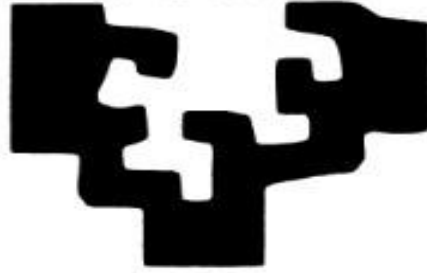


eman ta zabal zazu



universidad
del país vasco

euskal herriko
unibertsitatea

**Facultad de
Informática**

Informatika Fakultatea

Titulación: Ingeniería Informática

Gestión avanzada de listas de materiales

Alumno: Mikel Villoslada Echávarri

Director: José Javier Dolado Cosín

Tutores por parte de la empresa: Mónica San Sebastian y Fernando Barral

Proyecto Fin de Carrera, julio 2013

Agradecimientos

Quisiera agradecer a la Universidad del País Vasco y a Ibermática S.A. la oportunidad de haber realizado este proyecto.

En lo personal, agradecer a Javier Dolado por su orientación y ayuda; también al personal Ibermática Soluciones Empresariales de la oficina de Zuatzu por su amabilidad y apoyo, especialmente a Mónica San Sebastián, Fernando Barral, Jon Zabaleta e Iñigo Curto, entre otros muchos trabajadores que alguna vez me han ayudado.

Resumen

Este documento contiene una memoria del Proyecto Fin de Carrera de la titulación de Ingeniería Informática, realizado por Mikel Villoslada Echávarri en la empresa de consultoría y servicios informáticos Ibermática, bajo la supervisión de Mónica San Sebastián Arias (tutora por parte de la empresa) y José Javier Dolado Cosín (director del proyecto por parte de la Universidad del País Vasco).

El proyecto consiste en una funcionalidad para una aplicación de integración de datos entre programas de diseño asistido por ordenador (CAD) y sistemas de gestión de recursos de empresas (ERP). Se desea expandir la funcionalidad de envío de listas de materiales existente con nuevos casos de uso, rediseñando una nueva interfaz gráfica y modificando aspectos de la aplicación, para lograr una gestión avanzada de listas de materiales.

Adicionalmente, se contempla desarrollar nuevas funcionalidades para la integración como parte del ciclo de desarrollo del producto.

Índice

1.	Introducción.....	8
2.	Documento de objetivos del proyecto	9
2.1	Alcance	9
	Gestión avanzada de listas de materiales.....	9
2.2	Estructura de desglose de trabajo	10
2.3	Roles (empresa)	11
2.4	Metodología de trabajo	12
2.4.1	Ciclo de vida	12
2.4.2	Herramientas para el trabajo en equipo.....	13
2.5	Planificación del tiempo.....	14
2.6	Planificación de riesgos.....	16
2.6.1	Identificación de riesgos según origen	16
2.6.2	Tabla de riesgos	18
2.6.3	Planes de contingencia:	19
2.7	Planificación de calidad.....	21
2.7.1	Calidad en el proyecto	21
2.7.2	Calidad en el producto	21
2.7.3	Aseguramiento y control de calidad.	21
2.8	Entorno de trabajo.....	22
2.8.1	CAD.....	22
2.8.2	PDM.....	22
2.8.3	ERP	22
3.	Estado del arte: Integración CAD-ERP de Ibermática	23
3.1	Funcionalidades	23
3.2	Arquitectura de la aplicación	27
3.3	Casos de uso generales de la aplicación	29
3.3.1	Configuración general	29
3.3.2	Configuración CAD	39
3.3.3	Configuración ERP (RPS2012)	43
3.3.4	Información de artículo.....	50
3.3.5	Enviar lista de materiales.....	55
4.	Diseño	58
4.1	Gestión avanzada de listas de materiales.....	58

4.1.1 Reagrupar desde CAD:	59
4.1.2 Importar componente CAD:.....	59
4.1.3 Importar material desde ERP:	59
4.1.4 Editar datos de artículos:	60
4.1.5 Subir posición de material:	60
4.1.6 Bajar posición de material:	60
4.1.7 Introducir nueva posición manual:	61
4.1.8 Desagrupar materiales:.....	61
4.1.9 Agrupar materiales:	61
4.1.10 Eliminar material:.....	61
4.1.11 Importar ruta:	62
4.1.12 Importar todas las rutas:.....	62
4.1.13 Asignar operación de ruta a material	62
4.1.14 Guardar cambios (ejecución en CAD)	63
4.1.15 Guardar cambios (ejecución sin CAD).....	63
4.1.16 Guardar tablas en BBDD	63
4.1.17 Comparación CAD-TR.....	63
4.1.18 Codificar artículos	64
4.1.19 Enviar lista de materiales a estructura	64
4.1.20 Enviar lista de materiales a proyecto.....	64
4.2 Multilinguaje	65
4.3 Gestión de calidad.....	65
4.4 Archivo asociado en PDM	65
4.5 Mantenimiento de operaciones en lista de materiales	65
4.6 Entidades extendidas	65
4.7 Llamadas múltiples a métodos del servicio RPS2012	65
4.8 Siguiete revisión en PDM	65
4.9 Codificación manual RPS2012.....	66
4.10 Filtrar parámetros en ficha de artículo	66
4.11 Multiempresa en RPS2012.....	66
4.12 Comprobación de licencia para servicio RPS2012	66
4.13 Módulo iWIN.....	66
4.14 Módulo iSE	66
5. Implementación	67

5.1 Tecnologías	67
5.1.1 Presentación	67
5.1.2 Lógica de negocio.....	67
5.1.3 Acceso a datos	67
5.2 Software empleado.....	68
5.2.1 Microsoft Visual Studio	68
5.2.2 Microsoft SQL-Server	68
5.2.3 Internet Information Services	68
5.2.4 VMWare Player	69
5.2.5 SolidWorks	69
5.2.6 Enterprise PDM	69
5.2.7 Solid Edge.....	69
5.2.8 RPS2012	70
5.3 Gestión avanzada de listas de materiales - arquitectura de la funcionalidad	71
5.4 Descripción de las nuevas clases	72
5.4.1 Lista_de_materiales2.....	72
5.4.2 LM2_Gestion.....	77
5.4.3 Base de datos	80
5.5 Diagramas de secuencia.....	81
5.5.1 Caso de uso Reagrupar desde CAD	81
5.5.2 Caso de uso Importar componente CAD	86
5.5.3 Caso de uso Importar material desde ERP.....	88
5.5.4 Caso de uso Editar datos de artículos	92
5.5.5 Caso de uso Subir posición de material	94
5.5.6 Caso de uso Bajar posición de material	97
5.5.7 Caso de uso Introducir nueva posición manual	98
5.5.8 Caso de uso Desagrupar materiales.....	100
5.5.9 Caso de uso Agrupar materiales	102
5.5.10 Caso de uso Eliminar material.....	104
5.5.11 Casos de uso: Importar ruta / Importar todas las rutas	106
5.5.12 Caso de uso Asignar operación de ruta a material	108
5.5.13 Caso de uso Guardar cambios (ejecución en CAD)	110
5.5.14 Caso de uso Guardar cambios (ejecución sin CAD).....	114
5.5.15 Caso de uso Guardar tablas en BBDD	116

5.5.16 Caso de uso Comparación CAD-TR.....	118
5.5.17 Caso de uso Codificar Todos (ejecución en CAD).....	120
5.5.18 Caso de uso Codificar Todos (ejecución sin CAD)	124
5.5.19 Caso de uso Enviar lista de materiales a estructura	126
5.5.20 Caso de uso Enviar lista de materiales a proyecto	130
6. Pruebas	133
6.1 Pruebas de robustez	133
6.1.1 Agrupar artículos.....	133
6.1.2 Desagrupar artículos	133
6.1.3 Importar componente CAD.....	133
6.1.4 Importar artículo ERP.....	133
6.1.5 Eliminar nodo	133
6.1.6 Subir posición.....	133
6.1.7 Bajar posición.....	133
6.1.8 Asignar posición manual	134
6.1.9 Importar ruta	134
6.2 Errores corregidos.....	134
6.2.1 Formulario bloqueado	134
6.2.2 No se importan operaciones (RPS2012)	134
6.3 Análisis de rendimiento	135
7. Seguimiento	136
7.1 Desviación respecto a lo planificado.....	136
7.1.1 Desglose final de tareas	136
7.1.2 Gantt planificado vs Gantt real	139
7.2 Riesgos activados	140
7.3 Resumen de reuniones	141
7.4 Control de calidad	141
8. Resultados y conclusiones	142
8.1 Resultados.....	142
8.2 Conclusiones	142
8.2.1 Lecciones aprendidas.....	142
9. Bibliografía	143
Anexos.....	144
A1. Análisis de funcionalidades adicionales.....	144

