializeComponent()
        ' Asignar el manejador de eventos de mouse
        AddHandler Me.MouseEnter, AddressOf OnMouseEnter
        AddHandler Me.MouseLeave, AddressOf OnMouseLeave
    End Sub

    ''' <summary>
    ''' Método que dibuja el control
    ''' </summary>
    ''' <param name="e"></param>
    Protected Overrides Sub OnPaint(ByVal e As PaintEventArgs)
        MyBase.OnPaint(e)
        Dim g As Graphics = e.Graphics
        Dim rect As New Rectangle(0, 0, Me.Width - 1, Me.Height - 1) 'Rectángulo que delimita el control
        'Creo el pincel del fondo con el color definido en las prop. del control
        Dim backgroundBrush As New SolidBrush(ColorFondo)
        g.FillRectangle(backgroundBrush, rect) 'Dibujo el fondo del control
        Dim borderPen As New Pen(BordeColor, 1) 'Defino el pincel
        'Defino el camino y añado los puntos con los arcos que delimitan la forma
    End Sub

```

```

Dim borderPath As New Drawing2D.GraphicsPath()
borderPath.AddArc(New Rectangle(rect.X, rect.Y, RadioBorde * 2, RadioBorde * 2), 180, 90)
borderPath.AddArc(New Rectangle(rect.Right - RadioBorde * 2, rect.Y, RadioBorde * 2, RadioBorde * 2),
270, 90)
borderPath.AddArc(New Rectangle(rect.Right - RadioBorde * 2, rect.Bottom - RadioBorde * 2, RadioBorde *
2, RadioBorde * 2), 0, 90)
borderPath.AddArc(New Rectangle(rect.X, rect.Bottom - RadioBorde * 2, RadioBorde * 2, RadioBorde * 2),
90, 90)
borderPath.CloseFigure()
g.DrawPath(borderPen, borderPath) 'Pinto con ese pincel, el camino definido
'Pongo el nombre en el centro del control
'Creo el Pincel del texto, con el color definido en las prop.del Control
Dim textBrush As New SolidBrush(TextColor)
Dim formato As New StringFormat() 'Defino donde escribiré el nombre del control
formato.Alignment = StringAlignment.Center 'Centrado en las 2 dimensiones
formato.LineAlignment = StringAlignment.Center
Dim textoAEscribir As String = Text ' .Text heredado
If Text.Length > 22 Then 'Si el texto es demasiado largo,
    textoAEscribir = Text.Substring(0, 22) & "..." 'Añado puntos para mostrar que no es texto completo
End If
g.DrawString(textoAEscribir, Me.Font, textBrush, rect, formato) 'Dibujo el texto
End Sub

''' <summary>
''' Método para capturar el evento de cuando el ratón entra encima del control
''' Cambia el color del fondo y redibuja
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub OnMouseEnter(sender As Object, e As EventArgs)
    ColorFondoOriginal = _colorFondo
    ColorFondo = _colorRatonEncima
    Invalidate() ' Forzar el redibujado
End Sub

''' <summary>
''' Método para capturar el evento de cuando el ratón sale del control
''' Vuelve al color del fondo original y redibuja
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub OnMouseLeave(sender As Object, e As EventArgs)
    ColorFondo = ColorFondoOriginal
    Invalidate() ' Forzar el redibujado
End Sub

Public Property ColorRatonEncima As Color
    Get
        Return _colorRatonEncima
    End Get
    Set
        _colorRatonEncima = Value
    End Set
End Property

Public Property ColorFondo As Color
    Get
        Return _colorFondo
    End Get
    Set
        _colorFondo = Value
    End Set
End Property

Public Property RadioBorde As Integer
    Get
        Return _radioBorde
    End Get
    Set
        _radioBorde = Value
    End Set
End Property

Public Property TextColor As Color
    Get
        Return _colorTexto
    End Get
    Set
        _colorTexto = Value
    End Set

```

```

End Property

Public Property BordeColor As Color
    Get
        Return _colorBorde
    End Get
    Set
        _colorBorde = Value
    End Set
End Property

End Class

```

B.15 Clase FormDatosLinea

```

''' <summary>
''' Formulario para introducir los datos informativos de la línea a dimensionar.
''' Estos datos se mostrarán al principio del informe sobre la línea dimensionada
''' </summary>
Public Class FormDatosLinea
    ''' <summary>Texto que aparecerá en el campo NOMBRE del informe de la línea </summary>
    Private _nombre As String
    ''' <summary>Texto que aparecerá en el campo CLIENTE del informe de la línea </summary>
    Private _cliente As String
    ''' <summary>Texto que aparecerá en el campo PROYECTO del informe de la línea </summary>
    Private _proyecto As String
    ''' <summary>Texto que aparecerá en el campo VERSION del informe de la línea </summary>
    Private _version As String
    ''' <summary>Texto que aparecerá en el campo DESCRIPCION del informe de la línea </summary>
    Private _descripcion As String

    ''' <summary>
    ''' Constructor
    ''' Toma los datos que le pasan para colocarlos de base y permitir editarlos.
    ''' </summary>
    Public Sub New(Nombre As String, Cliente As String, Proyecto As String, Version As String, Descripcion As String)
        InitializeComponent()
        TextBoxNombre.Text = Nombre 'Asigno cada dato a su textbox correspondiente
        TextBoxCliente.Text = Cliente
        TextBoxProyecto.Text = Proyecto
        TextBoxVersion.Text = Version
        TextBoxDescripcion.Text = Descripcion
    End Sub

    ''' <summary>
    ''' Método que captura la acción de aceptar y crea y rellena el diccionario con los textos de
    ''' los textbox para dejarlos disponibles para el formulario que lo ha llamado.
    ''' Marco el resultado del diálogo = OK y cierro este formulario
    ''' </summary>
    ''' <param name="sender">Botón aceptar</param>
    ''' <param name="e">Pulsar el botón</param>
    Private Sub Aceptar_Click(sender As Object, e As EventArgs) Handles BttAceptar.Click
        Nombre = TextBoxNombre.Text 'Recojo los datos de los textbox
        Cliente = TextBoxCliente.Text
        Proyecto = TextBoxProyecto.Text
        Version = TextBoxVersion.Text
        Descripcion = TextBoxDescripcion.Text
        Me.DialogResult = DialogResult.OK 'Marco el resultado del diálogo y cierro este formulario
        Me.Close()
    End Sub

    ''' <summary>
    ''' Método que captura el evento de pulsar sobre el botón cancelar
    ''' Se pierden los posibles datos introducidos y se cierra el formulario
    ''' Marca el resultado del diálogo = Cancel
    ''' </summary>
    ''' <param name="sender">El botón cancelar</param>
    ''' <param name="e">El click sobre el botón</param>
    Private Sub Cancelar_Click(sender As Object, e As EventArgs) Handles BttCancelar.Click
        Me.DialogResult = DialogResult.Cancel 'Marco el resultado del diálogo y cierro este formulario
        Me.Close()
    End Sub

```

```

''' <summary>Texto que aparecerá en el campo NOMBRE del informe de la línea </summary>
Public Property Nombre As String
    Get
        Return _nombre
    End Get
    Set
        _nombre = Value
    End Set
End Property

''' <summary>Texto que aparecerá en el campo CLIENTE del informe de la línea </summary>
Public Property Cliente As String
    Get
        Return _cliente
    End Get
    Set
        _cliente = Value
    End Set
End Property

''' <summary>Texto que aparecerá en el campo PROYECTO del informe de la línea </summary>
Public Property Proyecto As String
    Get
        Return _proyecto
    End Get
    Set
        _proyecto = Value
    End Set
End Property

''' <summary>Texto que aparecerá en el campo VERSION del informe de la línea </summary>
Public Property Version As String
    Get
        Return _version
    End Get
    Set
        _version = Value
    End Set
End Property

''' <summary>Texto que aparecerá en el campo DESCRIPCION del informe de la línea </summary>
Public Property Descripcion As String
    Get
        Return _descripcion
    End Get
    Set
        _descripcion = Value
    End Set
End Property
End Class

```

B.16 Clase FormCondiciones

```

''' <summary>
''' Formulario para la selección de las condiciones de dimensionamiento de la línea.
''' Contendrá la información en cuanto a la producción mínima requerida, el paralelismo
''' de máquinas (cantidad máxima de máquinas trabajando en paralelo para
''' cumplir la producción mínima), y las características a minimizar en la selección.
''' </summary>
Public Class FormCondiciones
    ''' <summary>Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la producción</summary>
    Private _paralelismo As Integer
    ''' <summary>Producción mínima requerida </summary>
    Private _produccionMin As Single
    ''' <summary>Propiedad que se debe minimizar en la selección (primaria)</summary>
    Private _minimizarPrimaria As String
    ''' <summary>Propiedad que se debe minimizar en la selección (secundaria)</summary>
    Private _minimizarSecundaria As String

    ''' <summary>
    ''' Constructor con parámetros. Toma los valores de condiciones para inicializar sus campos
    ''' </summary>
    ''' <param name="paralelismo">Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la
    ''' producción</param>
    ''' <param name="produccionMin">Producción mínima requerida</param>
    ''' <param name="minimizarPrimaria">Propiedad que se debe minimizar en la selección (primaria)</param>

```

```

''' <param name="minimizarSecundaria">Propiedad que se debe minimizar en la selección (secundaria)</param>
Public Sub New(paralelismo As Integer, produccionMin As Single, minimizarPrimaria As String,
               minimizarSecundaria As String)
    ' Esta llamada es exigida por el diseñador.
    InitializeComponent()
    'Copio los valores pasados para mantener la consistencia con el estado actual
    Me.paralelismo = paralelismo
    Me.produccionMin = produccionMin
    Me.minimizarPrimaria = minimizarPrimaria
    Me.minimizarSecundaria = minimizarSecundaria
End Sub

''' <summary>
''' Método que maneja el evento de carga del formulario.
''' Toma los valores de sus variables y los muestra en los controles
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub FormCondiciones_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'Cargo el estado en los controles
    TextBoxProduccion.Text = produccionMin
    TextBoxParalelismo.Text = paralelismo
    Select Case _minimizarPrimaria
        Case Nothing
            RB1N.Checked = True
        Case "Operarios"
            RB1O.Checked = True
        Case "PotenciaElectrica"
            RB1P.Checked = True
        Case "ValorCoste"
            RB1C.Checked = True
        Case "ValorVenta"
            RB1V.Checked = True
    End Select
    Select Case _minimizarSecundaria
        Case Nothing
            RB2N.Checked = True
        Case "Operarios"
            RB2O.Checked = True
        Case "PotenciaElectrica"
            RB2P.Checked = True
        Case "ValorCoste"
            RB2C.Checked = True
        Case "ValorVenta"
            RB2V.Checked = True
    End Select
End Sub

''' <summary>
''' Método que controla que sólo se pueda seleccionar una condición secundaria si se ha seleccionado
''' una condición primaria.
''' Si la condición primaria está en Nothing, fuerza a que la secundaria también lo esté
''' </summary>
''' <param name="sender">El radioButton RB1N que representa Ninguno para la condición primaria</param>
''' <param name="e">El cambio del estado del radioButton</param>
Private Sub RadioButton_CheckedChanged(sender As Object, e As EventArgs) Handles RB1N.CheckedChanged
    If RB1N.Checked Then 'Si no se selecciona una condición primaria
        RB2N.Checked = True 'Tampoco se seleccionará una secundaria
        GroupBoxObj2.Enabled = False 'Y se deshabilitan todos los RadioButtons de la condición secundaria
    ElseIf Not RB1N.Checked Then 'Si se ha seleccionado cualquier condición primaria
        GroupBoxObj2.Enabled = True 'Se habilita la posibilidad de seleccionar una secundaria
    End If
End Sub

''' <summary>
''' Método para controlar que en textBox sólo se introduzcan números
''' </summary>
''' <param name="sender">textBoxProduccion y textBoxParalelismo</param>
''' <param name="e">Teclas pulsadas del teclado en el campo de texto de esos textbox</param>
Private Sub TextBox_KeyPress(sender As Object, e As KeyEventArgs) Handles TextBoxProduccion.KeyPress,
    TextBoxParalelismo.KeyPress
    If Char.IsDigit(e.KeyChar) Then 'Si la pulsación es un dígito
        e.Handled = False
    ElseIf Char.IsControl(e.KeyChar) Then 'Si la pulsación es una tecla de control
        e.Handled = False
    Else 'Resto de casos
        e.Handled = True
    End If
End Sub

```

```

''' <summary>
''' Método que maneja elevento de pulsar el botón de aceptar. Comprueba los datos introducidos, los
''' asigna a sus variables para que puedan ser recuperados,y marca el resultado del diálogo
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub Aceptar_Click(sender As Object, e As EventArgs) Handles BttAceptar.Click
'Tomo el valor de producción deseado y lo intento parsear a Single.
'Si no funciona, es que el usuario no ha usado sólo números
'Aunque ya nod ebería pasar pues su texto está controlado por el método "textBox_KeyPress"
Dim produccion As Single
If Single.TryParse(TextBoxProduccion.Text, produccion) Then
    produccionMin = produccion 'Lo asigno al atributo correspondiente
Else
    MessageBox.Show("El valor de producción debe ser numérico.")
    TextBoxProduccion.Clear()
    Return
End If
'Segunda propiedad a minimizar. Según sea el RadioButton pulsado, asigno el tipo de valor a minimizar
'La selección de sólo uno de ellos ya está gestionada por estar en un GroupBox
Select Case True
    Case RB1N.Checked
        minimizarPrimaria = Nothing
    Case RB1O.Checked
        minimizarPrimaria = "Operarios"
    Case RB1P.Checked
        minimizarPrimaria = "PotenciaElectrica"
    Case RB1C.Checked
        minimizarPrimaria = "ValorCoste"
    Case RB1V.Checked
        minimizarPrimaria = "ValorVenta"
End Select
'Segunda propiedad
Select Case True
    Case RB2N.Checked
        minimizarSecundaria = Nothing
    Case RB2O.Checked
        minimizarSecundaria = "Operarios"
    Case RB2P.Checked
        minimizarSecundaria = "PotenciaElectrica"
    Case RB2C.Checked
        minimizarSecundaria = "ValorCoste"
    Case RB2V.Checked
        minimizarSecundaria = "ValorVenta"
End Select
'Tomo el valor de paralelismo deseado y lo intento parsear a Single.
'Si no funciona, es que el usuario no ha usado sólo números
'Aunque ya no debería pasar pues su texto está controlado por el método "textBox_KeyPress"
Dim valorParalelismo As Single
If Single.TryParse(TextBoxParalelismo.Text, valorParalelismo) Then
    paralelismo = valorParalelismo 'Lo asigno al atributo paralelismo
Else
    MessageBox.Show("El valor de paralelismo debe ser numérico.")
    TextBoxParalelismo.Clear()
    Return
End If
Me.DialogResult = DialogResult.OK 'Marco el resultado del diálogo
Me.Close() 'Cierro este formulario
End Sub

''' <summary>
''' Método que captura la cancelación del formulario pro parte del usuario
''' </summary>
''' <param name="sender">El botón de cancelar</param>
''' <param name="e">El click sobre el botón</param>
Private Sub Cancelar_Click(sender As Object, e As EventArgs) Handles BttCancelar.Click
    Me.DialogResult = DialogResult.Cancel 'Marco el resultado del diálogo y cierro este formulario
    Me.Close()
End Sub

''' <summary>Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la producción</summary>
Public Property paralelismo As Integer
    Get
        Return _paralelismo
    End Get
    Set
        _paralelismo = Value
    End Set
End Property

```

```

''' <summary>Producción mínima requerida </summary>
Public Property produccionMin As Single
    Get
        Return _produccionMin
    End Get
    Set
        _produccionMin = Value
    End Set
End Property

''' <summary>Propiedad que se debe minimizar en la selección (primaria)</summary>
Public Property minimizarPrimaria As String
    Get
        Return _minimizarPrimaria
    End Get
    Set
        _minimizarPrimaria = Value
    End Set
End Property

''' <summary>Propiedad que se debe minimizar en la selección (secundaria)</summary>
Public Property minimizarSecundaria As String
    Get
        Return _minimizarSecundaria
    End Get
    Set
        _minimizarSecundaria = Value
    End Set
End Property

End Class

```

B.17 Clase FormAcercaDe

```

''' <summary>
''' Formulario informativo que muestra datos sobre la aplicación
''' Muestra: Nombre, Versión, Compañía y una descripción
''' </summary>
''' <remarks>Código creado por Visual Studio y posteriormente modificado por mi </remarks>

Public NotInheritable Class FormAcercaDe
    ''' <summary>
    ''' Método que captura el evento de carga del formulario. Carga los textos para mostrar la información
    ''' </summary>
    ''' <param name="sender">El propio formulario</param>
    ''' <param name="e">Load, la carga del formulario</param>
    Private Sub FormAcercaDe_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Dim ApplicationTitle As String
        If My.Application.Info.Title <> "" Then
            ApplicationTitle = My.Application.Info.Title
        Else
            ApplicationTitle = System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If
        Me.Text = String.Format("Acerca de {0}", ApplicationTitle)
        ' Inicialice todo el texto mostrado en el cuadro Acerca de.
        ' TODO: personalice la información del ensamblado de la aplicación en el panel "Aplicación" del
        ' cuadro de diálogo propiedades del proyecto (bajo el menú "Proyecto").
        Me.LabelProductName.Text = My.Application.Info.ProductName
        Me.LabelVersion.Text = String.Format("Versión {0}", My.Application.Info.Version.ToString)
        Me.LabelCopyright.Text = My.Application.Info.Copyright
        Me.LabelCompanyName.Text = My.Application.Info.CompanyName
        Me.TextBoxDescription.Text = My.Application.Info.Description
    End Sub

    ''' <summary>
    ''' Método que captura el evento de pulsar sobre el botón aceptar, para cerrar el formulario
    ''' </summary>
    ''' <param name="sender">El botón aceptar</param>
    ''' <param name="e">El clickado sobre él</param>
    Private Sub Aceptar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OKButton.Click
        Me.Close()
    End Sub