A1.1 Multilinguaje.....	144
A1.2 Gestión de calidad	145
A1.3 Archivo asociado en PDM.....	146
A1.4 Mantenimiento de operaciones y posiciones en listas de materiales (estructuras).....	147
A1.5 Entidades extendidas en RPS2012.....	148
A1.6 Llamadas múltiples a métodos del conector RPS2012	149
A1.7 Siguiete Revisión PDM	150
A1.8 Codificación manual de artículos en RPS2012.....	152
A1.9 Filtrar parámetros en formulario de artículo	153
A1.10 Trabajo con grupos empresariales en RPS2012	154
A1.11 Comprobación de licencia para utilizar el servicio conector RPS2012	155
A1.12 Módulo iWIN.....	156
A1.13 Módulo iSE	162
A2. Resumen de implementación de funcionalidades adicionales	175
A2.1 Multilinguaje.....	175
A2.2 Gestión de calidad	179
A2.3 Archivo asociado en PDM	184
A2.4 Mantenimiento de operaciones y posiciones en reenvíos LdM.....	187
A2.5 Siguiete Revisión PDM	190
A2.6 Filtrar parámetros en formulario de artículo	194
A2.7 Codificación manual de artículos en RPS2012.....	197
A2.8 Trabajo con grupos empresariales en RPS2012	200
A2.9 Comprobación de licencia para utilizar el servicio conector RPS2012	204

1. Introducción

El proyecto Gestión Avanzada de Listas de materiales de CAD_ERP consiste en el desarrollo de una nueva funcionalidad de manipulación y envío de listas de materiales para la aplicación Integración ERP de la empresa Ibermática.

Dado que el intercambio de datos entre los puestos de diseño y los puestos de gestión de recursos en una empresa puede llegar a ser una tarea de alto coste en tiempo, los programas de integración CAD-ERP surgen como respuesta a este problema, a fin de optimizar el tiempo empleado en comunicar la información relevante entre departamento.

Una integración CAD-ERP debe ser capaz de realizar de forma eficiente las siguientes labores:

- Codificación de artículos diseñados en el módulo CAD, con respecto a la metodología empleada por el módulo ERP
- Registrar artículos y ensamblajes nuevos en las bases de datos del programa ERP
- Creación de listas de materiales a partir de diseños realizados en CAD, y envío de dichas listas de materiales al programa ERP asignado

Una lista de materiales es una lista de las materias primas, subconjuntos, conjuntos intermedios, subcomponentes, componentes, partes y las cantidades de cada uno, necesarios para fabricar un producto final. Se trata de una estructura de datos jerárquica en la que constan los distintos componentes de un ensamblaje 3D generado en un programa CAD, junto con los materiales de que dichos componentes están compuestos. Una lista de materiales debe especificar información relativa a los componentes que sea de relevancia en el programa ERP pertinente, a fin de que el departamento de gestión cuente con la información necesaria para la adquisición y logística de los artículos que intervienen en la fabricación del ensamblaje.

Al inicio de este proyecto, se cuenta con una versión inicial de una integración CAD-ERP, la cual permite una gestión básica de listas de materiales. Sobre esta versión inicial, se desarrollarán nuevas funcionalidades para lograr una gestión avanzada de listas de materiales que facilite la manipulación de éstas directamente desde los puestos de diseño.

El proyecto se ha desarrollado siguiendo la metodología ágil de programación extrema; la propiedad del código generado queda sujeta a la propiedad de Ibermática S.A, y por lo tanto no se muestra en su totalidad.

2. Documento de objetivos del proyecto

2.1 Alcance

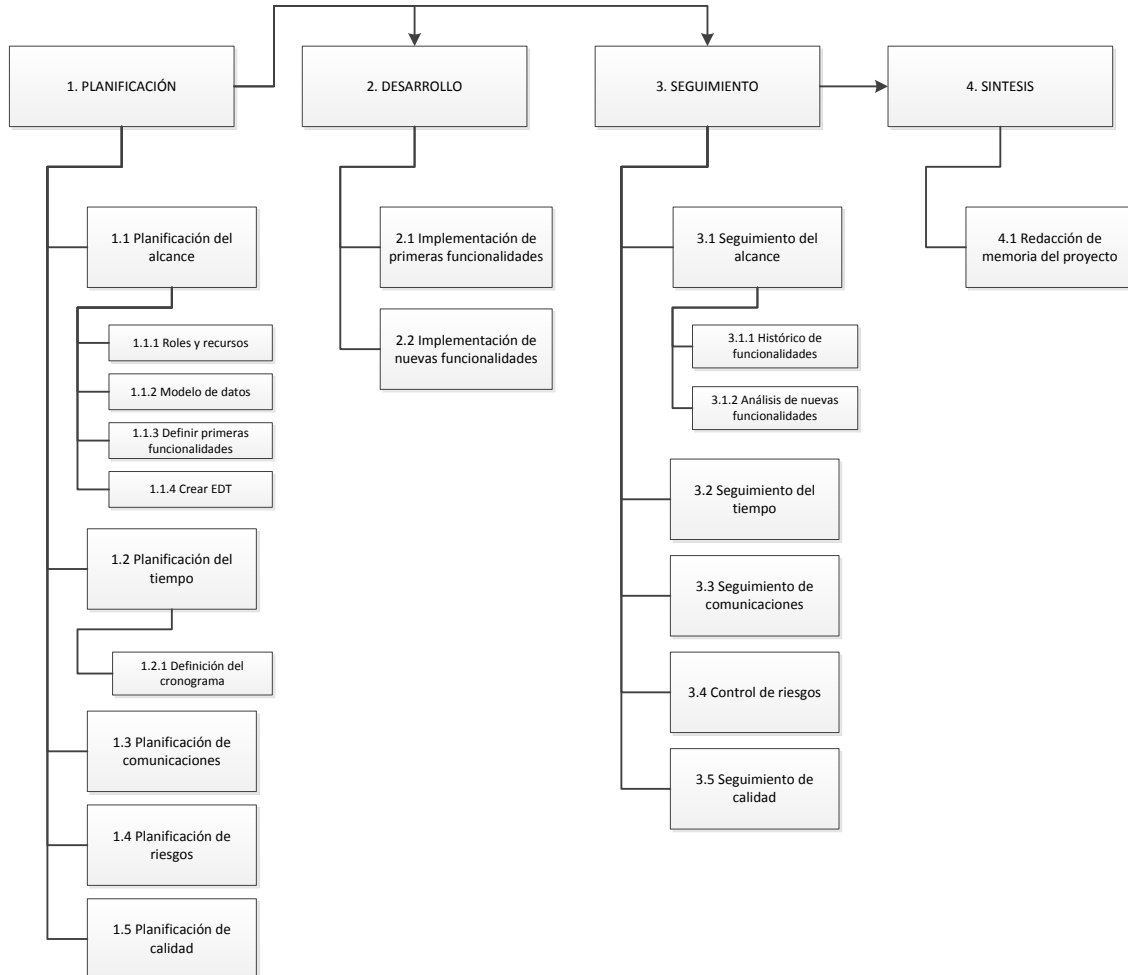
2.1.1 Gestión avanzada de listas de materiales

Planificar y desarrollar una ampliación de la funcionalidad de envío de lista de materiales. Para ello, se contempla incluir las siguientes funcionalidades iniciales en la integración:

- Manipulación de listas de materiales: Además de la capacidad de la integración de extraer los materiales de un ensamblaje y organizarlos en una estructura jerárquica acorde al diseño realizado en CAD (en adelante Árbol CAD), se plantea la posibilidad de que la integración modifique parámetros de los elementos que forman la estructura de la lista de materiales, así como la propia estructura.
Para ello, se contará con una estructura jerárquica auxiliar (en adelante Árbol de Transformación, o Árbol TR) que permita visualizar de forma gráfica los cambios que se realicen sobre la lista de materiales original.
- Comparación entre árbol CAD y árbol TR: Cuando el envío de una lista de materiales desde el programa CAD se repita, la aplicación mostrará en árbol CAD la nueva propuesta extraída desde el programa CAD, y en el árbol TR la anterior transformación realizada. La aplicación resaltará por medio de un código de colores las diferencias entre filas del árbol CAD y del árbol TR
- Visualización de la estructura en ERP: la integración contará con una estructura jerárquica adicional que representará la última versión activa de la lista de materiales en el ERP (en adelante árbol ERP). En función de si el sistema ERP en uso contempla la gestión de rutas / operaciones, estas podrán visualizarse en el árbol ERP e importarlas al árbol TR para su gestión en la lista de materiales.

Ámbito: Listas de materiales

2.2 Estructura de desglose de trabajo



2.3 Roles (empresa)

Desarrollador: Entre sus responsabilidades se encuentran:

Analizar los requisitos y las funcionalidades planteadas por el jefe de grupo,

- Estudiar su viabilidad e
- Implementar aquellas que sean aprobadas por el encargado de departamento.
- Responde ante el jefe de grupo y el jefe de departamento.

Jefe de grupo: El responsable de supervisar el progreso de un grupo de trabajo (compuesto por un número variable de desarrolladores), entre sus tareas:

- Reunirse con clientes o representantes para:
- Analizar requisitos
- Realizar demostraciones de productos
- Implementar funcionalidades junto con el equipo de desarrollo.
- Responde ante el jefe de departamento.

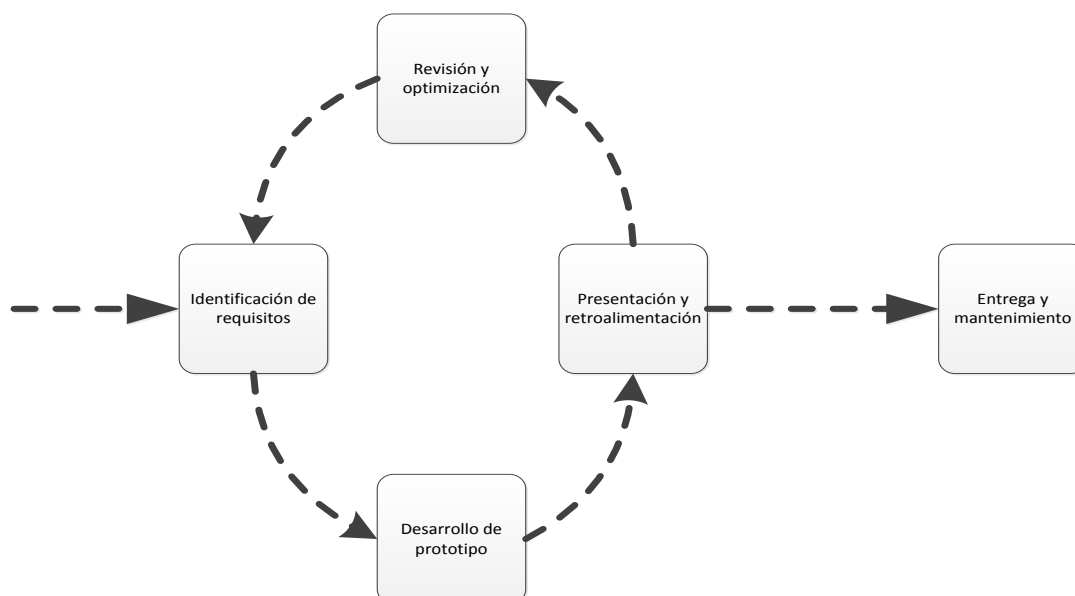
Jefe de departamento: Es el responsable de:

- Planificar las tareas a desarrollar por los grupos de trabajo del departamento
- Reunirse con clientes para
- Realizar presentaciones de productos y resultados
- Analizar requisitos

2.4 Metodología de trabajo

Para el desarrollo de este producto software, el departamento de ingeniería – soluciones empresariales de Ibermática sigue una metodología de desarrollo colaborativo ágil, basada en un modelo de prototipo evolutivo; este modelo favorece la interacción entre el equipo de desarrollo y el cliente, permitiendo una rápida identificación de requisitos, y ajustando las expectativas de los interesados. El desarrollo de éste proyecto se rige por esta misma metodología.

2.4.1 Ciclo de vida



Identificación de requisitos

Mediante reuniones con representantes de clientes e interesados en el producto, o de carácter interno, los miembros del departamento de ingeniería extraen los requisitos que deberá el cumplir el producto de cara al próximo ciclo de desarrollo. Estos requisitos se recopilan en un documento único a consultar por el grupo de desarrollo, categorizados según duración estimada, cliente destinatario y ciclo de desarrollo correspondiente.

Cuando deriva de una fase de revisión y optimización, la fase de identificación de requisitos se centra en el análisis de los cambios solicitados, y en la corrección de errores detectados.

Desarrollo del prototipo

Una vez constituido el documento de requisitos, y en base a las prioridades que les hayan sido asignadas, el equipo de desarrollo analiza la viabilidad de cada uno y procede a la implementación de

aquellos designados como viables. Dependiendo del nivel de divergencia exigido por el requisito o grupo de requisitos en cuestión, pueden generarse bifurcaciones en alguna de las ramas de desarrollo del producto. La integridad de estas nuevas bifurcaciones respecto del resto de ramificaciones se mantendrá sincronizando los cambios realizados según se estime necesario.

Los entregables resultantes del desarrollo se presentan a los clientes en forma de nuevas funcionalidades del producto, en presentaciones o demostraciones aplicadas.

Presentación y retroalimentación

El prototipo construido con la(s) nueva(s) funcionalidad(es) se entrega al personal responsable de su implantación en el destinatario; estos responsables testean el producto en un entorno que simula las condiciones en el destino. Si el producto supera unas pruebas de estabilidad, se hace entrega al destinatario para que realice pruebas adicionales en el destino.

Entrega y mantenimiento

Las pruebas realizadas en el destino determinan la viabilidad de las nuevas funcionalidades, informando al equipo de desarrollo de potencialidades, carencias y posibles errores detectados durante su uso. Los niveles de satisfacción se comunican de vuelta al departamento de implantación, y se remiten de vuelta al equipo de desarrollo, para obrar en consecuencia, corrigiendo u optimizando según sea preciso.

Revisión y optimización

Se analizan los informes remitidos por el destinatario, estudiando en cada caso el curso de acción a seguir, aplicando correcciones en el caso de errores detectados y optimizando donde la ejecución resulte costosa en función de las condiciones en el destino. Además se estudian las potencialidades que hayan podido identificarse, generando en ocasiones nuevas funcionalidades del producto. Como resultado de este proceso se constituye un informe de requisitos, con el formato adecuado a las necesidades de implantación de estas nuevas funcionalidades y correcciones.

2.4.2 Herramientas para el trabajo en equipo

Para coordinar el desempeño del equipo de desarrollo, el departamento cuenta con la herramienta de gestión de ciclo de vida Team Foundation Server, integrada en el Framework de Visual Studio 2008. Entre las funcionalidades de esta herramienta se cuentan:

- Control de versiones del código fuente
- Gestión de bifurcaciones de proyectos y soluciones
- Métricas de rendimiento y consumo de recursos

2.5 Planificación del tiempo

La duración de las actividades se mide en días o jornadas de trabajo, de una duración de 8 horas cada una. No obstante, no todo el desempeño de una jornada se invierte en una única actividad, permitiéndose el solapamiento de varias actividades en una misma jornada. El tiempo total empleado se desglosa en el apartado de seguimiento del tiempo.

Nombre de tarea	Duración	Comienzo	Fin
PLANIFICACION	15 días	vie 12/10/12	mié 31/10/12
Planificación del alcance	3 días	vie 12/10/12	lun 15/10/12
Definición de recursos y roles	1 día	vie 12/10/12	vie 12/10/12
Definición del modelo de datos	3 días	vie 12/10/12	lun 15/10/12
Definición de funcionalidades iniciales	1 día	vie 12/10/12	vie 12/10/12
Esquema de desglose de trabajo	2 días	sáb 13/10/12	dom 14/10/12
Planificación del tiempo	14 días	dom 14/10/12	mié 31/10/12
Definición del cronograma	14 días	dom 14/10/12	mié 31/10/12
Planificación de comunicaciones	13 días	lun 15/10/12	mié 31/10/12
Planificación de riesgos	12 días	mar 16/10/12	mié 31/10/12
Planificación de calidad	10 días	jue 18/10/12	mié 31/10/12
DESARROLLO	116 días	lun 22/10/12	lun 01/04/13
Desarrollo de funcionalidades iniciales	31 días	lun 22/10/12	sáb 01/12/12
Desarrollo de nuevas funcionalidades	108 días	jue 01/11/12	lun 01/04/13
SEGUIMIENTO	151 días	mié 31/10/12	jue 30/05/13
Seguimiento del alcance	151 días	mié 31/10/12	jue 30/05/13
Histórico de funcionalidades implementadas	484,25 días	mié 31/10/12	jue 11/04/13
Análisis de nuevas funcionalidades	619,25 días	lun 05/11/12	jue 30/05/13

Seguimiento del tiempo	631,25 días	mié 31/10/12	jue 30/05/13
Seguimiento de comunicaciones	631,25 días	mié 31/10/12	jue 30/05/13
Control de riesgos	625,25 días	mié 31/10/12	mar 28/05/13
Seguimiento de calidad	619,25 días	mié 31/10/12	dom 26/05/13
SINTESIS	1 día	mar 30/04/13	mar 30/04/13
Redacción de memoria del proyecto	1 día	mar 30/04/13	mar 30/04/13