End Class

```

ANEXO C DOCUMENTACIÓN DE CÓDIGO

Se muestra a continuación la documentación completa del código de la aplicación, generada con la herramienta [SandCastle Help File Builder](#). Se omiten explicaciones de métodos heredados que no aportan información sobre el trabajo realizado por el autor.

DraftLine Espacio de nombres

Clases

Condiciones	Clase para encapsular las condiciones de dimensionamiento, elegidas por el usuario, para una línea en particular. Formará una característica de esa línea y será utilizada para el cálculo de dimensionamiento. Cuenta con las variables para guardar: La producción mínima requerida, el paralelismo (cantidad máxima de máquinas en paralelo para conseguir la producción mínima) y las características a minimizar en la selección.
Dimensionar	Esta clase encapsula todo lo relacionado con los cálculos de selección según las condiciones del usuario. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame. Tomará cada una de las Familias de máquinas, seleccionadas por el usuario para la línea, y seleccionará la máquina, de cada Familia, que mejor se adapte a las condiciones indicadas por el usuario. El resultado será una lista de Máquinas que conformarán la línea dimensionada.
FamiliaMaquina	Clase que representa las Familias de máquinas. Posee los atributos que reflejan las características de cada Familia de Máquinas, así como una lista para albergar las Máquinas que pertenezcan a esta Familia.
FormAcercaDe	Formulario informativo que muestra datos sobre la aplicación. Muestra: Nombre, Versión, Compañía y una descripción.
FormCondiciones	Formulario para la selección de las condiciones de dimensionamiento de la línea. Contendrá la información en cuanto a la producción mínima requerida, el paralelismo de máquinas (cantidad máxima de máquinas trabajando en paralelo para cumplir la producción mínima), y las características a minimizar en la selección.
FormDatosLinea	Formulario para introducir los datos informativos de la línea a dimensionar. Estos datos se mostrarán al principio del informe sobre la línea dimensionada.
FormFamilia	Formulario para introducir los datos de la nueva Familia de máquinas a crear o editar una existente.

FormMaquina	Formulario para introducir los datos de la nueva Máquina a crear o editar una existente
FormPrincipal	Esta clase es el formulario base de la aplicación. Se carga al iniciar la aplicación y muestra los modos Editar catálogo, Dimensionar Línea y ayuda, desde donde trabajará el usuario. Contiene los controles (botones, menús, etc.) para acceder a todas las funcionalidades de la aplicación. Desde ella se crean los demás formularios que se le presentarán al usuario. Es este formulario el que tomará los datos de esos formularios para luego utilizar el resto de clases y cumplir los casos de uso definidos.
FormSelecFam	Clase con el formulario para que el usuario seleccione las Familias de máquinas que compondrán la Línea, así como su orden. Esa selección se realiza añadiendo Familias de máquinas representadas gráficamente como una caja con su nombre, a un Panel, donde el usuario podrá reorganizarlos arrastrando cada caja a la posición deseada, eliminando la que desee y añadiendo nuevas. Se obtiene un lista con los identificadores de las Familias de máquinas seleccionadas, así como una imagen del esquema formado por esa lista.
Imprimir	Clase donde se encapsulo todo lo necesario para realizar la impresión del informe. Toma el RichTextBox de la pantalla principal tal cual, define el área a imprimir, a partir de la configuración de impresión pasada y la va llenando con el contenido del RichTexBox. Cuando completa un área, pero aún no ha acabado el RichTextBox, informa al sistema de impresión de que aún quedan páginas por imprimir y vuelve a rellenar otro área de impresión, hasta que no queda más contenido en el RichTextBox.
Informe	Clase para instanciar los informes que la aplicación irá generando Servirá para almacenar los datos que se incluirán en el informe y para formatear el texto antes de mostrarlo y enviarlo a impresión. La fuente de los datos será la Línea actual en la aplicación.
Linea	Clase que representa la línea a dimensionar. Contendrá los atributos que definen sus propiedades, tales como los datos identificativos a mostrar en el informe de la propia línea (Nombre, cliente, proyecto, version, descripcion), la fecha en que se creó, las condiciones de dimensionamiento seleccionadas para esta línea, la selección de Familias de máquinas que forman su estructura y la selección de Máquinas resultado del dimensionamiento. Además, dispone de una variable para almacenar las familias utilizadas en ella en caso de que se exporte a un fichero.

Maquina	Clase que representa las Máquinas. Posee los atributos que reflejan las características de cada Máquina (nombre, descripción, Producción, Operarios, Potencia Eléctrica, Valor de Coste y Valor de Venta. También dispone de los métodos para actualizar dichos atributos así como para mostrarlos en texto, con o sin el Valor de Coste
RepresentacionFamilia	Clase para definir un control personalizado que representará a cada Familia en el esquema de selección de Familias de máquinas de la línea a dimensionar Serán "cajas" rectangulares, con las esquinas redondeadas
ServicioLineas	Esta clase es el gestor de las líneas a dimensionar. Alojará la línea sobre la que se está trabajando en la aplicación y todos los métodos para crearla, editarla, exportarla o importar una existente en fichero. Para la importación, gestionará la carga o no de las familias de máquinas incluidas en la línea si no se encuentran en el catálogo actual. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame
ServicioMaquinas	Esta clase es el gestor de las familias de máquinas y las propias máquinas Alojará el catálogo de Máquinas (listado de familias de máquinas) y todos los métodos para crearlos, editarlos, añadirlos o eliminarlos del catálogo y demás interacciones con ellos. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame.
VentanaPresentacion	Ventana con información de la aplicación, que se muestra brevemente durante el inicio de la aplicación

Estructuras

Imprimir.RANGOCARACTER	Almacena el rango de caracteres a imprimir
Imprimir.RANGOFORMATO	Almacena el rango ya formateado

Enumeraciones

ServicioMaquinas.TipoElemento	Tipo enumerado para que se pueda informar de qué tipo se trata, antes de crearlo incluso
---	--

Condiciones Clase

Clase para encapsular las condiciones de dimensionamiento, elegidas por el usuario, para una línea en particular. Formará una característica de esa línea y será utilizada para el cálculo de dimensionamiento. Cuenta con las variables para guardar: La producción mínima requerida, el paralelismo (cantidad máxima de máquinas en paralelo para conseguir la producción mínima) y las características a minimizar en la selección.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<code><SerializableAttribute></code> Public Class Condiciones

Inheritance [Object](#) → [Condiciones](#)

Constructores

Condiciones()	Constructor por defecto, necesario para la serialización.
Condiciones(String)	Constructor para ser llamado desde Linea. Inicializa todo por defecto

Propiedades

MinimizarPrimaria	Propiedad que se debe minimizar en la selección (primaria)
MinimizarSecundaria	Propiedad que se debe minimizar en la selección (secundaria)
Paralelismo	Cantidad de máquinas de la misma familia que se pueden tomar para cumplir la producción
ProduccionMin	Producción mínima requerida

Campos

_minimizarPrimaria	
_minimizarSecundaria	
_paralelismo	
_produccionMin	

Vea también

[Espacio de nombres DraftLine](#)

Condiciones Constructor

Constructor por defecto, necesario para la serialización.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New
```

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones(String) Constructor

Constructor para ser llamado desde Linea. Inicializa todo por defecto

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New (  
    nombre As String  
)
```

Parámetros

nombre [String](#)

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones.MinimizarPrimaria Propiedad

Propiedad que se debe minimizar en la selección (primaria)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property MinimizarPrimaria As String
    Get
    Set
```

Valor devuelto

[String](#)

_minimizarPrimaria

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones.MinimizarSecundaria Propiedad

Propiedad que se debe minimizar en la selección (secundaria)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property MinimizarSecundaria As String
    Get
    Set
```

Valor devuelto

[String](#)

_minimizarSecundaria

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones.Paralelismo Propiedad

Cantidad de máquinas de la misma familia que se pueden tomar para cumplir la producción

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Paralelismo As Integer
    Get
    Set
```

Valor devuelto

[Int32](#) paralelismo

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones.ProduccionMin Propiedad

Producción mínima requerida

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property ProduccionMin As Single
    Get
    Set
```

Valor devuelto

[Single](#) produccionMin

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones._minimizarPrimaria Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _minimizarPrimaria As String
```

Valor del campo

[String](#)

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones._minimizarSecundaria Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _minimizarSecundaria As String
```

Valor del campo

[String](#)

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones._paralelismo Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _paralelismo As Integer
```

Valor del campo

[Int32](#)

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Condiciones._produccionMin Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _produccionMin As Single
```

Valor del campo

[Single](#)

Vea también

[Condiciones Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar Clase

Esta clase encapsula todo lo relacionado con los cálculos de selección según las condiciones del usuario. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame. Tomará cada una de las Familias de máquinas, seleccionadas por el usuario para la línea, y seleccionará la máquina, de cada Familia, que mejor se adapte a las condiciones indicadas por el usuario. El resultado será una lista de Máquinas que conformarán la línea dimensionada.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Class Dimensionar

Inheritance [Object](#) → [Dimensionar](#)

Observaciones

Sólo permito una instancia (Singleton), ya que sólo se necesita un objeto calculador

Constructores

Dimensionar	
Dimensionar	Constructor de la clase. Inicializa las listas a utilizar durante los cálculos. Es privado para evitar crear nuevas instancias desde fuera de la propia clase (Singleton)

Propiedades

Instance	Método para obtener la instancia de este Singleton
--------------------------	--

Métodos

CreaNuevaMaquinaParalela	Función que, dado una máquina y un nº de paralelismo, devuelve una nueva Máquina con sus atributos multiplicados por ese número
Dimensiona	Función que realiza la selección de Máquinas, una por cada Familia de máquina requerida y cumpliendo las condiciones impuestas. Devolviendo un listado con las máquinas resultado

Anexos

Campos

<u>familia</u>	Variable para almacenar temporalmente la familia en la que escoger Máquinas
<u>instanciaDimensionar</u>	Almacena su propia y única instancia
<u>lineaDimensionada</u>	Lista para almacenar las máquinas finalmente seleccionados y devolverla
<u>listaCumplenProduccion</u>	Lista para almacenar temporalmente las máquinas que cumplen prod>condiciones(0)
<u>listaOrdenados</u>	'Lista para almacenar temporalmente las máquinas ordenadas según condiciones(1) y (2)
<u>sLineas</u>	Gestor de Informes
<u>sMaquinas</u>	Gestor de Máquinas (catálogo)

Vea también

[Espacio de nombres DraftLine](#)

Dimensionar Constructor

Inicializa los campos estáticos de la clase [Dimensionar](#)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Shared Sub](#) New

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar Constructor

Constructor de la clase. Inicializa las listas a utilizar durante los cálculos Es privado para evitar crear nuevas instancias desde fuera de la propia clase (Singleton)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private Sub](#) New

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.Instance Propiedad

Método para obtener la instancia de este Singleton

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Shared ReadOnly Property Instance As Dimensionar  
    Get
```

Valor devuelto

[Dimensionar](#)

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.CreaNuevaMaquinaParalela Método

Función que, dado una máquina y un nº de paralelismo, devuelve una nueva Máquina con sus atributos multiplicados por ese número

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Function CreaNuevaMaquinaParalela (  
    maquina As Maquina,  
    i As Integer  
) As Maquina
```

Parámetros

maquina [Maquina](#)

La máquina que se va a utilizar i veces en paralelo

i [Int32](#)

El número de veces que se utilizará la máquina en paralelo

Valor devuelto

[Maquina](#)

Una nueva Máquina representando el conjunto de i máquinas originales en paralelo

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.Dimensiona Método

Función que realiza la selección de Máquinas, una por cada Familia de máquina requerida y cumpliendo las condiciones impuestas Devolviendo un listado con las máquinas resultado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub Dimensiona
```

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.familia Campo

Variable para almacenar temporalmente la familia en la que escoger Máquinas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private familia As FamiliaMaquina
```

Valor del campo

[FamiliaMaquina](#)

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.instanciaDimensionar Campo

Almacena su propia y única instancia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Shared ReadOnly instanciaDimensionar As Dimensionar
```

Valor del campo

[Dimensionar](#)

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.lineaDimensionada Campo

Lista para almacenar las máquinas finalmente seleccionados y devolverla

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private lineaDimensionada As List(Of Maquina)
```

Valor del campo

[List](#)

([Maquina](#))

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.listaCumplenProduccion Campo

Lista para almacenar temporalmente las máquinas que cumplen prod>condiciones(0)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private listaCumplenProduccion As List(Of Maquina)
```

Valor del campo

[List](#)

([Maquina](#))

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.listaOrdenados Campo

'Lista para almacenar temporalmente las máquinas ordenadas según condiciones(1) y (2)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private listaOrdenados As List(Of Maquina)
```

Valor del campo

[List](#)

([Maquina](#))

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.sLineas Campo

Gestor de Informes

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private sLineas As ServicioLineas
```

Valor del campo

[ServicioLineas](#)

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

Dimensionar.sMaquinas Campo

Gestor de Máquinas (catálogo)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private sMaquinas As ServicioMaquinas
```

Valor del campo

[ServicioMaquinas](#)

Vea también

[Dimensionar Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina Clase

Clase que representa las Familias de máquinas. Posee los atributos que reflejan las características de cada Familia de Máquinas, así como una lista para albergar las Máquinas que pertenezcan a esta Familia.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
<SerializableAttribute>  
Public Class FamiliaMaquina
```

Inheritance [Object](#) → [FamiliaMaquina](#)

Observaciones

Serializable para poder realizar los guardados y cargados de archivo.

Constructores

FamiliaMaquina()	Constructor de la clase
FamiliaMaquina(String, String)	Constructor. Creo una nueva Instancia de la clase, con el nombre dado

Propiedades

Descripcion	Muestra o modifica la descripción de la familia
ID	Muestra o modifica el identificador de la familia
Maquinas	Muestra o modifica la lista de Máquinas asociadas a este familia
Nombre	Muestra o modifica el nombre de la familia

Métodos

AddMaquina	Método para añadir máquinas a la lista de esta familia
ToString	Función ToString sobreescrita, para obtener el Nombre en texto (Reemplaza a Object.ToString())

Campos

<u>_descripcion</u>	Descripción de la familia
<u>_ID</u>	Identificador de la familia
<u>_maquinas</u>	Lista de Máquinas asociadas a est Familia
<u>_nombre</u>	Nombre de la familia

Vea también [Espacio de nombres DraftLine](#)

FamiliaMaquina Constructor

Constructor de la clase

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Sub New

Observaciones

Sin parámetros para permitir la serialización XML

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina(String, String) Constructor

Constructor. Creo una nueva Instancia de la clase, con el nombre dado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Sub New (*nombre* **As** String,*descripcion* **As** String)

Parámetros

nombre [String](#)

El nombre de la Familia de máquinas

descripcion [String](#)

Descripción de la Familia de máquinas

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina.Descripcion Propiedad

Muestra o modifica la descripción de la familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Descripcion As String
    Get
    Set
```

Valor devuelto

[String](#) La descripción de la familia

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina.ID Propiedad

Muestra o modifica el identificador de la familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property ID As Guid
    Get
    Set
```

Valor devuelto

[Guid](#) El identificador de la familia

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina.Maquinas Propiedad

Muestra o modifica la lista de Máquinas asociadas a este familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Maquinas As List(Of Maquina)
    Get
    Set
```

Valor devuelto

[List\(Maquina\)](#) La lista de Máquinas

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina.Nombre Propiedad

Muestra o modifica el nombre de la familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Nombre As String
    Get
    Set
```

Valor devuelto

[String](#) El nombre de la familia

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina.AddMaquina Método

Método para añadir máquinas a la lista de esta familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub AddMaquina (  
    maquina As Maquina  
)
```

Parámetros

maquina [Maquina](#)

La máquina a añadir

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina.ToString Método

Función ToString sobreescrita, para obtener el Nombre en texto

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Overrides Function ToString As String
```

Valor devuelto

[String](#)

Nombre

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina._descripcion Campo

Descripción de la familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _descripcion As String
```

Valor del campo

[String](#)

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina._ID Campo

Identificador de la familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _ID As Guid
```

Valor del campo

[Guid](#)

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina._maquinas Campo

Lista de Máquinas asociadas a est Familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _maquinas As List(Of Maquina)
```

Valor del campo

[List](#)

([Maquina](#))

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FamiliaMaquina._nombre Campo

Nombre de la familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _nombre As String
```

Valor del campo

[String](#)

Vea también

[FamiliaMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormAcercaDe Clase

Formulario informativo que muestra datos sobre la aplicación Muestra: Nombre, Versión, Compañía y una descripción

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
Public NotInheritable Class FormAcercaDe Inherits Form

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
 → [ContainerControl](#) → [Form](#) → FormAcercaDe

Observaciones

Código creado por Visual Studio y posteriormente modificado por mi

Constructores

FormAcercaDe	Inicializa una nueva instancia de la clase FormAcercaDe
------------------------------	---

Métodos

Aceptar_Click	Método que captura el evento de pulsar sobre el botón aceptar, para cerrar el formulario
FormAcercaDe_Load	Método que captura el evento de carga del formulario. Carga los textos para mostrar la información
InitializeComponent	

Campos

components	
----------------------------	--

Vea también

[Espacio de nombres DraftLine](#)

FormAcercaDe Constructor

Inicializa una nueva instancia de la clase [FormAcercaDe](#)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New
```

Vea también

[FormAcercaDe Clase](#)

[Espacio de nombres DraftLine](#)

FormAcercaDe.Aceptar_Click Método

Método que captura el evento de pulsar sobre el botón aceptar, para cerrar el formulario

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Aceptar_Click (  
    sender As Object,  
    e As EventArgs  
)
```

Parámetros

sender [Object](#)

El botón aceptar

e [EventArgs](#)

El clickado sobre él

Vea también

[FormAcercaDe Clase](#)

[Espacio de nombres DraftLine](#)

FormAcercaDe.FormAcercaDe_Load Método

Método que captura el evento de carga del formulario. Carga los textos para mostrar la información

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub FormAcercaDe_Load (  
    sender As Object,  
    e As EventArgs  
)
```

Parámetros

sender [Object](#)

El propio formulario

e [EventArgs](#)

Load, la carga del formulario

Vea también

[FormAcercaDe Clase](#)

[Espacio de nombres DraftLine](#)

FormAcercaDe.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[FormAcercaDe Clase](#)

[Espacio de nombres DraftLine](#)

FormAcercaDe.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private components As IContainer
```

Valor del campo

[IContainer](#)

Vea también

[FormAcercaDe Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones Clase

Formulario para la selección de las condiciones de dimensionamiento de la línea. Contendrá la información en cuanto a la producción mínima requerida, el paralelismo de máquinas (cantidad máxima de máquinas trabajando en paralelo para cumplir la producción mínima), y las características a minimizar en la selección.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class FormCondiciones
    Inherits Form
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
 → [ContainerControl](#) → [Form](#) → [FormCondiciones](#)

Constructores

FormCondiciones	Constructor con parámetros. Toma los valores de condiciones para inicializar sus campos
---------------------------------	---

Propiedades

minimizarPrimaria	Propiedad que se debe minimizar en la selección (primaria)
minimizarSecundaria	Propiedad que se debe minimizar en la selección (secundaria)
paralelismo	Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la producción
produccionMin	Producción mínima requerida

Métodos

Aceptar_Click	Método que maneja el evento de pulsar el botón de aceptar. Comprueba los datos introducidos, los asigna a sus variables para que puedan ser recuperados, y marca el resultado del diálogo
Cancelar_Click	Método que captura la cancelación del formulario por parte del usuario
FormCondiciones_Load	Método que maneja el evento de carga del formulario. Toma los valores de sus variables y los muestra en los controles
InitializeComponent	
RoundButton_CheckedChanged	Método que controla que sólo se pueda seleccionar una condición secundaria si se ha seleccionado una condición primaria. Si la condición primaria está en Nothing, fuerza a que la secundaria también lo esté
TextBox_KeyPress	Método para controlar que en textBox sólo se introduzcan números

Campos

_minimizarPrimaria	Propiedad que se debe minimizar en la selección (primaria)
_minimizarSecundaria	Propiedad que se debe minimizar en la selección (secundaria)
_paralelismo	Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la producción
_produccionMin	Producción mínima requerida
components	

Vea también

[Espacio de nombres DraftLine](#)

FormCondiciones Constructor

Constructor con parámetros. Toma los valores de condiciones para inicializar sus campos

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New (paralelismo As Integer, produccionMin As Single, minimizarPrimaria As String,  
    minimizarSecundaria As String)
```

Parámetros

paralelismo [Int32](#)

Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la producción

produccionMin [Single](#)

Producción mínima requerida

minimizarPrimaria [String](#)

Propiedad que se debe minimizar en la selección (primaria)

minimizarSecundaria [String](#)

Propiedad que se debe minimizar en la selección (secundaria)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.minimizarPrimaria Propiedad

Propiedad que se debe minimizar en la selección (primaria)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property minimizarPrimaria As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.minimizarSecundaria Propiedad

Propiedad que se debe minimizar en la selección (secundaria)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property minimizarSecundaria As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.paralelismo Propiedad

Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la producción

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property paralelismo As Integer
    Get
    Set
```

Valor de propiedad

[Int32](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.produccionMin Propiedad

Producción mínima requerida

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property produccionMin As Single
    Get
    Set
```

Valor de propiedad

[Single](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.Aceptar_Click Método

Método que maneja el evento de pulsar el botón de aceptar. Comprueba los datos introducidos, los asigna a sus variables para que puedan ser recuperados, y marca el resultado del diálogo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Aceptar_Click (  
    sender As Object,  
    e As EventArgs  
)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

FormCondiciones.Cancelar_Click Método

Método que captura la cancelación del formulario por parte del usuario

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Cancelar_Click (  
    sender As Object,  
    e As EventArgs  
)
```

Parámetros

sender [Object](#) El botón de cancelar

e [EventArgs](#) El click sobre el botón

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.FormCondiciones_Load Método

Método que maneja el evento de carga del formulario. Toma los valores de sus variables y los muestra en los controles

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub FormCondiciones_Load (  
    sender As Object,  
    e As EventArgs  
)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.RoundButton_CheckedChanged Método

Método que controla que sólo se pueda seleccionar una condición secundaria si se ha seleccionado una condición primaria. Si la condición primaria está en Nothing, fuerza a que la secundaria también lo esté

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub RoundButton_CheckedChanged (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El radioButton RB1N que represenata Ninguno para la condición primaria

e [EventArgs](#) El cambio del estado del radioButton

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.TextBox_KeyPress Método

Método para controlar que en textBox sólo se introduzcan números

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub TextBox_KeyPress (sender As Object,e As KeyPressEventArgs)
```

Parámetros

sender [Object](#) textBoxProduccion y textBoxParalelismo

e [KeyPressEventArgs](#) Teclas pulsadas del teclado en el campo de texto de esos textbox

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones._minimizarPrimaria Campo

Propiedad que se debe minimizar en la selección (primaria)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _minimizarPrimaria As String
```

Valor del campo

[String](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones._minimizarSecundaria Campo

Propiedad que se debe minimizar en la selección (secundaria)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _minimizarSecundaria As String
```

Valor del campo

[String](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones._paralelismo Campo

Cantidad de máquinas de la misma Familia que se pueden tomar para cumplir la producción

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _paralelismo As Integer
```

Valor del campo

[Int32](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones._produccionMin Campo