2.6 Planificación de riesgos

2.6.1 Identificación de riesgos según origen

Recursos humanos:

- Rendimiento individual: El trabajo de alguno de los miembros del equipo puede verse afectado por los siguientes factores:
- Enfermedad: En un momento dado, algún miembro del equipo puede caer enfermo, viéndose afectado el trabajo que este debiera realizar en alguna de las fases del proyecto:
 - Coordinador: Si el coordinador cayese enfermo, repercutiría sobre la supervisión del trabajo realizado, imposibilitando las reuniones mientras dure la enfermedad (impacto: leve; probabilidad: baja)
 - Director / jefe de departamento: De caer enfermo el director, se verá afectada la planificación continua de las tareas mientras dure la enfermedad. También se verán afectadas las reuniones que el director tenga programadas (impacto: medio; probabilidad: baja)
 - Jefe de grupo: De caer enfermo el jefe de grupo, se verá afectado el rendimiento del equipo de desarrollo en las tareas de análisis de requisitos e implementación de funcionalidades (impacto: medio; probabilidad: baja)
 - Desarrollador: De caer enfermo alguno de los desarrolladores, se verá afectado su rendimiento en el desarrollo de sus tareas asignadas mientras dure la enfermedad. (impacto: medio; probabilidad: baja)
- Ausencia por motivos personales o profesionales: Si cualquiera de los miembros del equipo debiera ausentarse por razones ajenas al proyecto, su rendimiento sobre el mismo se vería afectado de forma similar al caso de enfermedad, con la salvedad de que en caso de ausencia es habitual conocer la fecha de retorno del miembro ausente.
 - Coordinador: Su ausencia repercutiría sobre la supervisión del trabajo realizado, imposibilitando las reuniones mientras dure la ausencia (impacto: leve; probabilidad: media)
 - Director / jefe de departamento: De ausentarse el director, se verá afectada la planificación continua de las tareas mientras dure la ausencia. También se verán afectadas las reuniones que el director tenga programadas (impacto: leve; probabilidad: media)
 - Jefe de grupo: De ausentarse el jefe de grupo, se verá afectado el rendimiento del equipo de desarrollo en las tareas de análisis de requisitos e implementación de funcionalidades (impacto: medio; probabilidad: media)
 - Desarrollador: De ausentarse alguno de los desarrolladores, se verá afectado su rendimiento en el desarrollo de sus tareas asignadas mientras dure la ausencia. (impacto: medio; probabilidad: baja)
- Funcionamiento del equipo: El desarrollo del producto se realiza por medio del trabajo conjunto del equipo de proyecto. Los factores que pueden afectar a este trabajo conjunto son:

- Fallo menor de coordinación entre miembros del equipo de desarrollo: En caso de descoordinación, el trabajo de uno o varios desarrolladores podría verse comprometido (sobreescritura, borrado...). (impacto: leve; probabilidad: baja)
- Fallo medio de coordinación entre miembros del equipo de desarrollo: En caso de descoordinación (sobreescritura, borrado...), y siendo el trabajo afectado de volumen o importancia considerables, el progreso puede verse seriamente comprometido. (impacto: medio; probabilidad: baja)
- Fallo mayor de coordinación entre miembros del equipo de desarrollo: En caso de descoordinación (sobreescritura, borrado...), y siendo el trabajo afectado de importancia crítica el progreso puede verse seriamente comprometido (impacto: alto; probabilidad: baja)

Sistemas de información:

- Fallo del sistema de correo: un fallo en el sistema de correo electrónico repercutiría sobre las comunicaciones entre los miembros del equipo (impacto: leve; probabilidad: media).
- Fallo del sistema de almacenamiento de datos: un fallo en el sistema de almacenamiento de datos provocaría retrasos en tareas de desarrollo o de seguimiento. (impacto: alto; probabilidad: media)
- Fallo del equipo de trabajo (PC): un fallo en las estaciones de trabajo podría producir retrasos en tareas de desarrollo o de seguimiento. (impacto: alto; probabilidad: media)

2.6.2 Tabla de riesgos

IMPACTO	MUY LEVE	LEVE	MEDIO	ALTO	MUY ALTO
PROBABILIDAD					
MUY BAJA				Fallo mayor coordinación	
BAJA		Enfermedad coordinador Fallo medio coordinación	Enfermedad director Enfermedad jefe de grupo Enfermedad / ausencia desarrollador		
MEDIA	Fallo menor coordinación	Ausencia coordinador Ausencia director Fallo sistema correo	Ausencia jefe de grupo	Fallo en almacén de datos Fallo del PC de trabajo	
ALTA					
MUY ALTA					

2.6.3 Planes de contingencia:

Enfermedad:

- **Coordinador:** En caso de enfermedad del coordinador, se retrasarán aquellas reuniones que estén planificadas durante la duración de la enfermedad.
- **Director:** En caso de enfermedad del director, se acometerán aquellas tareas planificadas de antemano para la semana. Si el director sigue enfermo al término de las tareas planificadas, el equipo de desarrollo, con la aprobación del jefe de grupo, planificará nuevas tareas a realizar.
- **Jefe de grupo:** En caso de enfermedad del jefe de grupo, se acometerán aquellas tareas planificadas de antemano para la semana. Si el director sigue enfermo al término de las tareas planificadas, el equipo de desarrollo, con la aprobación del director, planificará nuevas tareas a realizar.
- **Desarrollador:** En caso de enfermedad de uno de los desarrolladores, el resto del equipo de desarrollo acometerá las tareas asignadas a dicho desarrollador si estas son de importancia crítica; si el plan de trabajo contempla el retraso de las tareas, se postergarán mientras dure la enfermedad.

Ausencia:

- **Coordinador:** En caso de ausencia no planificada, se aplicará el mismo plan de contingencia que en el caso de enfermedad del coordinador. Si la ausencia es conocida de antemano, el plan de comunicaciones contempla esta eventualidad.
- **Director:** En caso de ausencia del director, se acometerán aquellas tareas planificadas de antemano para la semana. Si el director sigue ausente al término de las tareas planificadas, el equipo de desarrollo, con la aprobación del jefe de grupo, planificará nuevas tareas a realizar.
- **Jefe de grupo:** En caso de ausencia del jefe de grupo, se acometerán aquellas tareas planificadas de antemano para la semana. Si el director sigue ausente al término de las tareas planificadas, el equipo de desarrollo, con la aprobación del director, planificará nuevas tareas a realizar.
- **Desarrollador:** En caso de ausencia no planificada de uno de los desarrolladores, el plan de contingencia a aplicar es el mismo que en el caso de enfermedad del desarrollador.

Descoordinación:

- **Menor:** En caso de descoordinación menor (pérdida del progreso de un día de trabajo o menos), se restauraría el progreso a la fecha más próxima por medio de la herramienta de control de versiones disponible en el ámbito de la empresa.
- **Media:** En caso de descoordinación media (pérdida del progreso de una semana de trabajo aprox.), se restauraría todo lo posible el progreso por medio de la herramienta de control de versiones disponible en el ámbito de la empresa, y se volvería a implementar el trabajo perdido, con la asistencia de la documentación generada.
- **Mayor:** En caso de pérdida irreversible de información crítica se convocaría una reunión de carácter urgente dentro del ámbito de la empresa para decidir el rumbo a tomar al respecto.

Fallo técnico:

- Sistema de correo: En caso de fallo del sistema de correo electrónico, y si el Centro de Atención al Usuario de la empresa no consiguiese solventar la incidencia, se utilizarían medios alternativos de comunicación, como el teléfono, hasta que el sistema de correo fuese restaurado.
- Sistema de almacenamiento de datos: En caso de fallo irreversible del sistema de almacenamiento de datos, se recurriría a la copia de seguridad disponible en la empresa para restaurar el progreso todo lo posible.
- Equipo de trabajo (PC): En caso de fallo de uno de los equipos de trabajo en la oficina, y si el Centro de Atención al Usuario de la empresa no consiguiese solventar la incidencia, se recurriría a otra de las estaciones de trabajo disponibles en la oficina, aplicando el plan de contingencia correspondiente al fallo del sistema de almacenamiento de datos si hubiese pérdida de progreso debida al fallo del PC. Por el contrario, si el equipo conflictivo fuese el ordenador personal del alumno (recurso MV), éste tendrá que restituirlo a la mayor brevedad asumiendo los costes.

2.7 Planificación de calidad

2.7.1 Calidad en el proyecto

La calidad mínima en este proyecto serán los requisitos definidos en la planificación y definición del alcance, producto de una correcta comprensión, evaluación y definición de las expectativas del mismo, cumpliendo con los plazos acordados.

2.7.2 Calidad en el producto

La calidad en el producto se asegurará mediante pruebas unitarias de las funcionalidades implementadas, así como mediante líneas de mejora destinadas a reducir el coste computacional de dichas funcionalidades. Estas líneas de mejora se plantearán de forma individual, o mediante tormentas de ideas entre los miembros del grupo de desarrollo, y estarán reflejadas en el documento histórico de funcionalidades, anotadas junto a cada funcionalidad al finalizar su implementación.

2.7.3 Aseguramiento y control de calidad.

El aseguramiento de calidad se realizará de forma periódica, comprobando de forma semanal el documento que contiene las líneas de mejora anotadas durante la implementación de funcionalidades, convirtiendo aquellas que sean aprobadas por el jefe de departamento en paquetes de trabajo, planificadas junto al resto de tareas a realizar.

De forma mensual se medirá el rendimiento de la totalidad del producto (tiempo de ejecución, coste computacional, utilización de recursos...) por medio de las herramientas de *profiling* disponibles en Visual Studio, determinando los 'cuellos de botella' en el código, y optimizando funciones según lo apruebe el jefe de grupo, o el jefe de departamento en su defecto.

Métricas para el control de calidad

- Tiempo de ejecución: Se medirá el tiempo de ejecución total de cada caso de uso contemplado, el número de llamadas a cada función que intervenga en dicho caso de uso y el tiempo exclusivo en cada una de dichas funciones.
- Uso de memoria: Se medirá la memoria total utilizada por cada caso de uso, y el tiempo de vida de los objetos creados durante la ejecución de dicho caso de uso.
- Pruebas unitarias: Se diseñaran casos de prueba para aquellos métodos críticos a fin de medir su robustez.

Herramientas para el control de calidad

Visual Studio Profiler: Herramienta para medir tiempos de ejecución, tiempo de vida de objetos y recursos utilizados por aplicaciones desarrolladas mediante Visual Studio. Permite realizar mediciones por muestreo o por instrumentación.

Visual Studio Testing Tools: Herramienta para creación y análisis de pruebas unitarias de código. Permite diseñar pruebas por cada método a controlar.

2.8 Entorno de trabajo

2.8.1 CAD

Un programa CAD (Computer –aided Design) es aquel que permite diseñar modelos y artículos sólidos en 3D. En el caso del diseño CAD en el ámbito mecánico, los modelos a diseñar se clasifican en las siguientes categorías:

- Piezas: modelos diseñados en un programa CAD, habitualmente compuestos de un único material. Existen categorías de piezas, como soldaduras o chapas.
- Ensamblajes: modelos compuestos de una o varias piezas u otros componentes
- Planos: se trata de dibujos que representan un modelo 3D en una o varias vistas 2D

Listas de materiales en CAD

Se le llama lista de materiales en el programa CAD a la estructura simplificada de componentes de un ensamblaje u otro modelo, especificando la cantidad de cada componente según su unidad de medida establecida.

2.8.2 PDM

Las siglas PDM (Product Data Management) hacen referencia al uso de software específico para gestionar los archivos de diseños 3D y 2D dentro de un determinado flujo de trabajo, estableciendo estados, versiones y revisiones para dichos archivos. Además de la información de control relativa a archivos CAD, también se gestionan documentos asociados a estos archivos. Los programas PDM se emplean habitualmente en entornos de ingeniería, en conjunción con programas CAD, a fin de optimizar el proceso de diseño y aumentar la productividad.

2.8.3 ERP

Un programa ERP (Enterprise Resource Planning) permite realizar la gestión económico-financiera de una empresa: inventario de artículos, logística y facturación, uso de recursos en distintos proyectos... Este tipo de aplicaciones posibilitan el llevar un control de recursos humanos, inventario, costes y plazos de diferentes proyectos, incluyendo la fabricación o adquisición de piezas y ensamblajes diseñados en un programa CAD. Es una característica de los programas ERP la de asignar códigos únicos a los artículos existentes en el inventario de la empresa.

Listas de materiales en ERP

Un sistema ERP habitualmente incluye entre sus funcionalidades la de gestionar distintos tipos de listas de materiales: aquellas que componen los materiales para la producción de una estructura de fabricación, y aquellas que constan de los materiales asignados a una etapa determinada de un proyecto de empresa.

3. Estado del arte: Integración CAD-ERP de Ibermática

La integración entre sistemas CAD y ERP desarrollada por Ibermática está diseñada para trabajar con programas CAD, y una serie de distintos programas ERP:

- NAVISION
- RPS
- SPYRO (CTI)
- EXPERTIS (SOLMICRO)

Se apoya en una herramienta de gestión de datos de productos (Product Data Management, PDM), que permite llevar un control sobre distintas versiones de archivos y permisos de usuarios sobre dichos archivos. El PDM se utiliza para gestionar los diseños realizados en el programa CAD.

Dado que distintos programas ERP se valen de distintos parámetros, aquellos parámetros con los que trabaja la integración son totalmente configurables y se definen en la herramienta de configuración. Esta herramienta permite además parametrizar los datos modificables tanto en CAD como en ERP, definir el método de asignación de códigos de artículos y gestionar el mantenimiento de las bases de datos que alimentan la integración.

3.1 Funcionalidades

Toda la información se extrae de los modelos 3D del programa CAD (ensamblajes y piezas) y PDM es la herramienta que almacena los archivos permitiendo tener un control sobre los mismos.

3.1.1 Alta de artículos:

Es el proceso que se realiza cada vez que se quiere dar de alta un nuevo artículo en el sistema ERP. También se utiliza para modificar los valores que previamente se han adjudicado a un determinado artículo.

- Asignación de artículos: Cuando se ha creado un elemento en el programa CAD, y el usuario sabe que el artículo correspondiente ya existe en el ERP, tiene la opción de teclear el código manualmente o buscarlo en la base de datos.
- Código del artículo: el método de lectura y modificación del código de un modelo del programa CAD se indica en la herramienta de configuración
 - Código desde número de pieza: el código del artículo es, por omisión, el nombre del archivo CAD, La herramienta EPDM garantiza que dicho nombre es único a fin de evitar duplicidad de códigos
 - Código desde propiedad: En este caso el código del artículo esta en una propiedad, contenida en el archivo CAD que corresponde al componente o ensamblaje, y especificada

en la configuración de la integración.

- Cantidad en Lista de Materiales: Como regla general los componentes intervienen unitariamente en las listas de materiales. Sin embargo en ciertos artículos, como podrían ser tubos de distintas medidas, es común que no se creen códigos nuevos por cada artículo de una medida individual; en estos casos la cantidad refleja la dimensión del componente en las unidades de medida del artículo.
- Tipo de componente en CAD:
 - Piezas: representan sólidos íntegros no compuestos. Tienen asignada una o varias materias primas, y son los componentes básicos del resto de componentes.
 - Ensamblajes (conjunto estándar): Un ensamblaje representa un artículo semi-elaborado que contiene una lista de materiales con los artículos necesarios para su creación, en general piezas.
 - Kits: El concepto de KIT describe un componente (ensamblaje) de CAD que tiene estructura pero en ERP es un componente final, sin estructura. La consecuencia de esto es que durante el paso de la lista de materiales los componentes del ensamblaje no serán transferidos al ERP.
 - Ficticios: Los artículos ficticios (también llamados "Fantasma") son aquellos que se crean en CAD con el fin de tener una estructura mejor organizada pero que realmente no se venden ni fabrican. Se trata de componentes que forman parte de una lista de materiales que, a la hora de analizar una necesidad no van a generar órdenes de compra o fabricación por ellos mismos. La estructura de estos componentes se va a recibir en el artículo que lo contiene y son estos componentes hijos los que se compran o fabrican (siempre y cuando no sean ficticios a su vez).
 - Excluido de LM: Se trata de elementos auxiliares del diseño, que no tienen correspondencia con ningún artículo y se utilizan a modo de apoyo visual en la fase de diseño (layouts, dummies,...)

3.1.2 Materia Prima:

Cuando las piezas diseñadas en el programa CAD utilizan solo una materia prima, es posible, habilitando esta función en la configuración, utilizar propiedades para indicar la materia prima y su cantidad; la cantidad puede ser dinámica y vinculada a datos dimensionales de la pieza como pueden ser el peso, la longitud, etc.

En el caso de sólidos disjuntos, esta funcionalidad permite codificarlos como piezas individuales y asignarles una materia prima.

También es posible asignar materia prima a ensamblajes de tipo Kit (Tipo de componente en el programa CAD).

3.1.3 Envío de listas de materiales:

Se revisan todos los artículos que componen un conjunto y se da de alta una nueva lista de materiales en el ERP. Dependiendo de la configuración permite lanzar la misma lista varias veces y el sistema alertará sobre las variaciones que contiene la lista respecto al último paso. Se incluyen en el conjunto de CAD todos los artículos, incluyendo aquellos que no se dibujan pero que se listan.