Producción mínima requerida

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _produccionMin As Single
```

Valor del campo

[Single](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormCondiciones.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private components **As** IContainer

Valor del campo

[IContainer](#)

Vea también

[FormCondiciones Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea Clase

Formulario para introducir los datos informativos de la línea a dimensionar. Estos datos se mostrarán al principio del informe sobre la línea dimensionada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class FormDatosLinea
    Inherits Form
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
→ [ContainerControl](#) → [Form](#) → [FormDatosLinea](#)

Constructores

FormDatosLinea	Constructor Toma los datos que le pasan para colocarlos de base y permitir editarlos.
--------------------------------	---

Propiedades

Cliente	Texto que aparecerá en el campo CLIENTE del informe de la línea
Descripcion	Texto que aparecerá en el campo DESCRIPCION del informe de la línea
Nombre	Texto que aparecerá en el campo NOMBRE del informe de la línea
Proyecto	Texto que aparecerá en el campo PROYECTO del informe de la línea
Version	Texto que aparecerá en el campo VERSION del informe de la línea

Métodos

Aceptar_Click	Método que captura la acción de aceptar y crea y rellena el diccionario con los textos de los textbox para dejarlos disponibles para el formulario que lo ha llamado. Marco el resultado del diálogo = OK y cierro este formulario
Cancelar_Click	Método que captura el evento de pulsar sobre el botón cancelar Se pierden los posibles datos introducidos y se cierra el formulario Marca el resultado del diálogo = Cancel
InitializeComponent	

Anexos

Campos

<u>_cliente</u>	Texto que aparecerá en el campo CLIENTE del informe de la línea
<u>_descripcion</u>	Texto que aparecerá en el campo DESCRIPCION del informe de la línea
<u>_nombre</u>	Texto que aparecerá en el campo NOMBRE del informe de la línea
<u>_proyecto</u>	Texto que aparecerá en el campo PROYECTO del informe de la línea
<u>_version</u>	Texto que aparecerá en el campo VERSION del informe de la línea
<u>components</u>	

Vea también

[Espacio de nombres DraftLine](#)

FormDatosLinea Constructor

Constructor Toma los datos que le pasan para colocarlos de base y permitir editarlos.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New (Nombre As String, Cliente As String, Proyecto As String, Version As String,
    Descripcion As String)
```

Parámetros

Nombre [String](#)

Cliente [String](#)

Proyecto [String](#)

Version [String](#)

Descripcion [String](#)

FormDatosLinea.Cliente Propiedad

Texto que aparecerá en el campo CLIENTE del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Cliente As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.Descripcion Propiedad

Texto que aparecerá en el campo DESCRIPCION del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Descripcion As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.Nombre Propiedad

Texto que aparecerá en el campo NOMBRE del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Nombre As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.Proyecto Propiedad

Texto que aparecerá en el campo PROYECTO del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<pre>Public Property Proyecto As String Get Set</pre>

Valor de propiedad

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.Version Propiedad

Texto que aparecerá en el campo VERSION del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<pre>Public Property Version As String Get Set</pre>

Valor de propiedad

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.Aceptar_Click Método

Método que captura la acción de aceptar y crea y rellena el diccionario con los textos de los textbox para dejarlos disponibles para el formulario que lo ha llamado. Marco el resultado del diálogo = OK y cierro este formulario

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Aceptar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) Botón aceptar

e [EventArgs](#) Pulsar el botón

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.Cancelar_Click Método

Método que captura el evento de pulsar sobre el botón cancelar Se pierden los posibles datos introducidos y se cierra el formulario Marca el resultado del diálogo = Cancel

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Cancelar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón cancelar

e [EventArgs](#) El click sobre el botón

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea._cliente Campo

Texto que aparecerá en el campo CLIENTE del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _cliente As String
```

Valor del campo

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea._descripcion Campo

Texto que aparecerá en el campo DESCRIPCION del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _descripcion As String
```

Valor del campo

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea._nombre Campo

Texto que aparecerá en el campo NOMBRE del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _nombre As String
```

Valor del campo

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea._proyecto Campo

Texto que aparecerá en el campo PROYECTO del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _proyecto As String
```

Valor del campo

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea._version Campo

Texto que aparecerá en el campo VERSION del informe de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _version As String
```

Valor del campo

[String](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormDatosLinea.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private components **As** IContainer

Valor del campo

[IContainer](#)

Vea también

[FormDatosLinea Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia Clase

Formulario para introducir los datos de la nueva Familia de máquinas a crear o editar una existente

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class FormFamilia
    Inherits Form
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
→ [ContainerControl](#) → [Form](#) → [FormFamilia](#)

Constructores

FormFamilia	Construtor del formulario. Obtiene la instancia del gestor del catálogo y, si se trata de editar una familia existente, toma los valores.
-----------------------------	---

Propiedades

Descripcion	Almacena la descripción introducida
Nombre	Almacena el nombre introducido

Métodos

Aceptar_Click	Método que captura el evento de que el usuario haga click sobre el botón aceptar Si está en "modo edición", actualiza el campo descripción (el nombre será el mismo). Si está en "modo crear", comprueba que se haya introducido un nombre al menos, y no coincida con el nombre de ninguna familia existente en el catálogo. Marca el resultado del diálogo a OK
Cancelar_Click	Método que captura el evento de pulsar sobre el botón cancelar Se pierden los posibles datos introducidos y se cierra el formulario Marca el resultado del diálogo a Cancel
InitializeComponent	

Campos

Anexos

<u>_descripcion</u>	Almacena la descripción introducida
<u>_nombre</u>	Almacena el nombre introducido
<u>components</u>	
<u>nombreFamilia</u>	Variable que almacena el nombre de la Familia, si se trata de una edición
<u>sMaquinas</u>	La instancia del gestor de catalogo (ServicioMaquinas)

Vea también

[Espacio de nombres DraftLine](#)

FormFamilia Constructor

Constructor del formulario. Obtiene la instancia del gestor del catálogo y, si se trata de editar una familia existente, toma los valores.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New (Optional nombreFamilia As String = Nothing)
```

Parámetros

nombreFamilia [String](#) (Optional)

String opcional con el nombre de la Familia a editar

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia.Descripcion Propiedad

Almacena la descripción introducida

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Descripcion As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia.Nombre Propiedad

Almacena el nombre introducido

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Nombre As String
    Get
    Set
```

Valor de propiedad

[String](#)

FormFamilia.Aceptar_Click Método

Método que captura el evento de que el usuario haga click sobre el botón aceptar Si está en "modo edición", actualiza el campo descripción (el nombre será el mismo). Si está en "modo crear", comprueba que se haya introducido un nombre al menos, y no coincida con el nombre de ninguna familia existente en el catálogo. Marca el resultado del diálogo a OK

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Aceptar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón aceptar

e [EventArgs](#) El click sobre el botón

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia.Cancelar_Click Método

Método que captura el evento de pulsar sobre el botón cancelar Se pierden los posibles datos introducidos y se cierra el formulario Marca el resultado del diálogo a Cancel

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Cancelar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

El botón cancelar

e [EventArgs](#)

El click sobre el botón

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia._descripcion Campo

Almacena la descripción introducida

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _descripcion As String
```

Valor del campo

[String](#)

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia._nombre Campo

Almacena el nombre introducido

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _nombre As String
```

Valor del campo

[String](#)

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private components As IContainer
```

Valor del campo

[IContainer](#)

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia.nombreFamilia Campo

Variable que almacena el nombre de la Familia, si se trata de una edición

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private nombreFamilia As String
```

Valor del campo

[String](#)

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormFamilia.sMaquinas Campo

La instancia del gestor de catalogo (ServicioMaquinas)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<code>Private sMaquinas As ServicioMaquinas</code>

Valor del campo

[ServicioMaquinas](#)

Vea también

[FormFamilia Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina Clase

Formulario para introducir los datos de la nueva Máquina a crear o editar una existente

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class FormMaquina
    Inherits Form
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
→ [ContainerControl](#) → [Form](#) → [FormMaquina](#)

Constructores

FormMaquina	Construtor del formulario. Obtiene la instancia del gestor de catálogo y, si se trata de editar una máquina existente, toma los valores.
-----------------------------	--

Propiedades

Descripcion	Almacena la descripción de la máquina
EsEdicion	Marca que indica si se trata de una edición de una máquina existente o no
Familia	Variable que almacena la familia destino de esta máquina
ID	Variable que almacena el ID de la máquina a editar si se trata de una edición
Listaprop	Almacena la lista de propiedades de la máquina
Nombre	Almacena el nombre de la máquina

Métodos

Anexos

Aceptar_Click	Método que captura el evento de que el usuario haga click sobre el botón aceptar Comprueba que se ha introducido al menos un nombre, y, si está en modo editar, que ese nombre no exista ya en la Familia de Máquinas destino. Toma todos los valores introducidos y marca el resultado del diálogo a OK
Cancelar_Click	Método que captura el evento de pulsar sobre el botón cancelar Se pierden los posibles datos introducidos y se cierra el formulario Marca el resultado del diálogo a Cancel
InitializeComponent	
TextBox_KeyPress	Método para controlar que en textBox sólo se introduzcan números

Campos

_descripcion	Almacena la descripción de la máquina
_esEdicion	Marca que indica si se trata de una edición de una máquina existente o no
_familia	Variable que almacena la familia destino de esta máquina
_ID	Variable que almacena el ID de la máquina a editar si se trata de una edición
_listaprop	Almacena la lista de propiedades de la máquina
_nombre	Almacena el nombre de la máquina
components	
nombreMaquina	Variable que almacena el nombre de la máquina, si se trata de una edición
sMaquinas	La instancia del gestor de catálogo (ServicioMaquinas)

Vea también

[Espacio de nombres DraftLine](#)

FormMaquina Constructor

Constructor del formulario. Obtiene la instancia del gestor de catálogo y, si se trata de editar una máquina existente, toma los valores.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New (familia As FamiliaMaquina,
               Optional nombreMaquina As String = Nothing)
```

Parámetros

familia [FamiliaMaquina](#)

nombreMaquina [String](#) (Optional) String opcional con el nombre de la máquina a editar

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.Descripcion Propiedad

Almacena la descripción de la máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Descripcion As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.EsEdicion Propiedad

Marca que indica si se trata de una edición de una máquina existente o no

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property EsEdicion As Boolean
    Get
    Set
```

Valor de propiedad

[Boolean](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.Familia Propiedad

Variable que almacena la familia destino de esta máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Familia As FamiliaMaquina
    Get
    Set
```

Valor de propiedad

[FamiliaMaquina](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.ID Propiedad

Variable que almacena el ID de la máquina a editar si se trata de una edición

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<pre>Public Property ID As Guid Get Set</pre>

Valor de propiedad

[Guid](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.Listaprop Propiedad

Almacena la lista de propiedades de la máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<pre>Public Property Listaprop As List(Of Single) Get Set</pre>

Valor de propiedad

[List](#)

[\(Single\)](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.Nombre Propiedad

Almacena el nombre de la máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Nombre As String
    Get
    Set
```

Valor de propiedad

[String](#)

FormMaquina.Aceptar_Click Método

Método que captura el evento de que el usuario haga click sobre el botón aceptar Comprueba que se ha introducido al menos un nombre, y, si está en modo editar, que ese nombre no exista ya en la Familia de Máquinas destino. Toma todos los valores introducidos y marca el resultado del diálogo a OK

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Aceptar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón aceptar

e [EventArgs](#) El click sobre el botón

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.Cancelar_Click Método

Método que captura el evento de pulsar sobre el botón cancelar Se pierden los posibles datos introducidos y se cierra el formulario Marca el resultado del diálogo a Cancel

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Cancelar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

El botón cancelar

e [EventArgs](#)

El click sobre el botón

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.TextBox_KeyPress Método

Método para controlar que en textBox sólo se introduzcan números

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub TextBox_KeyPress (sender As Object,e As KeyPressEventArgs)
```

Parámetros

sender [Object](#)

textBoxProduccion y textBoxParalelismo

e [KeyPressEventArgs](#)

Teclas pulsadas del teclado en el campo de texto de esos textbox

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina._descripcion Campo

Almacena la descripción de la máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _descripcion As String
```

Valor del campo

[String](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina._esEdicion Campo

Marca que indica si se trata de una edición de una máquina existente o no

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _esEdicion As Boolean
```

Valor del campo

[Boolean](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina._familia Campo

Variable que almacena la familia destino de esta máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _familia As FamiliaMaquina
```

Valor del campo

[FamiliaMaquina](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina._ID Campo

Variable que almacena el ID de la máquina a editar si se trata de una edición

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _ID As Guid
```

Valor del campo

[Guid](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina._listaprop Campo

Almacena la lista de propiedades de la máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _listaprop As List(Of Single)
```

Valor del campo

[List](#)

([Single](#))

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina._nombre Campo

Almacena el nombre de la máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _nombre As String
```

Valor del campo

[String](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private components As IContainer
```

Valor del campo

[IContainer](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.nombreMaquina Campo

Variable que almacena el nombre de la máquina, si se trata de una edición

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private nombreMaquina As String
```

Valor del campo

[String](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormMaquina.sMaquinas Campo

La instancia del gestor de catálogo (ServicioMaquinas)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private sMaquinas As ServicioMaquinas
```

Valor del campo

[ServicioMaquinas](#)

Vea también

[FormMaquina Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal Clase

Esta clase es el formulario base de la aplicación. Se carga al iniciar la aplicación y muestra los modos Editar catálogo, Dimensionar Línea y ayuda, desde donde trabajará el usuario. Contiene los controles (botones, menús, etc.) para acceder a todas las funcionalidades de la aplicación. Desde ella se crean los demás formularios que se le presentarán al usuario. Es este formulario el que tomará los datos de esos formularios para luego utilizar el resto de clases y cumplir los casos de uso definidos.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class FormPrincipal
    Inherits Form
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
→ [ContainerControl](#) → [Form](#) → [FormPrincipal](#)

Constructores

FormPrincipal	Inicializa una nueva instancia de la clase FormPrincipal
-------------------------------	--

Métodos

AcercaDe_Click	Método que captura el evento de pulsar sobre el botón de Acerca de... y muestra la ventana de información
Borrar_Click	Método que elimina la familia de máquinas o la máquina seleccionada en el TreeView, capturando el evento desde el menú contextual del treeview, desde el elemento correspondiente del menú principal, o desde el botón borrar
CambiaPanel	Método que cambia el panel central en función del modo en que se encuentre la aplicación
CargaCatalogo	Método que llama a la carga de catálogo desde servicioMaquinas, actualizando, además, el estado devuelto
CargarDatosEnTreeView	Método que actualiza el TreeView para mostrar el listado de máquinas del catálogo
Condiciones_Click	Método que muestra el formulario para seleccionar las condiciones de dimensionamiento y realizar el dimensionamiento

<u>DatosLinea_Click</u>	Método que captura la pulsación sobre el botón para editar los datos del informe
<u>Editar_Click</u>	Método que captura el evento de clicar para editar un módulo Comprueba si hay un elemento seleccionado Si es una familia, muestra el formulario de editar familia Si es una máquina, muestra el formulario de editar máquina
<u>ExportarCatalogo_Click</u>	Método que captura la pulsación sobre el botón de exportar catálogo de máquinas
<u>ExportarLinea_Click</u>	Método que captura el evento para exportar la línea actual a un fichero. Abre el diálogo para seleccionar el fichero destino y con esa información llama al método correspondiente del servicioLineas.
<u>Form_FormClosing</u>	Método que maneja el evento de cierre del formulario Para controlar si hay cambios en la catálogo y si se desean guardar o no
<u>FormPrincipal_Load</u>	Método Constructor del formulario principal
<u>GuardarCatalogo_Click</u>	Método que captula la pulsación sobre el botón de guardar catálogo Guarda el estado de del catálogo actual en el archivo local a la aplicación
<u>ImportarCatalogo_Click</u>	Método que captura la pulsación sobre el botón de importar catálogo de máquinas
<u>ImportarLinea_Click</u>	Método que captura el evento para importar una línea guardada en un fichero. Abre el diálogo para seleccionar el fichero a importar. Llama al servicioLineas para que compruebe si existen en catálogo actual las máquinas necesarias para la línea a importar. Si existen, le indica al servicioLinea que acepte la línea importada, si no, consulta al usuario si quiere añadir las máquinas de la línea importada al catálogo. Si no acepta, no se carga la línea. Si se ha cargado la línea, actualiza el informe que se muestra de la nueva línea y abre el formulario que muestra el esquema de la nueva línea para que el usuario la revise y acepte o modifique.
<u>Imprimir_Click</u>	Método que captura la pulsación sobre el botón para imprimir el informe. Abre el diálogo para configurar la impresora y llama al método imprime de la clase que encapsula el tratamiento de impresión, pasándole el informe y la configuración de impresora
<u>InitializeComponent</u>	

<u>ModoAyuda</u>	Método que cambia el Modo de la interfaz a Ayuda, capturando el botón pulsado
<u>ModoCatalogo</u>	Método que cambia el Modo de la interfaz a Editar Catálogo, capturando el botón pulsado
<u>ModoDimensionar</u>	Método que cambia el Modo de la interfaz a Dimensionar, capturando el botón pulsado
<u>MostrarValorCoste_Click</u>	Método que captura el evento de pulsar sobre el botón de Mostrar y ocultar el valor de coste. Cambia booleano que define si se mostrará o no y el estado de los botones
<u>NuevaFamilia_Click</u>	Método que abre un nuevo formulario para pedir la información de la nueva familia de máquinas, con la que podrá crearla
<u>NuevaLinea_Click</u>	Método que captura la pulsación sobre los botones para crear una nueva linea. Avisa de que se perderán los datos de la linea anterior y, en caso de aceptar, crea una nueva linea por defecto y actualiza el informe y lo muestra en el richtextbox del formulario
<u>NuevaMaquina_Click</u>	Abre un nuevo formulario para crear una nueva máquinas y añadirla a la Familia seleccionada
<u>NuevoCatalogo_Click</u>	Método que captura la pulsación sobre el botón de olvidar catálogo Elimina todos los modelos del catálogo actual tras confirmación del usuario Pero no guarda los cambios en el fichero local
<u>Salir_Click</u>	Método que captura la pulsación sobre el botón salir del menú archivo
<u>SeleccionFamilias_Click</u>	Método que captura la pulsación sobre el botón para seleccionar las Familias de máquinas, que conformarán la linea, a tener en cuenta en el dimensionamiento
<u>TreeViewCatalogo_AfterSelected</u>	Método que captura la selección en el treeview para actualizar correspondientemente la ventana de descripción
<u>TreeViewCatalogo_KeyPress</u>	Método para permitir borrar desde el TreeView con el botón Supr del teclado
<u>TreeViewCatalogo_MouseClick</u>	Método para manejar el menú contextual del botón derecho en el TreeView

Campos

<u>catalogoLocal</u>	Dirección del archivo del catálogo de modelos actual
--------------------------------------	--

Anexos

<u>catalogoModificado</u>	Centinela para saber si se han guardado los cambios en el catálogo actual
<u>components</u>	
<u>dimensionar</u>	Clase que realiza los cálculos
<u>formAcercaDe</u>	Almacena el Formulario con info sobre esta aplicación
<u>formCond</u>	Almacena el Formulario para introducir las condiciones para el dimensionamiento
<u>formDatosLinea</u>	Almacena el Formulario para introducir la info de los datos de la línea
<u>formFam</u>	Almacena el Formulario para introducir la info de la nueva familia
<u>formMag</u>	Almacena el Formulario para introducir la info de la nueva máquina
<u>formSelfFam</u>	Almacena el Formulario para introducir la selección de familias para la línea
<u>imprimir</u>	Gestor de impresión
<u>informe</u>	Instancia de la clase donde se generará el informe
<u>rutaPDF</u>	Ruta del manual de usuario
<u>sLineas</u>	Gestor de Líneas
<u>sMaquinas</u>	Gestor de Máquinas (catálogo)
<u>valorCosteVisible</u>	Booleano para definir si se quiere mostrar o no el valor de coste

Vea también

[Espacio de nombres DraftLine](#)

FormPrincipal Constructor

Inicializa una nueva instancia de la clase [FormPrincipal](#)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New
```

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.AcercaDe_Click Método

Método que captura el evento de pulsar sobre el botón de Acerca de... y muestra la ventana de información

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub AcercaDe_Click (  
    sender As Object,  
    e As EventArgs  
)
```

Parámetros

sender [Object](#) El botón del menú de Acerca de...

e [EventArgs](#) El click sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.Borrar_Click Método

Método que elimina la familia de máquinas o la máquina seleccionada en el TreeView, capturando el evento desde el menú contextual del treeview, desde el elemento correspondiente del menú principal, o desde el botón borrar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Borrar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.CambiaPanel Método

Método que cambia el panel central en función del modo en que se encuentre la aplicación

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub CambiaPanel (modo As String)
```

Parámetros

modo [String](#)

El Modo de la interfaz al que hay que cambiar.