Al seleccionar la opción de Envío de Lista de Materiales se comprueban todos los artículos y su código en ERP. Si no existe algún artículo en ERP lo dará de alta previamente al registro de la lista según el procedimiento de codificación. Si existe, comprueba si hay una Lista previa para el mismo conjunto. Si existe y está permitido el reenvío de listas, las compara y envía diferencias y alertas a las personas indicadas.

- Agrupación de componentes, cantidades: Una opción de la configuración indica si los componentes de una estructura se agrupan sumando cantidades o si entrarán individualmente en la estructura en secuencias diferentes. Las opciones son las siguientes:
 - No agrupar.
 - Agrupar componentes cuya unidad de medida es la unidad (UDS): Si está activada la opción de agrupar componentes aquellos que teniendo el mismo código y están asignados a una misma operación crearán una única línea en la estructura (número de secuencia) con la cantidad correspondiente.
La excepción son los componentes que tienen una unidad de medida diferente a la unidad (UDS) en el artículo (mm. de tubo, Kg. de aceite, etc.) en cuyo caso generan entradas independientes en la estructura con el fin de tener perfectamente identificados los diferentes componentes.
 - Agrupar componentes cuya unidad de medida es UDS o es cualquier unidad y la cantidad unitaria coincide: En este caso si el componente se mide en unidades (UDS) se trata como en el caso anterior.
Si la unidad de medida diferente a la unidad (UDS) se analiza la cantidad unitaria, si coincide, se crea una única línea con la suma de las cantidades unitarias.
 - Agrupar todos los componentes y anotar lista de cortes de aquellos que no se miden en UDS: En este caso se agruparán todos los componentes independientemente de sus unidades de medida.
En los componentes que no se midan en unidades, se anotará la lista de cortes que da como resultado la cantidad total demandada.
- Reenvío de listas: En la configuración de la integración se puede establecer si las listas de materiales se pueden reenviar al ERP o no.
Si el reenvío está desactivado, una vez enviada una estructura al ERP no será posible reenviarla. Cualquier cambio en la estructura deberá ser realizado a mano en el ERP.
Si el reenvío de listas está permitido entran en consideración las versiones de las Listas de Materiales.
- Versiones de listas: Si el reenvío de Listas de materiales está permitido se tendrá en cuenta la versión correspondiente.
Al enviar una Lista se crea la versión de la misma correspondiente a la versión del documento en

PDM. Si la versión de la lista ya existe no se permitirá reenviar la lista del artículo, en caso contrario si se puede reenviar pasando a ser activa la nueva lista (versión).

- Envío a proyecto: Cuando esta activada la integración con proyectos, además de envío a la estructura de ERP es posible enviar materiales al proyecto.
El primer requisito para esto es que el ensamblaje abierto en el programa CAD represente al proyecto en cuestión. Esta asociación la aplica cuando la integración ve que el código proporcionado con el ensamblaje es el mismo que el código del proyecto PDM al que pertenece. La diferencia respecto a lo mencionado anteriormente es que al enviar a proyectos se “aplana” la estructura, se anotan únicamente los elementos codificados como recursos materiales del proyecto. Los subensamblajes intermedios desaparecen repercutiendo sus cantidades al elemento anotado.
Los componentes hijos se anotan en la estructura de dicho artículo.

3.2 Arquitectura de la aplicación

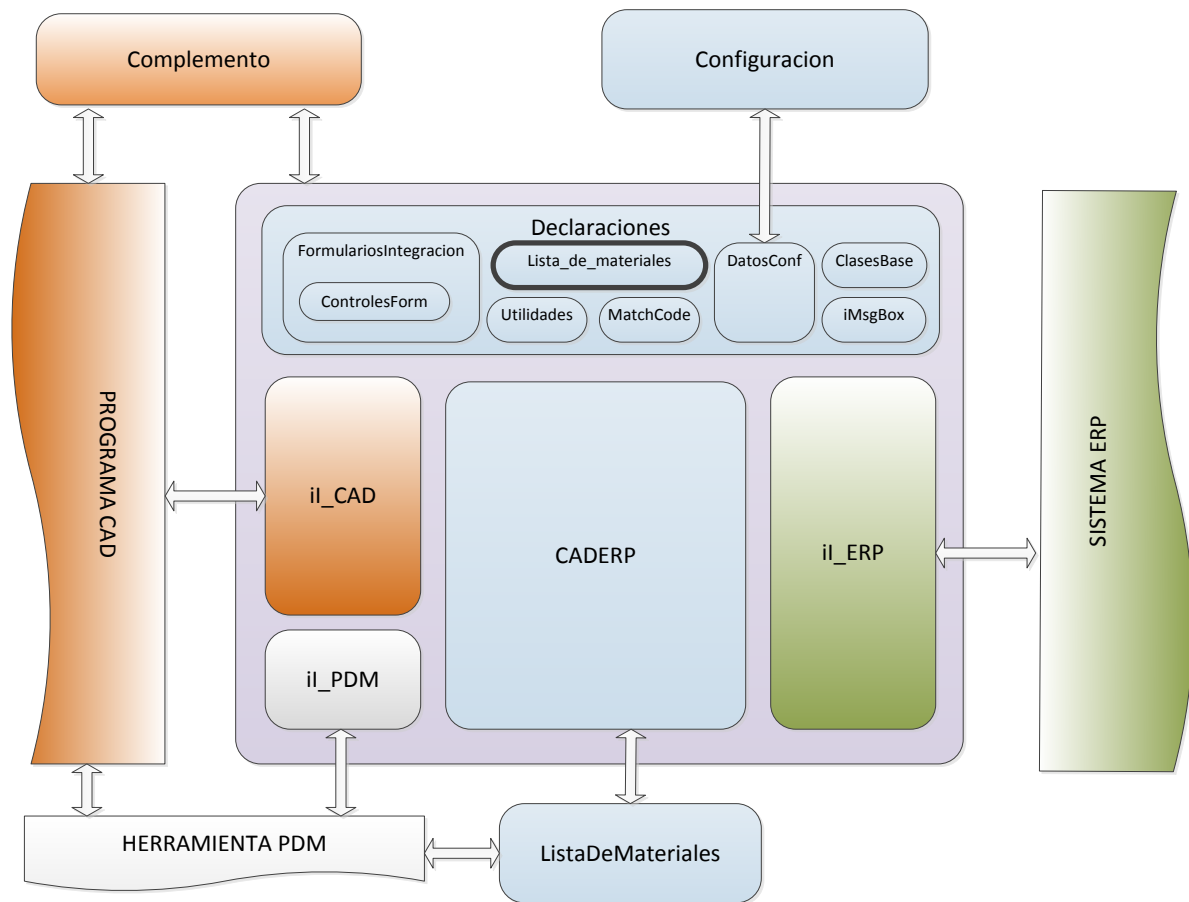


Ilustración 1: Diagrama de arquitectura de la aplicación

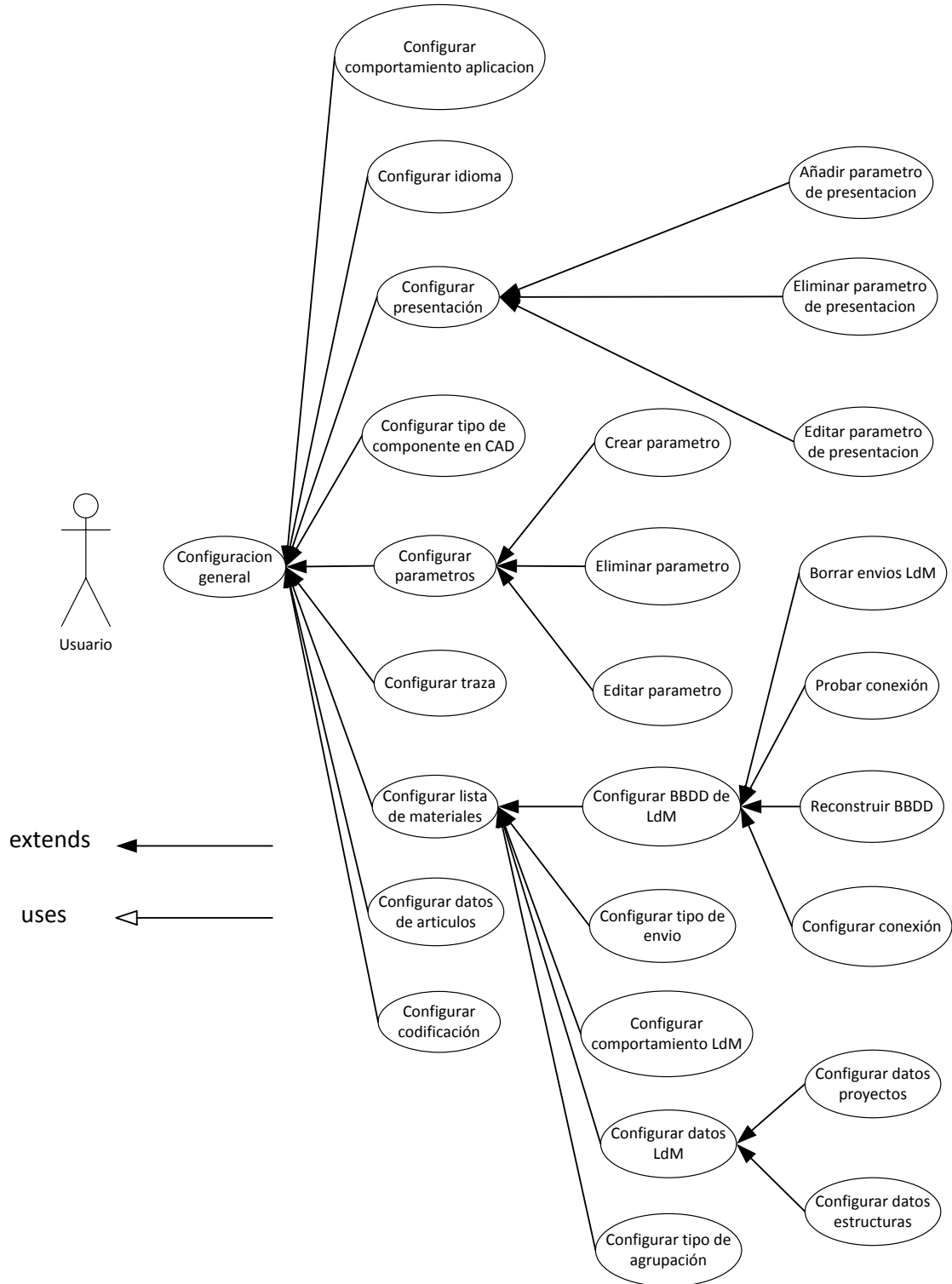
La aplicación, debido a sus necesidades de alta compatibilidad, está estructurada en una serie de módulos que forman la integración:

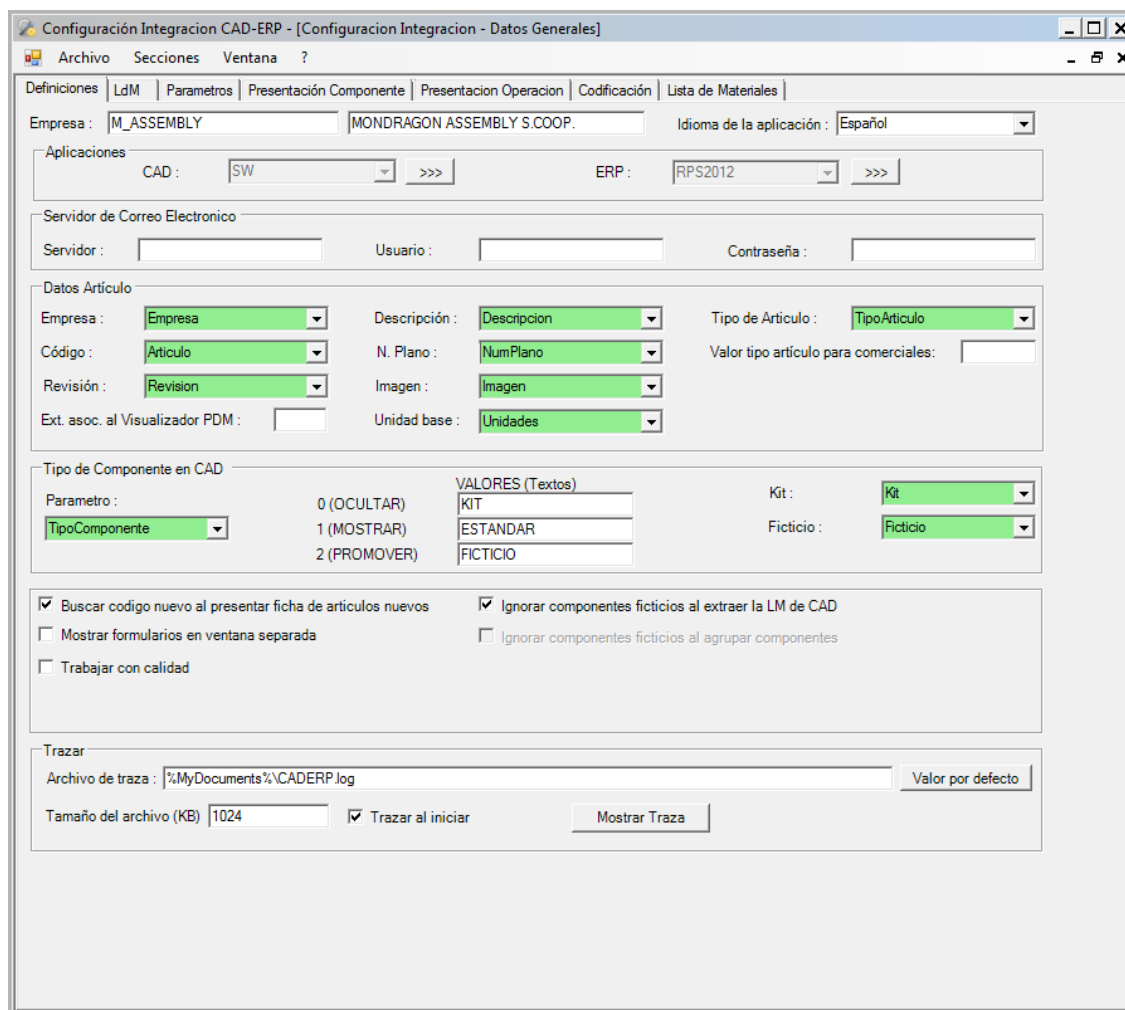
- CADERP: El 'núcleo' de la integración, desde donde se accede a todos los casos de uso de la aplicación.
- Declaraciones: Contiene los objetos y clases necesarios para la ejecución de la aplicación, tales como formularios, controles, especificaciones de constantes y funciones de utilidad.
 - FormulariosIntegracion: Biblioteca de clases con los distintos tipos de formularios personalizados empleados por la integración.
 - ControlesForm: Biblioteca de clases con un número de controles personalizados empleados por la integración.
 - DatosConf: Objeto que gestiona las opciones de configuración leídas desde el xml
 - **Lista_de_materiales: Formulario de envío de lista de materiales**
 - ClasesBase: Especificación de gran parte de clases empleadas en la integración. Se incluyen las interfaces *il_CAD*, *il_PDM* e *il_ERP*.
 - Utilidades: Módulo que contiene un número de funciones de utilidad genérica.

- MatchCode: Formulario empleado para la búsqueda y selección de artículos en el ERP
- iMsgBox: Formulario empleado a modo de cuadro de dialogo de propósito general con funcionalidades añadidas.
- Configuración: Este módulo consiste en una serie de formularios que permiten modificar el comportamiento de la aplicación para adecuarse a las necesidades de su entorno. Pueden editarse opciones globales, así como específicas a cada programa CAD o ERP. Almacena las opciones en un archivo xml.
- il_CAD: La interfaz a implementar por los módulos que comunican con los distintos programas CAD.
- il_PDM: La interfaz a implementar por los módulos que actúan de enlace con la herramienta PDM de gestión documental.
- il_ERP: La interfaz a implementar por los módulos que comunican con los distintos sistemas ERP.
- Complementos CAD-ERP: Módulos que permiten la ejecución de la integración a modo de *add-ins* de los distintos programas CAD.
- ListaDeMateriales: Módulo que permite la ejecución de la funcionalidad de envío de listas de materiales sin necesidad de ejecutar un programa CAD, apoyándose en la herramienta PDM.