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.CargaCatalogo Método

Método que llama a la carga de catálogo desde servicioMaquinas, actualizando, además, el estado devuelto

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub CargaCatalogo (dirArchivo As String)
```

Parámetros

dirArchivo [String](#)

Dirección del archivo que contiene el catálogo a cargar

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.CargarDatosEnTreeView Método

Método que actualiza el TreeView para mostrar el listado de máquinas del catálogo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub CargarDatosEnTreeView
```

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.Condiciones_Click Método

Método que muestra el formulario para seleccionar las condiciones de dimensionamiento y realizar el dimensionamiento

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Condiciones_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) Botón Condiciones

e [EventArgs](#) El clickado sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.DatosLinea_Click Método

Método que captura la pulsación sobre el botón para editar los datos del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub DatosLinea_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón Datos

e [EventArgs](#) El clickado sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.Editar_Click Método

Método que captura el evento de clicar para editar un módulo Comprueba si hay un elemento seleccionado Si es una familia, muestra el formulario de editar familia Si es una máquina, muestra el formulario de editar máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Editar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.ExportarCatalogo_Click Método

Método que captura la pulsación sobre el botón de exportar catálogo de máquinas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub ExportarCatalogo_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón exportar catálogo

e [EventArgs](#) El clickado sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.ExportarLinea_Click Método

Método que captura el evento para exportar la línea actual a un fichero. Abre el diálogo para seleccionar el fichero destino y con esa información llama al método correspondiente del servicioLineas.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub ExportarLinea_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.Form_FormClosing Método

Método que maneja el evento de cierre del formulario Para controlar si hay cambios en la catálogo y si se desean guardar o no

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Form_FormClosing (sender As Object,e As FormClosingEventArgs)
```

Parámetros

sender [Object](#) Este formulario

e [FormClosingEventArgs](#) El cierre del formulario

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.FormPrincipal_Load Método

Método Constructor del formulario principal

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub FormPrincipal_Load (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

FormPrincipal.GuardarCatalogo_Click Método

Método que captura la pulsación sobre el botón de guardar catálogo Guarda el estado de del catálogo actual en el archivo local a la aplicación

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub GuardarCatalogo_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

El botón guardar catálogo

e [EventArgs](#)

El clickado sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.ImportarCatalogo_Click Método

Método que captura la pulsación sobre el botón de importar catálogo de máquinas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub ImportarCatalogo_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

El botón importar catálogo

e [EventArgs](#)

El clickado sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.ImportarLinea_Click Método

Método que captura el evento para importar una línea guardada en un fichero. Abre el diálogo para seleccionar el fichero a importar. Llama al servicioLineas para que compruebe si existen en catálogo actual las máquinas necesarias para la línea a importar. Si existen, le indica al servicioLinea que acepte la línea importada, si no, consulta al usuario si quiere añadir las máquinas de la línea importada al catálogo. Si no acepta, no se carga la línea. Si se ha cargado la línea, actualiza el informe que se muestra de la nueva línea y abre el formulario que muestra el esquema de la nueva línea para que el usuario la revise y acepte o modifique.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub ImportarLinea_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

El botón de importar o el elementeo correspondiente del menú principal

e [EventArgs](#)

El clicado sobre el botón o elemento de menú

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.Imprimir_Click Método

Método que captura la pulsación sobre el botón para imprimir el informe. Abre el diálogo para configurar la impresora y llama al método imprime de la clase que encapsula el tratamiento de impresión, pasándole el informe y la configuración de impresora

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Imprimir_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.ModoAyuda Método

Método que cambia el Modo de la interfaz a Ayuda, capturando el botón pulsado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub ModoAyuda (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón del menú Manual de Usuario

e [EventArgs](#) El click sobre el botón. No usado

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.ModoCatalogo Método

Método que cambia el Modo de la interfaz a Editar Catálogo, capturando el botón pulsado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub ModoCatalogo (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón de modo pulsado

e [EventArgs](#) El click sobre el botón. No usado

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.Mododimensionar Método

Método que cambia el Modo de la interfaz a Dimensionar, capturando el botón pulsado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Mododimensionar (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón de modo pulsado

e [EventArgs](#) El click sobre el botón. No usado

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.MostrarValorCoste_Click Método

Método que captura el evento de pulsar sobre el botón de Mostrar y ocultar el valor de coste. Cambia booleano que define si se mostrará o no y el estado de los botones

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub MostrarValorCoste_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón del menú Mostrar

e [EventArgs](#) El click sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.NuevaFamilia_Click Método

Método que abre un nuevo formulario para pedir la información de la nueva familia de máquinas, con la que podrá crearla

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub NuevaFamilia_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón de crear nueva Familia o la selección correspondiente en el ToolStripMenu

e [EventArgs](#) El click sobre el botón

FormPrincipal.NuevaLinea_Click Método

Método que captura la pulsación sobre los botones para crear una nueva linea. Avisa de que se perderán los datos de la linea anterior y, en caso de aceptar, crea una nueva linea por defecto y actualiza el informe y lo muestra en el richtextbox del formulario

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub NuevaLinea_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) Los botones nueva linea

e [EventArgs](#) El click sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.NuevaMaquina_Click Método

Abre un nuevo formulario para crear una nueva máquinas y añadirla a la Familia seleccionada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub NuevaMaquina_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.NuevoCatalogo_Click Método

Método que captura la pulsación sobre el botón de olvidar catálogo Elimina todos los modelos del catálogo actual tras confirmación del usuario Pero no guarda los cambios en el fichero local

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub NuevoCatalogo_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón olvidar catálogo

e [EventArgs](#) El clickado sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.Salir_Click Método

Método que captura la pulsación sobre el botón salir del menú archivo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Salir_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) ToolStripMenuItem Salir

e [EventArgs](#) Click sobre el elemento

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.SeleccionFamilias_Click Método

Método que captura la pulsación sobre el botón para seleccionar las Familias de máquinas, que conformarán la línea, a tener en cuenta en el dimensionamiento

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub SeleccionFamilias_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón Selección de Familias de Máquinas

e [EventArgs](#) El clickado sobre el botón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.TreeViewCatalogo_AfterSelected Método

Método que captura la selección en el treeview para actualizar correspondientemente la ventana de descripción

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub TreeViewCatalogo_AfterSelected (sender As Object,e As TreeViewEventArgs)
```

Parámetros

sender [Object](#) El TreeView

e [TreeViewEventArgs](#) La selección de un nodo

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.TreeViewCatalogo_KeyPress Método

Método para permitir borrar desde el TreeView con el botón Supr del teclado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub TreeViewCatalogo_KeyPress (sender As Object,e As KeyEventArgs)
```

Parámetros

sender [Object](#)

e [KeyEventArgs](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.TreeViewCatalogo_MouseClick Método

Método para manejar el menú contextual del botón derecho en el TreeView

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub TreeViewCatalogo_MouseClick (sender As Object,e As MouseEventArgs)
```

Parámetros

sender [Object](#) El TreeView

e [MouseEventArgs](#) El Evento Click del ratón

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.catalogoLocal Campo

Dirección del archivo del catálogo de modelos actual

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private catalogoLocal As String
```

Valor del campo

[String](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.catalogoModificado Campo

Centinela para saber si se han guardado los cambios en el catálogo actual

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private catalogoModificado As Boolean
```

Valor del campo

[Boolean](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private components As IContainer
```

Valor del campo

[IContainer](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.dimensionar Campo

Clase que realiza los cálculos

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private dimensionar **As** Dimensionar

Valor del campo

[Dimensionar](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.formAcercaDe Campo

Almacena el Formulario con info sobre esta aplicación

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private formAcercaDe **As** FormAcercaDe

Valor del campo

[FormAcercaDe](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.formCond Campo

Almacena el Formulario para introducir las condiciones para el dimensionamiento

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private formCond As FormCondiciones
```

Valor del campo

[FormCondiciones](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.formDatosLinea Campo

Almacena el Formulario para introducir la info de los datos de la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private formDatosLinea As FormDatosLinea
```

Valor del campo

[FormDatosLinea](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.formFam Campo

Almacena el Formulario para introducir la info de la nueva familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private formFam As FormFamilia
```

Valor del campo

[FormFamilia](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.formMaq Campo

Almacena el Formulario para introducir la info de la nueva máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private formMaq As FormMaquina
```

Valor del campo

[FormMaquina](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.formSelFam Campo

Almacena el Formulario para introducir la selección de familias para la línea

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private formSelFam As FormSelecFam
```

Valor del campo

[FormSelecFam](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.imprimir Campo

Gestor de impresión

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private imprimir As Imprimir
```

Valor del campo

[Imprimir](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.informe Campo

Instancia de la clase donde se generará el informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private informe As Informe
```

Valor del campo

[Informe](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.rutaPDF Campo

Ruta del manual de usuario

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private rutaPDF As String
```

Valor del campo

[String](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.sLineas Campo

Gestor de Líneas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private](#) sLineas [As](#) ServicioLineas

Valor del campo

[ServicioLineas](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.sMaquinas Campo

Gestor de Máquinas (catálogo)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private](#) sMaquinas [As](#) ServicioMaquinas

Valor del campo

[ServicioMaquinas](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormPrincipal.valorCosteVisible Campo

Booleano para definir si se quiere mostrar o no el valor de coste

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private valorCosteVisible As Boolean
```

Valor del campo

[Boolean](#)

Vea también

[FormPrincipal Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam Clase

Clase con el formulario para que el usuario seleccione las Familias de máquinas que compondrán la Linea, así como su orden. Esa selección se realiza añadiendo Familias de máquinas representadas gráficamente como una caja con su nombre, a un Panel, donde el usuario podrá reorganizarlos arrastrando cada caja a la posición deseada, eliminando la que desee y añadiendo nuevas. Se obtiene un lista con los identificadores de las Familias de máquinas seleccionadas, así como una imagen del esquema formado por esa lista.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class FormSelecFam
    Inherits Form
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
→ [ContainerControl](#) → [Form](#) → [FormSelecFam](#)

Constructores

FormSelecFam	Constructor. Inicializa los componentes. También la listaPrevia. Se configura el formulario y, sobre todo, el panel con DoubleBuffered, para evitar el parpadeo en el redibujado
------------------------------	--

Propiedades

Esquema	La imagen esquema de la linea según sus familias de máquinas seleccionadas
ListaPrevia	Almacena la lista de Familias seleccionadas previamente si se la pasa el llamante
SeleccionModClase	La lista para almacenar los ID's de las famiias seleccionados para dimensionar

Métodos

Aceptar_Click	Método que captura la aceptación del usuario. Captura la imagen del esquema, prepara la lista de id's de familias de máquinas seleccionadas y devuelve el resultado de OK para informar al método llamante
AddFamilia_Click	Método para añadir la familia seleccionada en el TreeView al esquema del panel y a la selección de Familias de máquinas
Caja_MouseDown	Método que captura el evento de pulsar un botón del ratón sobre una de las representaciones Si es botón izquierdo, activa booleano que indica

	que se está arrastrando dicha representación, y guarda la representación seleccionada así como su posición original. Si es botón derecho, muestra el menú con el botón de eliminar ese modelo de los seleccionados
<u>Caja_MouseHover</u>	Método que captura el evento de pasar el ratón por encima de una de las representaciones, para mostrar los datos de la Familia de máquinas que representa
<u>Caja_MouseMove</u>	Método para gestionar el movimiento de una representación y las nuevas posiciones de ella y las demás, así como de las familias de máquinas asociadas
<u>Caja_MouseUp</u>	Método que captura el evento de dejar de pulsar el botón izq del ratón
<u>Cancelar_Click</u>	Método que captura la cancelación del formulario pro parte del usuario. Se pierden los cambios y se marca el resultado del diálogo = Cancel
<u>CargaListaPrevia</u>	Método que carga la selección previa de Familias en el panel y en la selección actual, para mantener la consistencia con el estado actual de la aplicación
<u>CargarDatosEnTreeView</u>	Método que actualiza el TreeView para mostrar el listado de Familias de máquinas disponibles del catálogo
<u>DibujarFlecha</u>	Método que dibuja las flechas entre representaciones. Dibuja una linea personalizada con una punta de flecha en su extremo, entre dos puntos dados y en el componente gráfico pasado
<u>Eliminar_Click</u>	Método que captura la pulsación sobre el elemento eliminar del menú contextual de las representaciones, para llamar al método que eliminará dicha representación y su familia asociada en los listados de seleccionadas
<u>EliminarCaja</u>	Método para eliminar una de las familias Seleccionadas de la lista que se está seleccionando, así como de su lista de representaciones y, por ello, del panel donde se muestran las representaciones
<u>EsUltimoDeLaFila</u>	Método que comprueba si una caja es la última de la fila en la representación del panel Si la posición de su extremo derecho más el ancho de la que sería la siguiente caja más la separación se pasan del extremo derecho del panel, entonces sí, es la última de la fila
<u>FormSelecfam_Load</u>	Método que maneja la carga del formulario. Toma la instancia del servicioMaquinas, carga las Familias de Máquinas disponibles para elección en el TreeView y, si hay una selección previa, llama al método para cargarla en el panel y en la lista actual

Anexos

InitializeComponent	
PanelRepresentaciones Paint	Método que captura cada redibujado del panel para crear las representaciones y las flechas que las conectan
ReordenarRepresentacion	Método que recoloca la posición de las cajas según su orden en la listaRepresentaciones, colocándolas en la siguiente línea si se pasan de ancho del panel donde se dibujan
SetDoubleBuffered	Método para forzar, mediante reflexión, el cambio de la propiedad protegida "DoubleBuffered" en los controles Y así evitar el parpadeo

Campos

_esquema	Para almacenar la captura en imagen del esquema dibujado en el panel
_listaPrevia	Almacena la lista de Familias seleccionadas previamente si se la pasa el llamante
_seleccionModFam	La lista para almacenar los ID's de las familias seleccionados para dimensionar
alturaCaja	Altura de la caja que representará la familia seleccionad
alturaEsquema	Almacena la altura de la imagen, que varía según cuantos elementos tenga, para la captura en imagen. Inicialmente alt.mín es la caja más la separación arriba y abajo
arrastrandoCaja	Boolean si la representación se está arrastrando
cajaSeleccionada	La representación seleccionada en el panel
components	
limiteSeleccion	Tamaño máximo de la selección defamilias. Impuesto para contar con un límite superior real
listaRepresentaciones	Lista de Representaciones de Familias
listaSeleccionFam	Lista de Familias de maquinas seleccionadas
longitudCaja	Longitud de la caja que representará a lafamilia seleccionada
posicionOriginalCaja	Guarda la posición original de la representación que se está arrastrando
separacion	Almacena la separación de la representación entre modelos
sMasquinas	Gestor de Maquinas (catalogo)

FormSelecFam Constructor

Constructor. Inicializa los componentes. También la listaPrevia. Se configura el formulario y, sobre todo, el panel con DoubleBuffered, para evitar el parpadeo en el redibujado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Public Sub](#) New

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.Eschema Propiedad

La imagen esquema de la linea según sus familias de máquinas seleccionadas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Public Property](#) Esquema [As](#) Image
 [Get](#)
 [Set](#)

Valor de propiedad

[Image](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.ListaPrevía Propiedad

Almacena la lista de Familias seleccionadas previamente si se la pasa el llamante

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property ListaPrevía As List(Of Guid)
    Get
    Set
```

Valor de propiedad

[List\(Guid\)](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.SeleccionModClase Propiedad

La lista para almacenar los ID's de las familias seleccionados para dimensionar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property SeleccionModClase As List(Of Guid)
    Get
    Set
```

Valor de propiedad

[List\(Guid\)](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.Aceptar_Click Método

Método que captura la aceptación del usuario. Captura la imagen del esquema, prepara la lista de id's de familias de máquinas seleccionadas y devuelve el resultado de OK para informar al método llamante

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Aceptar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón de aceptar

e [EventArgs](#) El click sobre el botón

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.AddFamilia_Click Método

Método para añadir la familia seleccionada en el TreeView al esquema del panel y a la selección de Familias de máquinas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub AddFamilia_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón de añadir Familia

e [EventArgs](#) El click sobre el botón

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecfam.Caja_MouseDown Método

Método que captura el evento de pulsar un botón del ratón sobre una de las representaciones Si es botón izquierdo, activa booleano que indica que se está arrastrando dicha representación, y guarda la representación seleccionada así como su posición original. Si es botón derecho, muestra el menú con el botón de eliminar ese modelo de los seleccionados

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Caja_MouseDown (sender As Object,e As MouseEventArgs)
```

Parámetros

sender [Object](#) Los botones del ratón

e [MouseEventArgs](#) la pulsación

FormSelecfam.Caja_MouseHover Método

Método que captura el evento de pasar el ratón por encima de una de las representaciones, para mostrar los datos de la Familia de máquinas que representa

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Caja_MouseHover (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) La representación (caja)

e [EventArgs](#) Posicionarse encima

Vea también

[FormSelecfam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.Caja_MouseMove Método

Método para gestionar el movimiento de una representación y las nuevas posiciones de ella y las demás, así como de las familias de máquinas asociadas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Caja_MouseMove (sender As Object,e As MouseEventArgs)
```

Parámetros

sender [Object](#) La representación que se está moviendo

e [MouseEventArgs](#) El movimiento del ratón

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.Caja_MouseUp Método

Método que captura el evento de dejar de pulsar el botón izq del ratón

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Caja_MouseUp (sender As Object,e As MouseEventArgs)
```

Parámetros

sender [Object](#) La caja que estaba pulsada

e [MouseEventArgs](#) Soltar el botón

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.Cancelar_Click Método

Método que captura la cancelación del formulario pro parte del usuario. Se pierden los cambios y se marca el resultado del diálogo = Cancel

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Cancelar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El botón de cancelar

e [EventArgs](#) El click sobre el botón

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.CargaListaPrevia Método

Método que carga la selección previa de Familias en el panel y en la selección actual, para mantener la consistencia con el estado actual de la aplicación

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub CargaListaPrevia
```

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.CargarDatosEnTreeView Método

Método que actualiza el TreeView para mostrar el listado de Familias de máquinas disponibles del catálogo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub CargarDatosEnTreeView
```

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.DibujarFlecha Método

Método que dibuja las flechas entre representaciones. Dibuja una linea personalizada con una punta de flecha en su extremo, entre dos puntos dados y en el componente gráfico pasado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub DibujarFlecha (g As Graphics,puntoInicio As Point,puntoFin As Point)
```

Parámetros

g [Graphics](#) El componente gráfico donde se dibujará la flecha

puntoInicio [Point](#) El punto inicial de donde parte la flecha

puntoFin [Point](#) El punto final donde acaba la flecha

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.Eliminar_Click Método

Método que captura la pulsación sobre el elemento eliminar del menú contextual de las representaciones, para llamar al método que eliminará dicha representación y su familia asociada en los listados de seleccionadas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub Eliminar_Click (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#) El elemento del menú contextual de la representación

e [EventArgs](#) El click sobre el elemento

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.EliminarCaja Método

Método para eliminar una de las familias Seleccionadas de la lista que se está seleccionando, así como de su lista de representaciones y, por ello, del panel donde se muestran las representaciones

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub EliminarCaja (caja As RepresentacionFamilia)
```

Parámetros

caja [RepresentacionFamilia](#) Representación de Familia que se pretende eliminar

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.EsUltimoDeLaFila Método

Método que comprueba si una caja es la última de la fila en la representación del panel Si la posición de su extremo derecho más el ancho de la que sería la siguiente caja más la separación se pasan del extremo derecho del panel, entonces sí, es la última de la fila

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private Function EsUltimoDeLaFila (caja **As** RepresentacionFamilia) **As** Boolean

Parámetros

caja [RepresentacionFamilia](#) La caja de la cual se quiere comprobar si es la última de la fila

Valor devuelto

[Boolean](#) true si es la última, false si no

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.FormSelecFam_Load Método

Método que maneja la carga del formulario. Toma la instancia del servicioMaquinas, carga las Familias de Máquinas disponibles para elección en el TreeView y, si hay una selección previa, llama al método para cargarla en el panel y en la lista actual

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Sub FormSelecFam_Load (sender **As** Object,e **As** EventArgs)

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.PanelRepresentaciones_Paint Método

Método que captura cada redibujado del panel para crear las representaciones y las flechas que las conectan

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub PanelRepresentaciones_Paint (sender As Object,e As PaintEventArgs)
```

Parámetros

sender [Object](#)

El Panel donde se muestran las representaciones de Modelos

e [PaintEventArgs](#)

El evento de redibujado del panel

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.ReordenarRepresentacion Método

Método que recoloca la posición de las cajas según su orden en la listaRepresentaciones, colocándolas en la siguiente línea si se pasan de ancho del panel donde se dibujan

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub ReordenarRepresentacion
```

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.SetDoubleBuffered Método

Método para forzar, mediante reflexión, el cambio de la propiedad protegida "DoubleBuffered" en los controles Y así evitar el parpadeo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub SetDoubleBuffered (control As Control,valor As Boolean)
```

Parámetros

control [Control](#)

El control del cual se quiere modificar el parámetro protegido

valor [Boolean](#)

El valor a poner en el parámetro (boolean)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam._esquema Campo

Para almacenar la captura en imagen del esquema dibujado en el panel

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _esquema As Image
```

Valor del campo

[Image](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam._listaPrevía Campo

Almacena la lista de Familias seleccionadas previamente si se la pasa el llamante

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _listaPrevía As List(Of Guid)
```

Valor del campo

[List](#)

[\(Guid\)](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam._seleccionModFam Campo

La lista para almacenar los ID's de las familias seleccionados para dimensionar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _seleccionModFam As List(Of Guid)
```

Valor del campo

[List](#)

([Guid](#))

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.alturaCaja Campo

Altura de la caja que representará la familia seleccionad

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private alturaCaja As Integer
```

Valor del campo

[Int32](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.alturaEsquema Campo

Almacena la altura de la imagen, que varía según cuantos elementos tenga, para la captura en imagen. Inicialmente alt.mín es la caja más la separación arriba y abajo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private alturaEsquema As Integer
```

Valor del campo

[Int32](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.arrastrandoCaja Campo

Boolean si la representación se está arrastrando

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private arrastrandoCaja As Boolean
```

Valor del campo

[Boolean](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.cajaSeleccionada Campo

La representación seleccionada en el panel

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private cajaSeleccionada As RepresentacionFamilia
```

Valor del campo

[RepresentacionFamilia](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private components As IContainer
```

Valor del campo

[IContainer](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.limiteSeleccion Campo

Tamaño máximo de la selección defamilias. Impuesto para contar con un límite superior real

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private limiteSeleccion As Integer
```

Valor del campo

[Int32](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.listaRepresentaciones Campo

Lista de Representaciones de Familias

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private listaRepresentaciones As List(Of RepresentacionFamilia)
```

Valor del campo

[List](#)

[\(RepresentacionFamilia\)](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.listaSeleccionFam Campo

Lista de Familias de maquinas seleccionadas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private listaSeleccionFam As List(Of FamiliaMaquina)
```

Valor del campo

[List](#)

([FamiliaMaquina](#))

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.longitudCaja Campo

Longitud de la caja que representará a la familia seleccionada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private longitudCaja As Integer
```

Valor del campo

[Int32](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.posicionOriginalCaja Campo

Guarda la posición original de la representación que se está arrastrando

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private posicionOriginalCaja As Point
```

Valor del campo

[Point](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.separacion Campo

Almacena la separación de la representación entre modelos

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Const separacion As Integer = 35
```

Valor del campo

[Int32](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

FormSelecFam.sMasquinas Campo

Gestor de Maquinas (catalogo)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private sMasquinas As ServicioMaquinas
```

Valor del campo

[ServicioMaquinas](#)

Vea también

[FormSelecFam Clase](#)

[Espacio de nombres DraftLine](#)

Imprimir Clase

Clase donde se encapsulo todo lo necesario para realizar la impresión del informe. Toma el RichTextBox de la pantalla principal tal cual, define el área a imprimir, a partir de la configuración de impresión pasada y la va llenando con el contenido del RichTextBox. Cuando completa un área, pero aún no ha acabado el RichTextBox, informa al sistema de impresión de que aún quedan páginas por imprimir y vuelve a rellenar otro área de impresión, hasta que no queda más contenido en el RichTextBox.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Class Imprimir

Inheritance [Object](#) → Imprimir

Constructores

Imprimir	Constructor de la clase
--------------------------	-------------------------

Métodos

Imprime	Método que crea un nuevo documento de impresión, asignando la configuración de la impresora que proviene de un printDialog y establece el manejador del evento PrintPage del documento. Reinicia el índice actual y comienza a imprimir
PrintPageHandler	Método que maneja el evento de imprimir el documento. Establece el rango de formato y los rectángulos en los que se imprimirá.
SendMessage	Función para interactuar con la API de Windows Pasará el mensaje a la ventana indicada. El mensaje irá desde un puntero de memoria a otro (Rango)

Campos

EM_FORMATRANGE	Constante para mensajes de Windows
indiceCaracterActual	Índice del caracter que se está imprimiendo
textoParaImprimir	RichTextBox que contendrá el texto formateado a imprimir

Vea también

[Espacio de nombres DraftLine](#)

Imprimir Constructor

Constructor de la clase

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New (RichTextBoxTexto As RichTextBox)
```

Parámetros

RichTextBoxTexto [RichTextBox](#) El RichTextBox cuyo contenido imprimiré

Vea también

[Imprimir Clase](#)

[Espacio de nombres DraftLine](#)

Imprimir.Imprime Método

Método que crea un nuevo documento de impresión, asignando la configuración de la impresora que proviene de un printDialog y establece el manejador del evento PrintPage del documento. Reinicia el índice actual y comienza a imprimir

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub Imprime (configuracionImpresion As PrinterSettings)
```

Parámetros

configuracionImpresion [PrinterSettings](#) La configuración seleccionada para imprimir, desde el printDialog

Vea también

[Imprimir Clase](#)

[Espacio de nombres DraftLine](#)

Imprimir.PrintPageHandler Método

Método que maneja el evento de imprimir el documento. Establece el rango de formato y los rectángulos en los que se imprimirá.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub PrintPageHandler (sender As Object,e As PrintPageEventArgs)
```

Parámetros

sender [Object](#)

e [PrintPageEventArgs](#)

Imprimir.SendMessage Método

Función para interactuar con la API de Windows Pasará el mensaje a la ventana indicada. El mensaje irá desde un puntero de memoria a otro (Rango)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Shared Function SendMessage (hWnd As IntPtr,msg As Integer,  
wParam As IntPtr,IParam As IntPtr) As Integer
```

Parámetros

hWnd [IntPtr](#) A donde se enviará el mensaje (al manejador del contenido a imprimir)

msg [Int32](#) El mensaje que se enviará

wParam [IntPtr](#) Parámetro inicial

IParam [IntPtr](#) Parámetro final

Valor devuelto [Int32](#)

Vea también

[Imprimir Clase](#)

[Espacio de nombres DraftLine](#)

Imprimir.EM_FORMATRANGE Campo

Constante para mensajes de Windows

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Const EM_FORMATRANGE As Integer = 1081
```

Valor del campo

[Int32](#)

Vea también

[Imprimir Clase](#)

[Espacio de nombres DraftLine](#)

Imprimir.indiceCaracterActual Campo

Índice del caracter que se está imprimiendo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private indiceCaracterActual As Integer
```

Valor del campo

[Int32](#)

Vea también

[Imprimir Clase](#)

[Espacio de nombres DraftLine](#)

Imprimir.textoParaImprimir Campo

RichTextBox que contendrá el texto formateado a imprimir

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private textoParaImprimir As RichTextBox
```

Valor del campo

[RichTextBox](#)

Vea también

[Imprimir Clase](#)

[Espacio de nombres DraftLine](#)

Imprimir.RANGOCARACTER Estructura

Almacena el rango de caracteres a imprimir

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Structure RANGOCARACTER
```

Inheritance [Object](#) → [ValueType](#) → Imprimir.RANGOCARACTER

Campos

cpMax	
cpMin	

Vea también

[Espacio de nombres DraftLine](#)

Imprimir.RANGOCARACTER.cpMax Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public cpMax As Integer
```

Valor del campo

[Int32](#)

Vea también

[Imprimir.RANGOCARACTER Estructura](#)

[Espacio de nombres DraftLine](#)

Imprimir.RANGOCARACTER.cpMin Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public cpMin As Integer
```

Valor del campo

[Int32](#)

Vea también

[Imprimir.RANGOCARACTER Estructura](#)

[Espacio de nombres DraftLine](#)

Imprimir.RANGOFORMATO Estructura

Almacena el rango ya formateado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private Structure](#) RANGOFORMATO

Inheritance [Object](#) → [ValueType](#) → Imprimir.RANGOFORMATO

Campos

chrg	
hdc	
hdcTarget	
rc	
rcPage	

Vea también

[Espacio de nombres DraftLine](#)

Imprimir.RANGOFORMATO.chrg Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public chrg As Imprimir.RANGOCARACTER
```

Valor del campo

[Imprimir.RANGOCARACTER](#)

Vea también

[Imprimir.RANGOFORMATO Estructura](#)

[Espacio de nombres DraftLine](#)

Imprimir.RANGOFORMATO.hdc Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public hdc As IntPtr
```

Valor del campo

[IntPtr](#)

Vea también

[Imprimir.RANGOFORMATO Estructura](#)

[Espacio de nombres DraftLine](#)

Imprimir.RANGOFORMATO.hdcTarget Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public hdcTarget As IntPtr
```

Valor del campo

[IntPtr](#)

Vea también

[Imprimir.RANGOFORMATO Estructura](#)

[Espacio de nombres DraftLine](#)

Imprimir.RANGOFORMATO.rc Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public rc As Rectangle
```

Valor del campo

[Rectangle](#)

Vea también

[Imprimir.RANGOFORMATO Estructura](#)

[Espacio de nombres DraftLine](#)

Imprimir.RANGOFORMATO.rcPage Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public rcPage As Rectangle
```

Valor del campo

[Rectangle](#)

Vea también

[Imprimir.RANGOFORMATO Estructura](#)

[Espacio de nombres DraftLine](#)

Informe Clase

Clase para instanciar los informes que la aplicación irá generando Servirá para almacenar los datos que se incluirán en el informe y para formatear el texto antes de mostrarlo y enviarlo a impresión. La fuente de los datos será la Linea actual en la aplicación.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Class Informe

Inheritance [Object](#) → Informe

Constructores

Informe	Constructor de la clase. Inicializa las variables.
-------------------------	--

Propiedades

Contenido	RichTextBox donde construyo el texto del informe para darle formato
valorCosteVisible	Booleano para definir si se quiere mostrar o no el valor de coste

Métodos

CalculaTotales	Método que realiza el cálculo de los valores totales resumen para la linea dimensionada La producción total de la línea será la del componente (máquina o máquinas en paralelo) que tenga la menor producción. El resto de propiedades, serán la suma total
CrealInforme	Método que genera el contenido del richtextbox, añadiendo paso a paso los datos con formato.
Margenes	Método que aplica los márgenes al texto del informe
Separador	Método que añade un separador al texto del informe
TextoEnNegrita	Método que encapsula el formato en negrita, para usar en el texto del informe

TituloSeccion	Método que encapsula el formato del título de sección, para usar en el texto del informe
-------------------------------	--

Campos

_contenido	RichTextBox donde construyo el texto del informe para darle formato
_valorCosteVisible	Booleano para definir si se quiere mostrar o no el valor de coste
margenDch	Márgen derecho del texto
margenIzg	Márgen izquierdo del texto
OperTotal	Operarios Totales necesarios
PETotal	Potencia Eléctrica Total de la linea dimensionada
pieDePagina	String con el texto para el pie de página del informe
prodTotal	Producción total de la linea dimensionada
sLineas	Gestor de Lineas
titulo	Título del informe
VCTotal	Valor de Coste Total de la linea dimensionada
versionApp	Guarda la Versión de esta propia aplicación
VVTotal	Valor de Venta total de la linea de la linea dimensionada

Vea también

[Espacio de nombres DraftLine](#)

Informe Constructor

Constructor de la clase. Inicializa las variables.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New
```

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.Contenido Propiedad

RichTextBox donde construyo el texto del informe para darle formato

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Contenido As RichTextBox
    Get
    Set
```

Valor de propiedad

[RichTextBox](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.valorCosteVisible Propiedad

Booleano para definir si se quiere mostrar o no el valor de coste

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property valorCosteVisible As Boolean
    Get
    Set
```

Valor de propiedad

[Boolean](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.CalculaTotales Método

Método que realiza el cálculo de los valores totales resumen para la línea dimensionada La producción total de la línea será la del componente (máquina o máquinas en paralelo) que tenga la menor producción. El resto de propiedades, serán la suma total

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub CalculaTotales
```

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.CreaInforme Método

Método que genera el contenido del richtextbox, añadiendo paso a paso los datos con formato.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Public Sub](#) CreaInforme

Informe.Margenes Método

Método que aplica los márgenes al texto del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private Sub](#) Margenes

Informe.Separador Método

Método que añade un separador al texto del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private Sub](#) Separador

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.TextoEnNegrita Método

Método que encapsula el formato en negrita, para usar en el texto del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub TextoEnNegrita (texto As String)
```

Parámetros

texto [String](#) El texto a poner en negrita

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.TituloSeccion Método

Método que encapsula el formato del título de sección, para usar en el texto del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub TituloSeccion (tituloSeccion As String)
```

Parámetros

tituloSeccion [String](#) El texto a poner en formato título de sección

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe._contenido Campo

RichTextBox donde construyo el texto del informe para darle formato

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _contenido As RichTextBox
```

Valor del campo

[RichTextBox](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe._valorCosteVisible Campo

Booleano para definir si se quiere mostrar o no el valor de coste

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _valorCosteVisible As Boolean
```

Valor del campo

[Boolean](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.margenDch Campo

Márgen derecho del texto

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Const margenDch As Integer = 10
```

Valor del campo

[Int32](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.margenIzq Campo

Márgen izquierdo del texto

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Const margenIzq As Integer = 20
```

Valor del campo

[Int32](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.OperTotal Campo

Operarios Totales necesarios

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private OperTotal **As** Single

Valor del campo

[Single](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.PETotal Campo

Potencia Eléctrica Total de la linea dimensionada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private PETotal **As** Single

Valor del campo

[Single](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.pieDePagina Campo

String con el texto para el pie de página del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Const pieDePagina As String = "Informe generado por la aplicación DraftLine. "
```

Valor del campo

[String](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.prodTotal Campo

Producción total de la linea dimensionada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private prodTotal As Single
```

Valor del campo

[Single](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.sLineas Campo

Gestor de Lineas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private sLineas As ServicioLineas
```

Valor del campo

[ServicioLineas](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.titulo Campo

Título del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Const titulo As String = ""
```

Valor del campo

[String](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.VCTotal Campo

Valor de Coste Total de la linea dimensionada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private VCTotal As Single
```

Valor del campo

[Single](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.versionApp Campo

Guarda la Versión de esta propia aplicación

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private versionApp As String
```

Valor del campo

[String](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Informe.VVTotal Campo

Valor de Venta total de la linea de la linea dimensionada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
Private VVTotal As Single

Valor del campo

[Single](#)

Vea también

[Informe Clase](#)

[Espacio de nombres DraftLine](#)

Linea Clase

Clase que representa la línea a dimensionar. Contendrá los atributos que definen sus propiedades, tales como los datos identificativos a mostrar en el informe de la propia línea (Nombre, cliente, proyecto, version, descripcion), la fecha en que se creó, las condiciones de dimensionamiento seleccionadas para esta línea, la selección de Familias de máquinas que forman su estructura y la selección de Máquinas resultado del dimensionamiento. Además, dispone de una variable para almacenar las familias utilizadas en ella en caso de que se exporte a un fichero.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
<SerializableAttribute>
Public Class Linea
```

Inheritance [Object](#) → [Linea](#)

Observaciones

Serializable para poder realizar los guardados y cargados de archivo.

Constructores

Linea	Constructor de la clase
-----------------------	-------------------------

Propiedades

Cliente	El Cliente de la línea que se está dimensionando (Dato de línea)
Condiciones	Las condiciones que se han tomado para la selección
Descripcion	La descripción de la línea que se está dimensionando (Dato de línea)
Esquema	Imagen con el esquema de las familias de máquinas escogidas
FamiliasLinea	Lista que almacenará las familias de máquinas usadas en esta línea en caso de que se quiera guardar en fichero para poder recuperarlos en un futuro
Fecha	La fecha de creación del informe
Nombre	El nombre de la línea que se está dimensionando (Dato de línea)

Anexos

Proyecto	El nombre del proyecto al que pertenece la línea que se está dimensionando (Dato de línea)
SeleccionFam	Lista que almacenará las ID de las familias seleccionados para dimensionar
seleccionMag	Lista de las máquinas seleccionadas
Version	La versión de la línea que se está dimensionando (Dato de línea)

Campos

cliente	
condiciones	Las condiciones que se han tomado para la selección
descripcion	
esquema	Imagen con el esquema de las familias de máquinas escogidas
familiasLinea	Lista que almacenará las familias de máquinas usadas en esta linea en caso de que se quiera guardar en fichero para poder recuperarlos en un futuro
fecha	La fecha de creación del informe
nombre	Datos informativos de la linea a dimensionar
proyecto	
seleccionFam	Lista que almacenará las ID de las familias seleccionados para dimensionar
seleccionMag	Lista de las máquinas seleccionadas
version	

Vea también

[Espacio de nombres DraftLine](#)

Linea Constructor

Constructor de la clase

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New
```

Observaciones

Sin parámetros para permitir la serialización XML

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Cliente Propiedad

El Cliente de la línea que se está dimensionando (Dato de línea)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Cliente As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Condiciones Propiedad

Las condiciones que se han tomado para la selección

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Condiciones As Condiciones
    Get
    Set
```

Valor de propiedad

[Condiciones](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Descripcion Propiedad

La descripción de la línea que se está dimensionando (Dato de línea)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Descripcion As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Eschema Propiedad

Imagen con el esquema de las familias de máquinas escogidas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Esquema As Image
    Get
    Set
```

Valor de propiedad

[Image](#)

Linea.FamiliasLinea Propiedad

Lista que almacenará las familias de máquinas usadas en esta linea en caso de que se quiera guardar en fichero para poder recuperarlos en un futuro

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property FamiliasLinea As List(Of FamiliaMaquina)
    Get
    Set
```

Valor de propiedad

[List\(FamiliaMaquina\)](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Fecha Propiedad

La fecha de creación del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Fecha As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Nombre Propiedad

El nombre de la línea que se está dimensionando (Dato de línea)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Nombre As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Proyecto Propiedad

El nombre del proyecto al que pertenece la línea que se está dimensionando (Dato de línea)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Proyecto As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.SeleccionFam Propiedad

Lista que almacenará las ID de las familias seleccionados para dimensionar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property SeleccionFam As List(Of Guid)
    Get
    Set
```

Valor de propiedad

[List\(Guid\)](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.seleccionMq Propiedad

Lista de las máquinas seleccionadas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property seleccionMq As List(Of Maquina)
    Get
    Set
```

Valor de propiedad

[List\(Maquina\)](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea.Version Propiedad

La versión de la línea que se está dimensionando (Dato de línea)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Version As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._cliente Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _cliente As String
```

Valor del campo

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._condiciones Campo

Las condiciones que se han tomado para la selección

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _condiciones As Condiciones
```

Valor del campo

[Condiciones](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

[Linea._descripcion Campo](#)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _descripcion As String
```

Valor del campo

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

[Linea._esquema Campo](#)

Imagen con el esquema de las familias de máquinas escogidas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _esquema As Image
```

Valor del campo

[Image](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._familiasLinea Campo

Lista que almacenará las familias de máquinas usadas en esta linea en caso de que se quiera guardar en fichero para poder recuperarlos en un futuro

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _familiasLinea As List(Of FamiliaMaquina)
```

Valor del campo

[List](#)

([FamiliaMaquina](#))

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._fecha Campo

La fecha de creación del informe

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _fecha As String
```

Valor del campo

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._nombre Campo

Datos informativos de la linea a dimensionar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _nombre As String
```

Valor del campo

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._proyecto Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _proyecto As String
```

Valor del campo

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._seleccionFam Campo

Lista que almacenará las ID de las familias seleccionados para dimensionar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _seleccionFam As List(Of Guid)
```

Valor del campo

[List](#)

([Guid](#))

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Linea._seleccionMaq Campo

Lista de las máquinas seleccionadas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _seleccionMaq As List(Of Maquina)
```

Valor del campo

[List](#)

([Maquina](#))

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

[Linea._version](#) Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _version As String
```

Valor del campo

[String](#)

Vea también

[Linea Clase](#)

[Espacio de nombres DraftLine](#)

Maquina Clase

Clase que representa las Máquinas. Posee los atributos que reflejan las características de cada Máquina (nombre, descripción, Producción, Operarios, Potencia Eléctrica, Valor de Coste y Valor de Venta. También dispone de los métodos para actualizar dichos atributos así como para mostrarlos en texto, con o sin el Valor de Coste

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
<SerializableAttribute>
Public Class Maquina
```

Inheritance [Object](#) → [Maquina](#)

Observaciones

Serializable para poder realizar los guardados y cargados de archivo

Constructores

Maquina()	Constructor de la clase
Maquina(String, String)	Constructor, Crea una nueva Instancia de la clase, con el nombre dado

Propiedades

Descripcion	La descripción de la Máquina
ID	El identificador de la Máquina
Nombre	El nombre de la Máquina
Operarios	Los operarios necesarios para hacer funcionar la Máquina
PotenciaElectrica	La potencia eléctrica de la Máquina
Produccion	La producción de la Máquina
ValorCoste	El coste económico, de fabricación o compra, de la Máquina

Anexos

ValorVenta	El valor de venta de la Máquina
----------------------------	---------------------------------

Métodos

AsignaPropiedades	Método que recibe como argumento una lista de valores para asignar a las propiedades de la Máquina
ToString	Función ToString para obtener una representación en texto de la Máquina (Reemplaza a Object.ToString())
ToStringNoCoste	Función ToString para obtener una representación en texto de la Máquina, pero sin mostrar el valor de coste

Campos

_descripcion	Descripción de la Máquina
_ID	Identificador de la Máquina
_nombre	Nombre de la Máquina
_operarios	Valor Operarios necesarios
_potenciaElectrica	Valor de Potencia eléctrica de la Máquina
_produccion	Valor de Producción
_valorCoste	Valor de Coste económico
_valorVenta	Valor de Venta de la Máquina

Vea también

[Espacio de nombres DraftLine](#)

Maquina Constructor

Constructor de la clase

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New
```

Observaciones

Sin parámetros para permitir la serialización XML

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina(String, String) Constructor

Constructor, Crea una nueva Instancia de la clase, con el nombre dado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub New (nombre As String,descripcion As String)
```

Parámetros

nombre [String](#)

El nombre de la Máquina

descripcion [String](#)

La descripción de la Máquina

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.Descripcion Propiedad

La descripción de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Description As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.ID Propiedad

El identificador de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property ID As Guid
    Get
    Set
```

Valor de propiedad

[Guid](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.Nombre Propiedad

El nombre de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Nombre As String
    Get
    Set
```

Valor de propiedad

[String](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.Operarios Propiedad

Los operarios necesarios para hacer funcionar la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Operarios As Single
    Get
    Set
```

Valor de propiedad

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.PotenciaElectrica Propiedad

La potencia eléctrica de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Property PotenciaElectrica **As** Single
 Get
 Set

Valor de propiedad

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.Produccion Propiedad

La producción de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Property Produccion **As** Single
 Get
 Set

Valor de propiedad

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.ValorCoste Propiedad

El coste económico, de fabricación o compra, de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<code>Public Property ValorCoste As Single Get Set</code>

Valor de propiedad

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.ValorVenta Propiedad

El valor de venta de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB
<code>Public Property ValorVenta As Single Get Set</code>

Valor de propiedad

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.AsignaPropiedades Método

Método que recibe como argumento una lista de valores para asignar a las propiedades de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub AsignaPropiedades (listaPropiedades As List(Of Single))
```

Parámetros

listaPropiedades [List\(Single\)](#)

Lista de valores de los atributos de la Máquina, almacenados como Single

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.ToString Método

Función ToString para obtener una representación en texto de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Overrides Function ToString As String
```

Valor devuelto

[String](#)

Representación en texto de la Máquina

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina.ToStringNoCoste Método

Función ToString para obtener una representación en texto de la Máquina, pero sin mostrar el valor de coste

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function ToStringNoCoste As String
```

Valor devuelto

[String](#)

Representación en texto de la Máquina sin valor de coste

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._descripcion Campo

Descripción de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _descripcion As String
```

Valor del campo

[String](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._ID Campo

Identificador de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

`Private _ID As Guid`

Valor del campo

[Guid](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._nombre Campo

Nombre de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

`Private _nombre As String`

Valor del campo

[String](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._operarios Campo

Valor Operarios necesarios

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _operarios As Single
```

Valor del campo

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._potenciaElectrica Campo

Valor de Potencia eléctrica de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _potenciaElectrica As Single
```

Valor del campo

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._produccion Campo

Valor de Producción

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _produccion As Single
```

Valor del campo

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._valorCoste Campo

Valor de Coste económico

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _valorCoste As Single
```

Valor del campo

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

Maquina._valorVenta Campo

Valor de Venta de la Máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _valorVenta As Single
```

Valor del campo

[Single](#)

Vea también

[Maquina Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia Clase

Clase para definir un control personalizado que representará a cada Familia en el esquema de selección de Familias de máquinas de la línea a dimensionar. Serán "cajas" rectangulares, con las esquinas redondeadas.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class RepresentacionFamilia
    Inherits Control
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [RepresentacionFamilia](#)

Constructores

RepresentacionFamilia	Constructor para asignar los eventos de cada caja
---------------------------------------	---

Propiedades

BordeColor	
ColorFondo	
ColorRatonEncima	
RadioBorde	
TextColor	

Métodos

InitializeComponent	
OnMouseEnter	Método para capturar el evento de cuando el ratón entra encima del control. Cambia el color del fondo y redibuja.

OnMouseLeave	Método para capturar el evento de cuando el ratón sale del control Vuelve al color del fondo original y redibuja
OnPaint	Método que dibuja el control (Reemplaza a Control.OnPaint(PaintEventArgs))

Campos

_colorBorde	Color del rectángulo redondeado
_colorFondo	Color del fondo del control
_colorRatonEncima	Color del fondo cuando el ratón pasa por encima
_colorTexto	Color del texto
_radioBorde	Almaceno el radio del rectángulo a dibujar
ColorFondoOriginal	Color del fondo original antes de pasar por encima
components	

Vea también

[Espacio de nombres DraftLine](#)

RepresentacionFamilia Constructor

Constructor para asignar los eventos de cada caja

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Public Sub](#) New

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.BordeColor Propiedad

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Public Property](#) BordeColor [As](#) Color
 [Get](#)
 [Set](#)

Valor de propiedad

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.ColorFondo Propiedad

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property ColorFondo As Color
    Get
    Set
```

Valor de propiedad

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.ColorRatonEncima Propiedad

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property ColorRatonEncima As Color
    Get
    Set
```

Valor de propiedad

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.RadioBorde Propiedad

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property RadioBorde As Integer
    Get
    Set
```

Valor de propiedad

[Int32](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.TextColor Propiedad

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property TextColor As Color
    Get
    Set
```

Valor de propiedad

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub InitializeComponent
```

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.OnMouseEnter Método

Método para capturar el evento de cuando el ratón entra encima del control Cambia el color del fondo y redibuja

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub OnMouseEnter (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.OnMouseLeave Método

Método para capturar el evento de cuando el ratón sale del control Vuelve al color del fondo original y redibuja

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub OnMouseLeave (sender As Object,e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.OnPaint Método

Método que dibuja el control

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Protected Overrides Sub OnPaint (  
    e As PaintEventArgs  
)
```

Parámetros

e [PaintEventArgs](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia._colorBorde Campo

Color del rectángulo redondeado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _colorBorde As Color
```

Valor del campo

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia._colorFondo Campo

Color del fondo del control

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _colorFondo As Color
```

Valor del campo

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia._colorRatonEncima Campo

Color del fondo cuando el ratón pasa por encima

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _colorRatonEncima As Color
```

Valor del campo

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia._colorTexto Campo

Color del texto

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _colorTexto As Color
```

Valor del campo

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia._radioBorde Campo

Almaceno el radio del rectángulo a dibujar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _radioBorde As Integer
```

Valor del campo

[Int32](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.ColorFondoOriginal Campo

Color del fondo original antes de pasar por encima

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private ColorFondoOriginal As Color
```

Valor del campo

[Color](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

RepresentacionFamilia.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private components **As** IContainer

Valor del campo

[IContainer](#)

Vea también

[RepresentacionFamilia Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas Clase

Esta clase es el gestor de las líneas a dimensionar. Alojara la línea sobre la que se está trabajando en la aplicación y todos los métodos para crearla, editarla, exportarla o importar una existente en fichero. Para la importación, gestionará la carga o no de las familias de máquinas incluidas en la línea si no se encuentran en el catálogo actual. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Class ServicioLineas

Inheritance [Object](#) → [ServicioLineas](#)

Constructores

ServicioLineas	Constructor privado. Sólo se genera la instancia desde la propia clase
--------------------------------	--

Propiedades

Instance	Método (propiedad) para acceder a la clase
Linea	La línea actual

Métodos

AceptaLinea	Método que acepta la linea recién importada, y la convierte en la Linea a trabajar Si así se le indica, se copiarán sus familias de máquinas en el catálogo actual
CargarDesde	Lee la linea desde un archivo y lo carga en la aplicación
GuardarEn	Guarda la linea en un archivo
NuevaLinea	Método que crea una nueva linea. Es llamado por el constructor de esta clase y por el método para generar una nueva linea a dimensionar Crea una nueva Linea con todos los valores por defecto

Campos

Anexos

<u>linea</u>	Almacena la Línea que se va a dimensionar
<u>lineaImportada</u>	Almacena temporalmente la línea que se quiere importar pero aún no comprobada
<u>instanciaServicioLineas</u>	Almacena su propia y única instancia
<u>sMaquinas</u>	Gestor de máquinas (catálogo)

Vea también

[Espacio de nombres DraftLine](#)

ServicioLineas Constructor

Constructor privado. Sólo se genera la instancia desde la propia clase

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub New
```

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.Instance Propiedad

Método (propiedad) para acceder a la clase

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Shared ReadOnly Property Instance As ServicioLineas  
    Get
```

Valor devuelto

[ServicioLineas](#)

la única instancia de la clase

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.Linea Propiedad

La línea actual

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Property Linea As Linea
    Get
    Set
```

Valor de propiedad

[Linea](#)

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.AceptaLinea Método

Método que acepta la línea recién importada, y la convierte en la Linea a trabajar Si así se le indica, se copiarán sus familias de máquinas en el catálogo actual

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function AceptaLinea (Optional cargarFamilias As Boolean = false) As Object
```

Parámetros

cargarFamilias [Boolean](#) (Optional) Boolean que indica si se desea cargar las familias de máquinas de la línea en el catálogo actual

Valor devuelto

[Object](#)

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.CargarDesde Método

Lee la línea desde un archivo y lo carga en la aplicación

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub CargarDesde (archivo As String)
```

Parámetros

archivo [String](#)

El archivo origen

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.GuardarEn Método

Guarda la línea en un archivo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub GuardarEn (archivo As String)
```

Parámetros

archivo [String](#)

El archivo destino

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.NuevaLinea Método

Método que crea una nueva línea. Es llamado por el constructor de esta clase y por el método para generar una nueva línea a dimensionar Crea una nueva Línea con todos los valores por defecto

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub NuevaLinea
```

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas._linea Campo

Almacena la Línea que se va a dimensionar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _linea As Linea
```

Valor del campo

[Linea](#)

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas._linealImportada Campo

Almacena temporalmente la línea que se quiere importar pero aún no comprobada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private _linealImportada As Linea
```

Valor del campo

[Linea](#)

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.instanciaServicioLineas Campo

Almacena su propia y única instancia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Shared instanciaServicioLineas As ServicioLineas
```

Valor del campo

[ServicioLineas](#)

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioLineas.sMaquinas Campo

Gestor de máquinas (catálogo)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private](#) sMaquinas [As](#) ServicioMaquinas

Valor del campo

[ServicioMaquinas](#)

Vea también

[ServicioLineas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas Clase

Esta clase es el gestor de las familias de máquinas y las propias máquinas Alojara el catálogo de Máquinas (listado de familias de máquinas) y todos los métodos para crearlos, editarlos, añadirlos o eliminarlos del catálogo y demás interacciones con ellos. Se tratará como Singleton para que sea siempre la misma instancia a la que se llame.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Public Class](#) ServicioMaquinas

Inheritance [Object](#) → [ServicioMaquinas](#)

Observaciones Sólo permito una instancia (Singleton), obligando así a mantener un estado consistente de la gestión de la biblioteca

Constructores

ServicioMaquinas	Constructor privado. Sólo se genera la instancia desde la propia clase Inicializa el catálogo
----------------------------------	---

Propiedades

Instance	Método (propiedad) para acceder a la clase
--------------------------	--

Métodos

ActualizaFamilia	Método para actualizar los atributos de una familia
ActualizaMaquina	Método que actualiza los atributos de una Maquina
BorraCatalogo	Método para borrar el listado de familias Borra el catálogo
CargaDesde	Lee el archivo de guardado de Familias de máquinas (catálogo) y los carga en la aplicación Devolviendo texto sobre el estado en que queda el catálogo
CompruebaNombre	Comprueba si alguna familia o máquina ya existe con un nombre dado
CreaFamilia	Método para crear Nuevas Familias de máquinas
CrearMaquina	Método para crear nuevas máquinas y añadirlas a la familia correspondiente

Anexos

DelFamilia	Elimina la familia de máquinas de la lista (catálogo) que coincida con el nombre
DelMaquina	Elimina la máquina de la lista en la familia a la que pertenece
GetAtributosMaquina	Función que buscará la máquina dada perteneciente a la familia dada y devolverá sus atributos de descripción, operadores, pot.Elect., Valor Coste y Valor Venta
GetDatosFamilia	Devuelve un texto ya formateado con las propiedades de la familia indicada
GetDatosMaquina	Devuelve un texto ya formateado con las propiedades de la máquina indicada, perteneciente a la familia también indicada. Muestra o no el valor de coste según se indique
GetDescripcion	Devuelve el atributo descripción de la familia o máquina correspondiente al nombre pasado como argumento Si se incluye el argumento nombreMaq, se refiere al atributo de una máquina
GetFamiliaID	Función que devuelve una familia, dado su ID
GetFamilias	Función que devuelve la lista Familias de máquinas
GetIDFamilia	Función que devuelve el ID de una familia en particular
GetIDMaquina	Función que devuelve el ID de una máquina en particular
GetNombrePorID	Función que devuelve el nombre correspondiente de una familia, dado su ID
GuardarEn	Guarda el catálogo actual. Guarda la lista actual de familias de máquinas (con sus máquinas) en un archivo.

Campos

catalogo	Almacena las familias de máquinas del catálogo
instanciaSMaquinas	Almacena su propia y única instancia

Vea también

[Espacio de nombres DraftLine](#)

ServicioMaquinas Constructor

Constructor privado. Sólo se genera la instancia desde la propia clase Inicializa el catálogo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub New
```

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.Instance Propiedad

Método (propiedad) para acceder a la clase

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Shared ReadOnly Property Instance As ServicioMaquinas  
    Get
```

Valor devuelto

[ServicioMaquinas](#)

la única instancia de la clase

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.ActualizaFamilia Método

Método para actualizar los atributos de una familia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub ActualizaFamilia (nombre As String, descripcion As String)
```

Parámetros

nombre [String](#) La familia a actualizar

descripcion [String](#) El Atributo Descripción a actualizar

ServicioMaquinas.ActualizaMaquina Método

Método que actualiza los atributos de una Maquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub ActualizaMaquina (ID As Guid, nombre As String, descripcion As String,  
                             familia As FamiliaMaquina, listaPropiedades As List(Of Single))
```

Parámetros

ID [Guid](#) El Identificador de la Maquina que se va a editar

nombre [String](#) El nuevo nombre de la Maquina

descripcion [String](#) La nueva descripción de la Maquina

familia [FamiliaMaquina](#) La familia a la que pertenece la Maquina

listaPropiedades [List\(Single\)](#) Los nuevos valores de los atributos de la Maquina

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.BorraCatalogo Método

Método para borrar el listado de familias Borra el catálogo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Sub BorraCatalogo

Observaciones

Atención - Se perderá todo!

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.CargaDesde Método

Lee el archivo de guardado de Familias de máquinas (catálogo) y los carga en la aplicación Devolviendo texto sobre el estado en que queda el catálogo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Function CargaDesde (*archivo* **As** String) **As** String

Parámetros

archivo [String](#) El archivo origen

Valor devuelto

[String](#) Texto a mostrar en estado catálogo

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.CompruebaNombre Método

Comprueba si alguna familia o máquina ya existe con un nombre dado

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function CompruebaNombre (nombre As String,  
    Optional familia As FamiliaMaquina = Nothing) As Object
```

Parámetros

nombre [String](#) Nombre a comprobar

familia [FamiliaMaquina](#) (Optional) La familia a la que pertenece si es una máquina a comprobar

Valor devuelto

[Object](#) True si ya existe, False si no se ha encontrado coincidencia

ServicioMaquinas.CreaFamilia Método

Método para crear Nuevas Familias de máquinas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub CreaFamilia (nombre As String,descripcion As String)
```

Parámetros

nombre [String](#) El nombre de la familia

descripcion [String](#) La descripción de la familia

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.CrearMaquina Método

Método para crear nuevas máquinas y añadirlas a la familia correspondiente

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Sub CrearMaquina (nombre As String, descripcion As String,
    familia As FamiliaMaquina, listaPropiedades As List(Of Single))
```

Parámetros

<i>nombre</i> String	El nombre de la máquina
<i>descripcion</i> String	La descripción de la máquina
<i>familia</i> FamiliaMaquina	La clase a la que añadir la nueva máquina
<i>listaPropiedades</i> List(Single)	Los valores de los atributos de la máquina

ServicioMaquinas.DelFamilia Método

Elimina la familia de máquinas de la lista (catálogo) que coincida con el nombre

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function DelFamilia (nombreFamilia As String) As Boolean
```

Parámetros

<i>nombreFamilia</i> String	El nombre de la familia a eliminar
---	------------------------------------

Valor devuelto

[Boolean](#)

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.DelMaquina Método

Elimina la máquina de la lista en la familia a la que pertenece

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function DelMaquina (nombre As String,familia As FamiliaMaquina) As Boolean
```

Parámetros

nombre [String](#) Nombre de la máquina a eliminar

familia [FamiliaMaquina](#) Familia de Maáquinas a la que pertenece

Valor devuelto

[Boolean](#)

ServicioMaquinas.GetAtributosMaquina Método

Función que buscará la máquina dada perteneciente a la familia dada y devolverá sus atributos de descripción, operadores, pot.Elect., Valor Coste y Valor Venta

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetAtributosMaquina (nombreFamilia As String,  
    nombreMaquina As String) As List(Of String)
```

Parámetros

nombreFamilia [String](#) El nombre de la familia a la que pertenece la máquina que se busca

nombreMaquina [String](#) El nombre de la familia dela que se buscan sus atributos

Valor devuelto

[List\(String\)](#) Lista de Strings conteniendo los atributos descripción, operadores, pot.Elect., Valor Coste y Valor Venta de la máquina

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.GetDatosFamilia Método

Devuelve un texto ya formateado con las propiedades de la familia indicada

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetDatosFamilia (nombre As String) As String
```

Parámetros

nombre [String](#) Nombre de la familia de la que se quieren las propiedades

Valor devuelto

[String](#) Texto formateado, con las propiedades de la familia indicada

ServicioMaquinas.GetDatosMaquina Método

Devuelve un texto ya formateado con las propiedades de la máquina indicada, perteneciente a la familia también indicada. Muestra o no el valor de coste según se indique

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetDatosMaquina (nombre As String,nombreFam As String,  
    valorCosteVista As Boolean) As String
```

Parámetros

nombre [String](#) Nombre de la máquina de la que se quieren las propiedades

nombreFam [String](#) Nombre de la familia a la que pertenece la máquina

valorCosteVista [Boolean](#) Booleano que indica si mostrar o no el valor de coste

Valor devuelto

[String](#) Texto formateado, con las propiedades de la máquina indicada

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.GetDescripcion Método

Devuelve el atributo descripción de la familia o máquina correspondiente al nombre pasado como argumento Si se incluye el argumento nombreMq, se refiere al atributo de una máquina

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetDescripcion (nombre As String,Optional nombreMq As String = Nothing) As String
```

Parámetros

nombre [String](#) Nombre del cual se requiere el atributo

nombreMq [String](#) (Optional) En caso de incluir éste parámetro, éste será la máquina del que se requiere el atributo y "nombre" será la familia al que pertenece

Valor devuelto

[String](#)

ServicioMaquinas.GetFamiliaID Método

Función que devuelve una familia, dado su ID

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetFamiliaID (identificador As Guid) As FamiliaMaquina
```

Parámetros

identificador [Guid](#) El ID que identifica a la familia

Valor devuelto

[FamiliaMaquina](#) la familia

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.GetFamilias Método

Función que devuelve la lista Familias de máquinas

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetFamilias As List(Of FamiliaMaquina)
```

Valor devuelto

[List\(FamiliaMaquina\)](#)

Lista de Familias de máquinas

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.GetIDFamilia Método

Función que devuelve el ID de una familia en particular

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetIDFamilia (nombreFamilia As String) As Guid
```

Parámetros

nombreFamilia [String](#) El nombre de la familia de la que se requiere el ID

Valor devuelto

[Guid](#) Devuelve el ID de la familia requerida

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.GetIDMaquina Método

Función que devuelve el ID de una máquina en particular

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetIDMaquina (nombreFamilia As String,nombreMaquina As String) As Guid
```

Parámetros

nombreFamilia [String](#) Nombre de la familia a la que pertenece la máquina

nombreMaquina [String](#) Nombre de la máquina

Valor devuelto

[Guid](#) Devuelve el ID de la máquina requerida

ServicioMaquinas.GetNombrePorID Método

Función que devuelve el nombre correspondiente de una familia, dado su ID

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Function GetNombrePorID (identificador As Guid) As String
```

Parámetros

identificador [Guid](#) El ID del que se requiere su nombre

Valor devuelto

[String](#) El nombre de la familia

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.GuardarEn Método

Guarda el catálogo actual. Guarda la lista actual de familias de máquinas (con sus máquinas) en un archivo.

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Function GuardarEn (archivo [As String](#)) [As Object](#)

Parámetros

archivo [String](#) El archivo destino

Valor devuelto

[Object](#)

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.catalogo Campo

Almacena las familias de máquinas del catálogo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private catalogo [As List\(Of FamiliaMaquina\)](#)

Valor del campo

[List \(FamiliaMaquina\)](#)

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.instanciaSMaquinas Campo

Almacena su propia y única instancia

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Private Shared](#) instanciaSMaquinas [As](#) ServicioMaquinas

Valor del campo

[ServicioMaquinas](#)

Vea también

[ServicioMaquinas Clase](#)

[Espacio de nombres DraftLine](#)

ServicioMaquinas.TipoElemento Enumeración

Tipo enumerado para que se pueda informar de qué tipo se trata, antes de crearlo incluso

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

[Public Enumeration](#) TipoElemento

Miembros

TipoFamilia	0	
TipoMaquina	1	

Vea también

[Espacio de nombres DraftLine](#)

VentanaPresentacion Clase

Ventana con información de la aplicación, que se muestra brevemente durante el inicio de la aplicación

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Public Class VentanaPresentacion
    Inherits Form
```

Inheritance [Object](#) → [MarshalByRefObject](#) → [Component](#) → [Control](#) → [ScrollableControl](#)
→ [ContainerControl](#) → [Form](#) → VentanaPresentacion

Constructores

VentanaPresentacion	Inicializa una nueva instancia de la clase VentanaPresentacion
-------------------------------------	--

Métodos

InitializeComponent	
PantallaPresentacion_Load	Carga la información a mostrar

Campos

components	
----------------------------	--

Vea también

[Espacio de nombres DraftLine](#)

VentanaPresentacion Constructor

Inicializa una nueva instancia de la clase [VentanaPresentacion](#)

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Public Sub New

Vea también

[VentanaPresentacion Clase](#)

[Espacio de nombres DraftLine](#)

VentanaPresentacion.InitializeComponent Método

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

Private Sub InitializeComponent

Vea también

[VentanaPresentacion Clase](#)

[Espacio de nombres DraftLine](#)

VentanaPresentacion.PantallaPresentacion_Load Método

Carga la información a mostrar

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private Sub PantallaPresentacion_Load (sender As Object, e As EventArgs)
```

Parámetros

sender [Object](#)

e [EventArgs](#)

Vea también

[VentanaPresentacion Clase](#)

[Espacio de nombres DraftLine](#)

VentanaPresentacion.components Campo

Definition

Espacio de nombres: [DraftLine](#)

Ensamblado: DraftLine (en DraftLine.exe) Versión: 1.1.0.0 (1.1.0.0)

VB

```
Private components As IContainer
```

Valor del campo

[IContainer](#)

Vea también

[VentanaPresentacion Clase](#)

[Espacio de nombres DraftLine](#)

ANEXO D CÓDIGO FUENTE PRUEBAS

D.1 Clase ServicioMaquinasPrueba

```
Imports System.IO
Imports System.ServiceModel
Imports DraftLine
Imports Microsoft.VisualStudio.TestTools.UnitTesting

<TestClass>
Public Class ServicioMaquinasPrueba

    ''' <summary>
    ''' Borro el catálogo de máquinas antes de cada prueba
    ''' </summary>
    <TestInitialize>
    Public Sub InicializaPrueba()
        'Inicializo la instancia de ServicioMaquinas para cada prueba borrando el catálogo
        ServicioMaquinas.Instance.BorraCatalogo()
    End Sub

    ''' <summary>
    ''' Compruebo que obtengo la instancia del servicioMaquinas y que puedo añadir una familia de máquinas
    ''' </summary>
    <TestMethod>
    Public Sub CreaFamilia_AñadeNuevaFamilia_ListaContieneFamilia()
        ' Arrange
        Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
        Dim nombre As String = "Familia 01"
        Dim descripcion As String = "Descripción de Familia 01"

        ' Act
        servicioMaquinas.CreaFamilia(nombre, descripcion)
        Dim familias As List(Of FamiliaMaquina) = servicioMaquinas.GetFamilias()

        ' Assert
        Assert.AreEqual(1, familias.Count) 'Que hay sólo una familia
        Assert.AreEqual(nombre, familias(0).Nombre) 'Que el nombre coincide
        Assert.AreEqual(descripcion, familias(0).Descripcion) 'Que la descripción coincide
    End Sub

    ''' <summary>
    ''' Verifico que obtengo la instancia del servicioMaquinas, que puedo añadir una familia de máquinas y
    ''' que puedo crear una nueva máquina y añadirla a esa familia
    ''' </summary>
    <TestMethod>
    Public Sub CrearMaquina_AñadeNuevaMaquinaAFamilia_FamiliaContieneMaquina()
        ' Arrange
        Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
        servicioMaquinas.CreaFamilia("Familia 01", "Descripción de Familia 01")
        Dim familiaMaquina As FamiliaMaquina = servicioMaquinas.GetFamilias()(0)
        Dim nombreMaquina As String = "Maquina 01"
        Dim descripcionMaquina As String = "Descripción de Maquina 01"
        Dim listaPropiedades As New List(Of Single)({1.0F, 2.0F, 3.0F, 4.0F, 5.0F})

        ' Act
        servicioMaquinas.CrearMaquina(nombreMaquina, descripcionMaquina, familiaMaquina, listaPropiedades)

        ' Assert
        Assert.AreEqual(1, familiaMaquina.Maquinas.Count) 'que la familia contiene una máquina
        Assert.AreEqual(nombreMaquina, familiaMaquina.Maquinas(0).Nombre) 'Que el nombre coincide
        Assert.AreEqual(descripcionMaquina, familiaMaquina.Maquinas(0).Descripcion) 'Que la descripción coincide
        Assert.AreEqual(listaPropiedades(0), familiaMaquina.Maquinas(0).Operarios) 'Que coincide la 1º propiedad
        Assert.AreEqual(listaPropiedades(1), familiaMaquina.Maquinas(0).PotenciaElectrica) 'Que coincide la 2º
prop
        Assert.AreEqual(listaPropiedades(2), familiaMaquina.Maquinas(0).Produccion) 'Que coincide la 3º
propiedad
        Assert.AreEqual(listaPropiedades(3), familiaMaquina.Maquinas(0).ValorCoste) 'Que coincide la 4º
propiedad
        Assert.AreEqual(listaPropiedades(4), familiaMaquina.Maquinas(0).ValorVenta) 'Que coincide la 5º
propiedad

    End Sub

    ''' <summary>
    ''' Compruebo que el catálogo creado se guarda en un archivo
    ''' </summary>
    <TestMethod>
```

```

Public Sub GuardarEn_GuardaCatalogoEnArchivo_ArchivoContieneDatosEsperados()
    ' Arrange
    Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
    servicioMaquinas.CreaFamilia("Familia 01", "Descripción de Familia 01")
    Dim catalogoPrueba As String = "catalogoPrueba.xml"
    If File.Exists(catalogoPrueba) Then File.Delete(catalogoPrueba)

    ' Act
    servicioMaquinas.GuardarEn(catalogoPrueba)

    ' Assert
    Assert.IsTrue(File.Exists(catalogoPrueba)) 'El archivo se ha creado y existe
    Dim contenido As String = File.ReadAllText(catalogoPrueba) 'Se puede leer su contenido
    Assert.IsTrue(contenido.Contains("Familia 01")) 'En el contenido está la familia añadida
    'CleanUp
    If File.Exists(catalogoPrueba) Then File.Delete(catalogoPrueba) 'Borro el archivo generado
End Sub

''' <summary>
''' Compruebo la carga de un catálogo desde un archivo al catálogo actual en uso
''' </summary>
<TestMethod>
Public Sub CargaDesde_CargaCatalogoDesdeArchivo_ListaContieneDatosEsperados()
    ' Arrange
    Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
    servicioMaquinas.CreaFamilia("Familia 01", "Descripción de Familia 01")
    Dim catalogoPrueba As String = "catalogoPrueba.xml"
    If File.Exists(catalogoPrueba) Then File.Delete(catalogoPrueba)
    servicioMaquinas.GuardarEn(catalogoPrueba)
    servicioMaquinas.BorraCatalogo() 'Borro el catálogo actual en uso

    ' Act
    servicioMaquinas.CargaDesde(catalogoPrueba)
    Dim familias As List(Of FamiliaMaquina) = servicioMaquinas.GetFamilias()

    ' Assert
    Assert.AreEqual(1, familias.Count) 'Que hay una familia
    Assert.AreEqual("Familia 01", familias(0).Nombre) 'Que el nombre coincide
    Assert.AreEqual("Descripción de Familia 01", familias(0).Descripcion) 'Que la descripción coincide
End Sub

''' <summary>
''' Compruebo que se puede eliminar una familia existente en el catálogo
''' </summary>
<TestMethod>
Public Sub DelFamilia_EliminaFamiliaExistente_ListaNoContieneFamilia()
    ' Arrange
    Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
    servicioMaquinas.CreaFamilia("Familia 01", "Descripción de Familia 01")

    ' Act
    Dim resultado As Boolean = servicioMaquinas.DelFamilia("Familia 01")
    Dim familias As List(Of FamiliaMaquina) = servicioMaquinas.GetFamilias()

    ' Assert
    Assert.IsTrue(resultado) 'Que el resultado es true
    Assert.AreEqual(0, familias.Count) 'Que no quedan familias en el catálogo
End Sub

''' <summary>
''' Compruebo que la verificación de que el nombre de una familia ya está usado funciona
''' </summary>
<TestMethod>
Public Sub CompruebaNombre_FamiliaExistente_DevuelveTrue()
    ' Arrange
    Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
    servicioMaquinas.CreaFamilia("Familia 01", "Descripción de Familia 01")

    ' Act
    Dim resultado As Boolean = servicioMaquinas.CompruebaNombre("Familia 01")
    Dim resultadoFalso As Boolean = servicioMaquinas.CompruebaNombre("OtroNombre")

    ' Assert
    Assert.IsTrue(resultado) 'Que ya existe ese nombre = true
    Assert.IsFalse(resultadoFalso) 'Que no existe ese nombre = false
End Sub

''' <summary>
''' Compruebo que se elimina correctamente la máquina añadida a una familia
''' </summary>

```

```

<TestMethod>
Public Sub DelMaquina_EliminaMaquina_FamiliaNoContieneMaquina()
    'Arrange
    Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
    servicioMaquinas.CreaFamilia("Familia 01", "Descripción de Familia 01")
    Dim familiaMaquina As FamiliaMaquina = servicioMaquinas.GetFamilias()(0)
    Dim nombreMaquina As String = "Maquina 01"
    Dim descripcionMaquina As String = "Descripción de Maquina 01"
    Dim listaPropiedades As New List(Of Single){1.0F, 2.0F, 3.0F, 4.0F, 5.0F}
    servicioMaquinas.CrearMaquina(nombreMaquina, descripcionMaquina, familiaMaquina, listaPropiedades)

    'Act
    servicioMaquinas.DelMaquina(nombreMaquina, familiaMaquina)

    'Assert
    Assert.AreEqual(0, familiaMaquina.Maquinas.Count) 'Que la máquina se ha eliminado

End Sub

''' <summary>
''' Compruebo que los datos recuperados de una familia de máquinas son los esperados
''' </summary>
<TestMethod>
Public Sub GetDatosFamilia_ObtengoDatosFamilia_DatosCoinciden()
    'Arrange
    Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
    Dim nombreFamilia = "Familia 01"
    Dim descripcionFamilia = "Descripción de Familia 01"
    servicioMaquinas.CreaFamilia(nombreFamilia, descripcionFamilia)
    Dim textoARecibir = "NOMBRE: " & nombreFamilia & vbCrLf & "DESCRIPCIÓN: " & descripcionFamilia

    'act
    Dim textoRecibido = servicioMaquinas.GetDatosFamilia(nombreFamilia)

    'Assert
    Assert.AreEqual(textoARecibir, textoRecibido) 'Los textos coinciden

End Sub

''' <summary>
''' Compruebo que los datos recuperados de una máquina son los esperados
''' </summary>
<TestMethod>
Public Sub GetDatosMaquina_ObtengoDatosMaquina_DatosCoinciden()
    'Arrange
    Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
    servicioMaquinas.CreaFamilia("Familia 01", "Descripción de Familia 01")
    Dim familiaMaquina As FamiliaMaquina = servicioMaquinas.GetFamilias()(0)
    Dim nombreMaquina As String = "Maquina 01"
    Dim descripcionMaquina As String = "Descripción de Maquina 01"
    Dim listaPropiedades As New List(Of Single){1.0F, 2.0F, 3.0F, 4.0F, 5.0F}
    servicioMaquinas.CrearMaquina(nombreMaquina, descripcionMaquina, familiaMaquina, listaPropiedades)
    Dim textoArecibirConCoste = "NOMBRE:".PadRight(20) & nombreMaquina & vbCrLf & vbCrLf & vbTab &
        "DESCRIPCIÓN:".PadRight(20) & descripcionMaquina & vbCrLf & vbCrLf & vbTab &
        "PRODUCCIÓN:".PadRight(20) & $"{vbTab & listaPropiedades(2)} Kg/h" & vbCrLf & vbTab &
        "POT. ELÉCTRICA:".PadRight(20) & $"{vbTab & listaPropiedades(1)} Kw" & vbCrLf & vbTab &
        "OPERARIOS:".PadRight(20) & $"{vbTab & listaPropiedades(0)}" & vbCrLf & vbTab &
        "PRECIO COSTE:".PadRight(20) & $"{vbTab & listaPropiedades(3)} €" & vbCrLf & vbTab &
        "PRECIO VENTA:".PadRight(20) & $"{vbTab & listaPropiedades(4)} €" & vbCrLf & vbTab
    Dim textoArecibirSinCoste = "NOMBRE:".PadRight(20) & nombreMaquina & vbCrLf & vbCrLf & vbTab &
        "DESCRIPCIÓN:".PadRight(20) & descripcionMaquina & vbCrLf & vbCrLf & vbTab &
        "PRODUCCIÓN:".PadRight(20) & $"{vbTab & listaPropiedades(2)} Kg/h" & vbCrLf & vbTab &
        "POT. ELÉCTRICA:".PadRight(20) & $"{vbTab & listaPropiedades(1)} Kw" & vbCrLf & vbTab &
        "OPERARIOS:".PadRight(20) & $"{vbTab & listaPropiedades(0)}" & vbCrLf & vbTab &
        "PRECIO VENTA:".PadRight(20) & $"{vbTab & listaPropiedades(4)} €" & vbCrLf & vbTab

    'Act
    Dim textoRecibidoConCoste = servicioMaquinas.GetDatosMaquina(nombreMaquina, familiaMaquina.Nombre, True)
    Dim textoRecibidoSinCoste = servicioMaquinas.GetDatosMaquina(nombreMaquina, familiaMaquina.Nombre,
False)

    'Assert
    Assert.AreEqual(textoArecibirConCoste, textoRecibidoConCoste) 'Los textos coinciden
    Assert.AreEqual(textoArecibirSinCoste, textoRecibidoSinCoste) 'Los textos coinciden

End Sub

''' <summary>
''' Compruebo que se devuelve la descripción de una familia y de una máquina, correctamente
''' </summary>
<TestMethod>
Public Sub GetDescripcion_ObtengoDescripcion_DescripcionCoincide()

```

```

'Arrange
Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
Dim nombreFamilia = "Familia 01"
Dim descripcionFamilia = "Descripción de Familia 01"
servicioMaquinas.CreaFamilia(nombreFamilia, descripcionFamilia)
Dim familiaMaquina As FamiliaMaquina = servicioMaquinas.GetFamilias()(0)
Dim nombreMaquina As String = "Maquina 01"
Dim descripcionMaquina As String = "Descripción de Maquina 01"
Dim listaPropiedades As New List(Of Single)({1.0F, 2.0F, 3.0F, 4.0F, 5.0F})
servicioMaquinas.CrearMaquina(nombreMaquina, descripcionMaquina, familiaMaquina, listaPropiedades)
Dim textoARecibirFamilia = descripcionFamilia
Dim textoARecibirMaquina = descripcionMaquina

'Act
Dim textoRecibidoFamilia = servicioMaquinas.GetDescripcion(nombreFamilia)
Dim textoRecibidoMaquina = servicioMaquinas.GetDescripcion(nombreFamilia, nombreMaquina)

'Assert
Assert.AreEqual(textoARecibirFamilia, textoRecibidoFamilia) 'Que coinciden las descripciones
Assert.AreEqual(textoARecibirMaquina, textoRecibidoMaquina) 'Que coinciden las descripciones

End Sub

''' <summary>
''' Compruebo que se cambia la descripción de una familia
''' </summary>
<TestMethod>
Public Sub ActualizaFamilia_CambiaDescripcion_DescripcionCoincide()
'Arrange
Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
Dim nombreFamilia = "Familia 01"
Dim descripcionFamilia = "Descripción de Familia 01"
servicioMaquinas.CreaFamilia(nombreFamilia, descripcionFamilia)
Dim nuevaDescripcion = "Nueva Descripción de Familia 01"

'Act
servicioMaquinas.ActualizaFamilia(nombreFamilia, nuevaDescripcion)

'Assert
Assert.AreEqual(nuevaDescripcion, servicioMaquinas.GetDescripcion(nombreFamilia)) 'La nueva descripción coincide

End Sub

''' <summary>
''' Compruebo que se obtiene correctamente el nombre de una familia
''' </summary>
<TestMethod>
Public Sub GetNombrePorID_ObtengoNombreDesdeID_NombreCoincide()
'Arrange
Dim servicioMaquinas As ServicioMaquinas = ServicioMaquinas.Instance
Dim nombreFamilia = "Familia 01"
Dim descripcionFamilia = "Descripción de Familia 01"
servicioMaquinas.CreaFamilia(nombreFamilia, descripcionFamilia)
Dim identificador = servicioMaquinas.GetIDFamilia(nombreFamilia)

'Act
Dim nombreObtenido = servicioMaquinas.GetNombrePorID(identificador)

'Assert
Assert.AreEqual(nombreFamilia, nombreObtenido)
End Sub

End Class

```

D.2 Clase ServicioLineasPrueba

```

Imports System.IO
Imports DraftLine

<TestClass(> Public Class ServicioLineasPrueba
Private servicioLineas As ServicioLineas

''' <summary>
''' Compruebo que se obtiene la instancia y reinicio antes de cada prueba

```

```

''' </summary>
<TestInitialize>
Public Sub InicializaPrueba()
    servicioLineas = ServicioLineas.Instance
    servicioLineas.NuevaLinea() ' Reinicia la línea antes de cada prueba
End Sub

''' <summary>
''' Compruebo que se crea una nueva línea
''' </summary>
<TestMethod>
Public Sub NuevaLinea_CreaUnaNuevaLinea_LineaAsignada()
    'Arrange 'Act
    servicioLineas.NuevaLinea()
    'Assert
    Assert.IsNotNull(servicioLineas.Linea)
End Sub

''' <summary>
''' Compruebo que se guarda la línea
''' </summary>
<TestMethod>
Public Sub GuardarEn_GuardaLaLineaCorrectamente()
    'Arrange
    Dim pathLinea As String = "línea.xml"
    servicioLineas.Linea.Nombre = "LíneaPrueba"
    'Act
    servicioLineas.GuardarEn(pathLinea)
    'Assert
    Assert.IsTrue(File.Exists(pathLinea)) 'Existe el archivo que se ha creado con la línea
    Dim contenido As String = File.ReadAllText(pathLinea) 'Se puede leer su contenido
    Assert.IsTrue(contenido.Contains("LíneaPrueba")) 'En el contenido está el nombre de la línea

    ' Cleanup
    If File.Exists(pathLinea) Then File.Delete(pathLinea) 'Borro el archivo creado

End Sub

''' <summary>
''' Compruebo la carga de una línea desde un archivo
''' </summary>
<TestMethod>
Public Sub CargarDesde_CargaLaLineaCorrectamente()
    Dim pathLinea As String = "línea.xml"
    servicioLineas.Linea.Nombre = "Línea de prueba"
    If File.Exists(pathLinea) Then File.Delete(pathLinea)
    servicioLineas.GuardarEn(pathLinea)
    'Act
    servicioLineas.CargarDesde(pathLinea)
    servicioLineas.AceptaLinea()
    'Assert
    Assert.AreEqual("Línea de prueba", servicioLineas.Linea.Nombre)
    'Cleanup
    If File.Exists(pathLinea) Then File.Delete(pathLinea) 'Borro el archivo creado

End Sub
End Class

```

D.3 Clase FamiliaMaquinaPrueba

```

Imports DraftLine
Imports Microsoft.VisualStudio.TestTools.UnitTesting

<TestClass()> Public Class FamiliaMaquinaPrueba

    ''' <summary>
    ''' Se comprueba el constructor por defecto (utilizado para serializar)
    ''' Debe crear los atributos por defecto
    ''' </summary>
    <TestMethod>
    Public Sub Constructor_SinParametros_AtributosPorDefecto()

```

```

'Arrange (utilizo nomenclatura de Microsoft.Learn. Arrange para la "disposición", Act para las ejecuciones
y Assert para las comprobaciones)
Dim familia As New FamiliaMaquina()
' Assert
Assert.AreEqual(Guid.Empty, familia.ID) 'El ID declarado pero vacío
Assert.IsNull(familia.Nombre) 'El nombre delcarado pero vacío
Assert.IsNull(familia.Descripcion) 'La descripción declarada pero vacía
Assert.IsNull(familia.Maquinas) 'Su lista de máquinas no se ha inicializado
End Sub

<TestMethod()> Public Sub Constructor_ConParametros_AtributosEnviados()
' Arrange
Dim nombre As String = "Familia 01"
Dim descripcion As String = "Descripción de la familia 01"
' Act
Dim familia As New FamiliaMaquina(nombre, descripcion)
' Assert
Assert.IsNotNull(familia.ID) 'Compruebo que ID ya no es null
Assert.AreEqual(nombre, familia.Nombre) 'Compruebo que coincide el nombre
Assert.AreEqual(descripcion, familia.Descripcion) 'Coincide la descripción
Assert.AreEqual(0, familia.Maquinas.Count) 'Su lista de máquinas aún no contiene ninguna
End Sub

<TestMethod()> Public Sub TestAddMaquina()
' Arrange
Dim familia As New FamiliaMaquina("Familia 01", "Descripción de la familia 01")
Dim maquina As New Maquina("Máquina de prueba", "Descripción de la máquina de prueba")
' Act
familia.AddMaquina(maquina)
' Assert
Assert.AreEqual(1, familia.Maquinas.Count) 'Hay una máquina
Assert.AreEqual(maquina, familia.Maquinas(0)) 'coincide la máquina
End Sub

<TestMethod()> Public Sub TestToString()
' Arrange
Dim nombre As String = "Familia 01"
Dim descripcion As String = "Descripción de la familia 01"
Dim familia As New FamiliaMaquina(nombre, descripcion)
' Act
Dim resultado As String = familia.ToString()
' Assert
Assert.AreEqual(nombre, resultado) 'Concide el nombre devuelto
End Sub
End Class

```

D.4 Clase MaquinaPrueba

```

Imports DraftLine

<TestClass>
Public Class MaquinaPrueba

''' <summary>
''' Se comprueba el constructor por defecto (utilizado para serializar)
''' Debe crear los atributos por defecto
''' </summary>
<TestMethod>
Public Sub Constructor_SinParametros_AtributosPorDefecto()
'Arrange (utilizo nomenclatura de Microsoft.Learn. Arrange para la "disposición", Act para las ejecuciones
y Assert para las comprobaciones)
Dim maquina As New Maquina()
' Assert
Assert.AreEqual(Guid.Empty, maquina.ID) 'El ID declarado pero vacío
Assert.IsNull(maquina.Nombre) 'El nombre delcarado pero vacío
Assert.IsNull(maquina.Descripcion) 'La descripción declarada pero vacía
Assert.AreEqual(0, maquina.Produccion) 'Estos son Single, se inicializan por defecto a 0 al declararlos
Assert.AreEqual(0, maquina.Operarios)
Assert.AreEqual(0, maquina.PotenciaElectrica)
Assert.AreEqual(0, maquina.ValorCoste)
Assert.AreEqual(0, maquina.ValorVenta)
End Sub

<TestMethod>
Public Sub Constructor_ConParametros_AtributoEnviados()

```

```

' Arrange
Dim nombre As String = "Máquina 01"
Dim descripcion As String = "Descripción de Máquina 01"
' Act
Dim maquina As New Maquina(nombre, descripcion)
' Assert
Assert.AreEqual(Guid.Empty, maquina.ID) 'Compruebo que ID ya no es null
Assert.AreEqual(nombre, maquina.Nombre) 'Compruebo que coincide el nombre
Assert.AreEqual(descripcion, maquina.Descripcion) 'Coincide la descripción
Assert.AreEqual(0, maquina.Produccion) 'Estos son Single, se inicializan por defecto a 0 al declararlos
Assert.AreEqual(0, maquina.Operarios)
Assert.AreEqual(0, maquina.PotenciaElectrica)
Assert.AreEqual(0, maquina.ValorCoste)
Assert.AreEqual(0, maquina.ValorVenta)
End Sub

<TestMethod>
Public Sub AsignaPropiedades_AtributosEnviados()
' Arrange
Dim maquina As New Maquina()
Dim listaPropiedades As New List(Of Single) From {5, 10, 20, 30, 40}
' Act
maquina.AsignaPropiedades(listaPropiedades)
' Assert
Assert.AreEqual(5, maquina.Operarios) 'Compruebo coinciden los singles
Assert.AreEqual(10, maquina.PotenciaElectrica)
Assert.AreEqual(20, maquina.Produccion)
Assert.AreEqual(30, maquina.ValorCoste)
Assert.AreEqual(40, maquina.ValorVenta)
End Sub

<TestMethod>
Public Sub ToString_TextoCorrecto()
' Arrange
Dim maquina As New Maquina("Máquina 01", "Descripción de Máquina 01") With {
    .Produccion = 100,
    .Operarios = 5,
    .PotenciaElectrica = 50,
    .ValorCoste = 5000,
    .ValorVenta = 10000
}
Dim expected As String = "NOMBRE:".PadRight(20) & "Máquina 01" & vbCrLf & vbCrLf & vbTab &
    "DESCRIPCIÓN:".PadRight(20) & "Descripción de Máquina 01" & vbCrLf & vbCrLf &
vbTab &
    "PRODUCCIÓN:".PadRight(20) & $"{vbTab & 100} Kg/h" & vbCrLf & vbTab &
    "POT. ELÉCTRICA:".PadRight(20) & $"{vbTab & 50} Kw" & vbCrLf & vbTab &
    "OPERARIOS:".PadRight(20) & $"{vbTab & 5}" & vbCrLf & vbTab &
    "PRECIO COSTE:".PadRight(20) & $"{vbTab & 5000} €" & vbCrLf & vbTab &
    "PRECIO VENTA:".PadRight(20) & $"{vbTab & 10000} €" & vbCrLf & vbTab

' Act
Dim result As String = maquina.ToString()
' Assert
Assert.AreEqual(expected, result)
End Sub

<TestMethod>
Public Sub ToStringNoCoste_TextoCorrecto()
' Arrange
Dim maquina As New Maquina("Máquina 01", "Descripción de Máquina 01") With {
    .Produccion = 100,
    .Operarios = 5,
    .PotenciaElectrica = 50,
    .ValorVenta = 10000
}
Dim expected As String = "NOMBRE:".PadRight(20) & "Máquina 01" & vbCrLf & vbCrLf & vbTab &
    "DESCRIPCIÓN:".PadRight(20) & "Descripción de Máquina 01" & vbCrLf & vbCrLf &
vbTab &
    "PRODUCCIÓN:".PadRight(20) & $"{vbTab & 100} Kg/h" & vbCrLf & vbTab &
    "POT. ELÉCTRICA:".PadRight(20) & $"{vbTab & 50} Kw" & vbCrLf & vbTab &
    "OPERARIOS:".PadRight(20) & $"{vbTab & 5}" & vbCrLf & vbTab &
    "PRECIO VENTA:".PadRight(20) & $"{vbTab & 10000} €" & vbCrLf & vbTab

' Act
Dim result As String = maquina.ToStringNoCoste()
' Assert
Assert.AreEqual(expected, result)
End Sub
End Class

```

D.5 Clase LineaPrueba

```
Imports DraftLine
Imports Microsoft.VisualStudio.TestTools.UnitTesting

<TestClass()> Public Class LineaPrueba

    ''' <summary>
    ''' Se comprueba el constructor por defecto (utilizado para serializar)
    ''' Debe crear los atributos por defecto
    ''' </summary>
    <TestMethod>
    Public Sub Constructor_SinParametros_AtributosPorDefecto()
        'Arrange
        Dim linea As New Linea()
        'Assert
        Assert.AreEqual(linea.Nombre, "Línea de producción genérica")
        Assert.AreEqual(linea.Cliente, "Cliente sin definir")
        Assert.AreEqual(linea.Proyecto, "Proyecto genérico")
        Assert.AreEqual(linea.Version, "1.0")
        Assert.AreEqual(linea.Descripcion, "No se ha indicado descripción de la línea")

    End Sub
End Class
```

D.6 Clase CondicionesPrueba

```
Imports DraftLine
Imports Microsoft.VisualStudio.TestTools.UnitTesting

<TestClass()> Public Class CondicionesPrueba

    ''' <summary>
    ''' Compruebo que en el constructor por nombre se crean los atributos por defecto
    ''' </summary>
    <TestMethod>
    Public Sub ConstructorConNombre_AtributosPorDefecto()
        'Arrange
        Dim condiciones As New Condiciones("nuevo")
        'Assert
        Assert.AreEqual(1, condiciones.Paralelismo) 'Se inicializa a 1
        Assert.AreEqual(0, condiciones.ProduccionMin) 'Se inicializa a 0
        Assert.AreEqual(Nothing, condiciones.MinimizarPrimaria) 'Se inicializa a Nothing
        Assert.AreEqual(Nothing, condiciones.MinimizarSecundaria) 'Se inicializa a Nohting

    End Sub
End Class
```

ANEXO E DOCUMENTACIÓN DE CÓDIGO PRUEBAS

Se muestra a continuación la documentación completa del código de pruebas, generada con la herramienta [SandCastle Help File Builder](#). Se omiten explicaciones de métodos heredados que no aportan información sobre el trabajo realizado por el autor.

Pruebas01 Espacio de nombres

Clases

CondicionesPrueba	
FamiliaMaquinaPrueba	
LineaPrueba	
MaquinaPrueba	
ServicioLineasPrueba	
ServicioMaquinasPrueba	

CondicionesPrueba Clase

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Class CondicionesPrueba

Inheritance [Object](#) → CondicionesPrueba

Constructores

[CondicionesPrueba](#)

Inicializa una nueva instancia de la clase [CondicionesPrueba](#)

Métodos

[ConstructorConNombre AtributosPorDefecto](#)

Compruebo que en el constructor por nombre se crean los atributos por defecto

Vea también

[Espacio de nombres Pruebas01](#)

CondicionesPrueba Constructor

Inicializa una nueva instancia de la clase [CondicionesPrueba](#)

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub New

Vea también

[CondicionesPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

CondicionesPrueba.ConstructorConNombre_AtributosPorDefecto Método

Compruebo que en el constructor por nombre se crean los atributos por defecto

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub ConstructorConNombre_AtributosPorDefecto

Vea también

[CondicionesPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

FamiliaMaquinaPrueba Clase

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Class FamiliaMaquinaPrueba

Inheritance [Object](#) → FamiliaMaquinaPrueba

Constructores

[FamiliaMaquinaPrueba](#)

Inicializa una nueva instancia de la clase FamiliaMaquinaPrueba

Métodos

[Constructor ConParametros AtributosEnviados](#)

[Constructor SinParametros AtributosPorDefecto](#)

Se comprueba el constructor por defecto (utilizado para serializar) Debe crear los atributos por defecto

[TestAddMaquina](#)

[TestToString](#)

Vea también

[Espacio de nombres Pruebas01](#)

FamiliaMaquinaPrueba Constructor

Inicializa una nueva instancia de la clase [FamiliaMaquinaPrueba](#)

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

[Public Sub](#) New

Vea también

[FamiliaMaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

FamiliaMaquinaPrueba.Constructor_ConParametros_AtributosEnviados Método

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

[Public Sub](#) Constructor_ConParametros_AtributosEnviados

Vea también

[FamiliaMaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

FamiliaMaquinaPrueba.Constructor_SinParametros_AtributosPorDefecto Método

Se comprueba el constructor por defecto (utilizado para serializar) Debe crear los atributos por defecto

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub Constructor_SinParametros_AtributosPorDefecto

Vea también

[FamiliaMaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

FamiliaMaquinaPrueba.TestAddMaquina Método

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub TestAddMaquina

Vea también

[FamiliaMaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

FamiliaMaquinaPrueba.TestToString Método

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub TestToString

Vea también

[FamiliaMaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

LineaPrueba Clase

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

```
Public Class LineaPrueba
```

Inheritance [Object](#) → [LineaPrueba](#)

Constructores

LineaPrueba	Inicializa una nueva instancia de la clase LineaPrueba
-----------------------------	--

Métodos

Constructor SinParametros AtributosPorDefecto	Se comprueba el constructor por defecto (utilizado para serializar) Debe crear los atributos por defecto
---	--

Vea también

[Espacio de nombres Pruebas01](#)

LineaPrueba Constructor

Inicializa una nueva instancia de la clase [LineaPrueba](#)

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub New

Vea también

[LineaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

LineaPrueba.Constructor_SinParametros_AtributosPorDefecto Método

Se comprueba el constructor por defecto (utilizado para serializar) Debe crear los atributos por defecto

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub Constructor_SinParametros_AtributosPorDefecto

Vea también

[LineaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

MaquinaPrueba Clase

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Class MaquinaPrueba

Inheritance [Object](#) → MaquinaPrueba

Constructores

MaquinaPrueba	Inicializa una nueva instancia de la clase MaquinaPrueba
-------------------------------	--

Métodos

AsignaPropiedades AtributosEnviados	
Constructor ConParametros AtributoEnviados	
Constructor SinParametros AtributosPorDefecto	Se comprueba el constructor por defecto (utilizado para serializar) Debe crear los atributos por defecto
ToString TextoCorrecto	
ToStringNoCoste TextoCorrecto	

Vea también

[Espacio de nombres Pruebas01](#)

MaquinaPrueba Constructor

Inicializa una nueva instancia de la clase [MaquinaPrueba](#)

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

[Public Sub](#) New

Vea también

[MaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

MaquinaPrueba.AsignaPropiedades_AtributosEnviados Método

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

[Public Sub](#) AsignaPropiedades_AtributosEnviados

Vea también

[MaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

MaquinaPrueba.Constructor_ConParametros_AtributoEnviados Método

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub Constructor_ConParametros_AtributoEnviados

Vea también

[MaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

MaquinaPrueba.Constructor_SinParametros_AtributosPorDefecto Método

Se comprueba el constructor por defecto (utilizado para serializar) Debe crear los atributos por defecto

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub Constructor_SinParametros_AtributosPorDefecto

Vea también

[MaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

MaquinaPrueba.ToString_TextoCorrecto Método

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

```
Public Sub ToString_TextoCorrecto
```

Vea también

[MaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

MaquinaPrueba.ToStringNoCoste_TextoCorrecto Método

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

```
Public Sub ToStringNoCoste_TextoCorrecto
```

Vea también

[MaquinaPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioLineasPrueba Clase

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Class ServicioLineasPrueba

Inheritance [Object](#) → [ServicioLineasPrueba](#)

Constructores

ServicioLineasPrueba	Inicializa una nueva instancia de la clase ServicioLineasPrueba
--------------------------------------	---

Métodos

CargarDesde_CargaLaLineaCorrectamente	Compruebo la carga de una linea desde un archivo
GuardarEn_GuardaLaLineaCorrectamente	Compruebo que se guarda la linea
InicializaPrueba	Compruebo que se obtiene la instancia y reinicio antes de cada prueba
NuevaLinea_CreaUnaNuevaLinea_LineaAsignada	Compruebo que se crea una nueva linea

Campos

servicioLineas	
--------------------------------	--

Vea también

[Espacio de nombres Pruebas01](#)

ServicioLineasPrueba Constructor

Inicializa una nueva instancia de la clase [ServicioLineasPrueba](#)

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub New

Vea también

[ServicioLineasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioLineasPrueba.CargarDesde_CargaLaLineaCorrectamente Método

Compruebo la carga de una linea desde un archivo

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub CargarDesde_CargaLaLineaCorrectamente

Vea también

[ServicioLineasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioLineasPrueba.GuardarEn_GuardaLaLineaCorrectamente Método

Compruebo que se guarda la linea

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub GuardarEn_GuardaLaLineaCorrectamente

Vea también

[ServicioLineasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioLineasPrueba.InicializaPrueba Método

Compruebo que se obtiene la instancia y reinicio antes de cada prueba

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub InicializaPrueba

Vea también

[ServicioLineasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioLineasPrueba.NuevaLinea_CreaUnaNuevaLinea_LineaAsignada Método

Compruebo que se crea una nueva linea

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub NuevaLinea_CreaUnaNuevaLinea_LineaAsignada

Vea también

[ServicioLineasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioLineasPrueba.servicioLineas Campo

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Private servicioLineas **As** ServicioLineas

Valor del campo

ServicioLineas

Vea también

[ServicioLineasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba Clase

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Class ServicioMaquinasPrueba

Inheritance [Object](#) → ServicioMaquinasPrueba

Constructores

ServicioMaquinasPrueba	Inicializa una nueva instancia de la clase ServicioMaquinasPrueba
--	---

Métodos

ActualizarFamilia CambiaDescripcion DescripcionCoincide	Compruebo que se cambia la descripción de una familia
CargaDesde CargaCatalogoDesdeArchivo ListaContieneDatosEsperados	Compruebo la carga de un catálogo desde un archivo al catálogo actual en uso
CompruebaNombre FamiliaExistente DevuelveTrue	Compruebo que la verificación de que el nombre de una familia ya está usado funciona
CreaFamilia AñadeNuevaFamilia ListaContieneFamilia	Compruebo que obtengo la instancia del servicioMaquinas y que puedo añadir una familia de máquinas
CrearMaquina AñadeNuevaMaquinaAFamilia FamiliaContieneMaquina	Verifico que obtengo la instancia del servicioMaquinas, que puedo añadir una familia de máquinas y que puedo crear

	una nueva máquina y añadirla a esa familia
<u>DelFamilia EliminaFamiliaExistente ListaNoContieneFamilia</u>	Compruebo que se puede eliminar una familia existente en el catálogo
<u>DelMaquina EliminaMaquina FamiliaNoContieneMaquina</u>	Compruebo que se elimina correctamente la máquina añadida a una familia
<u>GetDatosFamilia ObtengoDatosFamilia DatosCoinciden</u>	Compruebo que los datos recuperados de una familia de máquinas son los esperados
<u>GetDatosMaquina ObtengoDatosMaquina DatosCoinciden</u>	Compruebo que los datos recuperados de una máquina son los esperados
<u>GetDescripcion ObtengoDescripcion DescripcionCoincide</u>	Compruebo que se devuelve la descripción de una familia y de una máquina, correctamente
<u>GetNombrePorID ObtengoNombreDesdeID NombreCoincide</u>	Compruebo que se obtiene correctamente el nombre de una familia
<u>GuardarEn GuardaCatalogoEnArchivo ArchivoContieneDatosEsperados</u>	Compruebo que el catálogo creado se guarda en un archivo
<u>InicializaPrueba</u>	Borro el catálogo de máquinas antes de cada prueba

Vea también

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba Constructor

Inicializa una nueva instancia de la clase [ServicioMaquinasPrueba](#)

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

[Public Sub](#) New

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.ActualizaFamilia_CambiaDescripcion_DescripcionCoincide Método

Compruebo que se cambia la descripción de una familia

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

[Public Sub](#) ActualizaFamilia_CambiaDescripcion_DescripcionCoincide

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.CargaDesde_CargaCatalogoDesdeArchivo_ListaContieneDatosEsperados Método

Compruebo la carga de un catálogo desde un archivo al catálogo actual en uso

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub CargaDesde_CargaCatalogoDesdeArchivo_ListaContieneDatosEsperados

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.CompruebaNombre_FamiliaExistente_DevuelveTrue Método

Compruebo que la verificación de que el nombre de una familia ya está usado funciona

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub CompruebaNombre_FamiliaExistente_DevuelveTrue

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.CreaFamilia_AñadeNuevaFamilia_ListaContieneFamilia Método

Compruebo que obtengo la instancia del servicioMaquinas y que puedo añadir una familia de máquinas

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub CreaFamilia_AñadeNuevaFamilia_ListaContieneFamilia

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.CrearMaquina_AñadeNuevaMaquinaAFamilia_FamiliaContieneMaquina Método

Verifico que obtengo la instancia del servicioMaquinas, que puedo añadir una familia de máquinas y que puedo crear una nueva máquina y añadirla a esa familia

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub CrearMaquina_AñadeNuevaMaquinaAFamilia_FamiliaContieneMaquina

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.DelFamilia_EliminaFamiliaExistente_ListaNoContieneFamilia Método

Compruebo que se puede eliminar una familia existente en el catálogo

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub DelFamilia_EliminaFamiliaExistente_ListaNoContieneFamilia

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.DelMaquina_EliminaMaquina_FamiliaNoContieneMaquina Método

Compruebo que se elimina correctamente la máquina añadida a una familia

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub DelMaquina_EliminaMaquina_FamiliaNoContieneMaquina

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.GetDatosFamilia_ObtengoDatosFamilia_DatosCoinciden Método

Compruebo que los datos recuperados de una familia de máquinas son los esperados

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub GetDatosFamilia_ObtengoDatosFamilia_DatosCoinciden

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.GetDatosMaquina_ObtengoDatosMaquina_DatosCoinciden Método

Compruebo que los datos recuperados de una máquina son los esperados

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub GetDatosMaquina_ObtengoDatosMaquina_DatosCoinciden

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.GetDescripcion_ObtengoDescripcion_DescripcionCoincide Método

Compruebo que se devuelve la descripción de una familia y de una máquina, correctamente

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub GetDescripcion_ObtengoDescripcion_DescripcionCoincide

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.GetNombrePorID_ObtengoNombreDesdeID_NombreCoincide Método

Compruebo que se obtiene correctamente el nombre de una familia

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub GetNombrePorID_ObtengoNombreDesdeID_NombreCoincide

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.GuardarEn_GuardaCatalogoEnArchivo_ArchivoContieneDatosEsperados Método

Compruebo que el catálogo creado se guarda en un archivo

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub GuardarEn_GuardaCatalogoEnArchivo_ArchivoContieneDatosEsperados

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)

ServicioMaquinasPrueba.InicializaPrueba Método

Borro el catálogo de máquinas antes de cada prueba

Definition

Espacio de nombres: [Pruebas01](#)

Ensamblado: Pruebas01 (en Pruebas01.dll) Versión: 1.0.0.0 (1.0.0.0)

VB

Public Sub InicializaPrueba

Vea también

[ServicioMaquinasPrueba Clase](#)

[Espacio de nombres Pruebas01](#)