3.3 Casos de uso generales de la aplicación

3.3.1 Configuración general





Configuración Integración CAD-ERP - [Configuración Integración - Datos Generales]

Definiciones | LdM | Parametros | Presentación Componente | Presentación Operación | Codificación | Lista de Materiales

Empresa : M_ASSEMBLY | MONDRAGON ASSEMBLY S.COOP. | Idioma de la aplicación : Español

Aplicaciones
 CAD : SW >>> | ERP : RPS2012 >>>

Servidor de Correo Electronico
 Servidor : | Usuario : | Contraseña :

Datos Artículo
 Empresa : Empresa | Descripción : Descripción | Tipo de Artículo : TipoArticulo
 Código : Articulo | N. Plano : NumPlano | Valor tipo artículo para comerciales :
 Revisión : Revision | Imagen : Imagen
 Ext. asociado al Visualizador PDM : | Unidad base : Unidades

Tipo de Componente en CAD
 Parametro : 0 (OCULTAR) | VALORES (Textos) | Kit : Kit
 TipoComponente | 1 (MOSTRAR) | ESTANDAR | Ficticio : Ficticio
 2 (PROMOVER) | FICTICIO

Buscar código nuevo al presentar ficha de artículos nuevos | Ignorar componentes ficticios al extraer la LM de CAD
 Mostrar formularios en ventana separada | Ignorar componentes ficticios al agrupar componentes
 Trabajar con calidad

Trazar
 Archivo de traza : %MyDocuments%\CADERP.log | Valor por defecto
 Tamaño del archivo (KB) 1024 | Trazar al iniciar | Mostrar Trazas

2. Pestaña principal del apartado de configuración general

Configurar datos artículos: Se definen los parámetros asignados a cada uno de los datos obligatorios para registrar artículos en ERP:

- Código: código con el que se registra el artículo en ERP
- Empresa: código identificativo de la empresa en ERP
- Revisión: versión / revisión del artículo
- Descripción:
- N. Plano:
- Imagen:
- Unidad base:
- Tipo de artículo:
- Extensión visualizador PDM:
- Valor tipo artículo comerciales:

Configurar comportamiento aplicación: Controla las siguientes opciones:

- Buscar código nuevo al presentar ficha de artículos nuevos: Si esta opción se encuentra activada, la integración consultará en el ERP el código con el que se registrará el artículo, y se lo pre-asigna al mostrar la ficha de artículo.
- Mostrar formularios en ventana separada: si se activa esta opción, los formularios se mostrarán en nuevas ventanas de Windows. Si se desactiva, los formularios se mostrarán en el panel de tareas del programa CAD (a excepción de formulario de envío de lista de materiales, que siempre se muestra en ventana separada)
- Ignorar componentes ficticios al extraer LdM de CAD: Si está activado, se omitirán los componentes identificados como ficticios por la integración a la hora de extraer la estructura de CAD.
- Ignorar componentes ficticios al agrupar componentes: si se activa, se omitirán los componentes ficticios al reagrupar desde CAD.
- Trabajar con calidad: activa / desactiva la gestión de calidad.

Configurar tipo de componente en CAD: Define el valor de los parámetros utilizados para identificar componentes CAD como estándar, ficticios o kits (estos últimos sólo ensamblajes).

- Parámetro: indica el parámetro asignado a almacenar el tipo de componente.
- VALORES: indica el valor del parámetro en función del tipo de componente identificado:
- ESTANDAR: el valor del parámetro cuando el componente es estándar.
- FICTICIO: el valor del parámetro cuando el componente es ficticio.
- KIT: el valor del parámetro cuando el componente es un kit (sólo ensamblajes)
- Kit: indica el parámetro que informa mediante un valor booleano si el componente es un ensamblaje de tipo Kit.
- Ficticio: indica el parámetro que informa mediante un valor booleano si el componente es ficticio.

Configurar tipo de agrupación: establece el criterio bajo el que agrupar componentes con el mismo código y bajo el mismo nivel de estructura.

- No agrupar: No hay ninguna agrupación, todos los componentes se anotan sueltos, tal y como hay sido extraídos de CAD.
- Agrupar componentes medidos en 'Unidad': Se agrupan aquellos componentes que se midan en la unidad por defecto; aquellos que se midan en otras unidades (KG, MM, etc.) no se agrupan, se anotan sueltos
- Agrupar componentes con cantidades iguales: Se agrupan aquellos componentes medidos en 'Unidad', y aquellos con cualquier unidad de medida y cantidades iguales.
- Agrupar todos: se agrupan todos los componentes.

Configurar traza: Controla las opciones de traza, entre las cuales se cuentan:

- Activar / desactivar traza
- Ubicación del archivo de traza
- Tamaño máximo del archivo de traza (al superarlo se genera un nuevo archivo y se guarda el anterior)

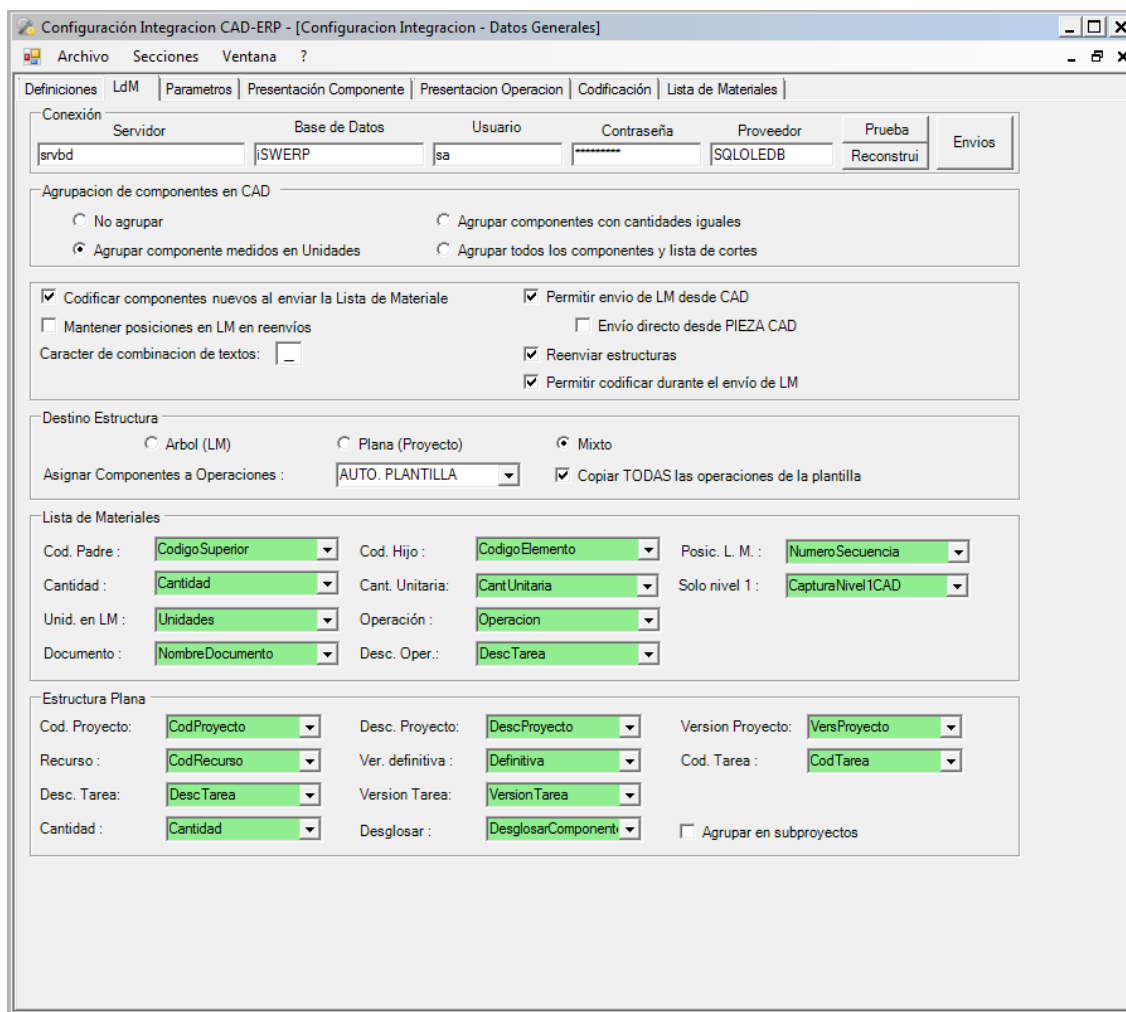
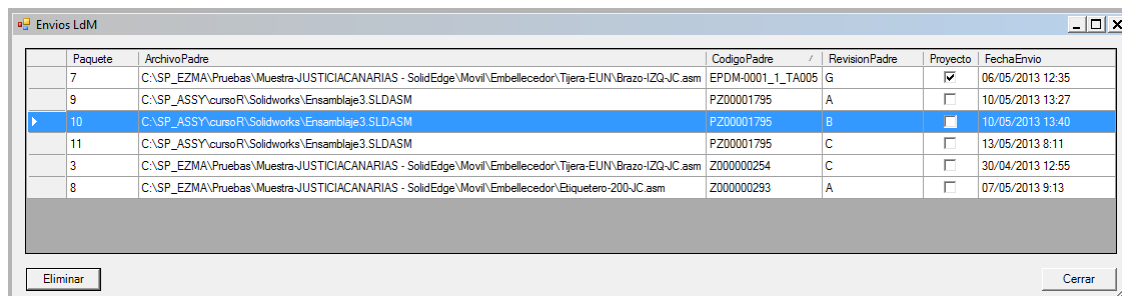


Ilustración 3. Pestaña de configuración LdM

Configurar Lista de Materiales: En este apartado se definen las opciones para configurar la funcionalidad de envío de listas de materiales:

Configurar BBDD de listas de materiales: Se definen los parámetros de conexión y funcionalidades de mantenimiento de la base de datos de la integración:

- Configurar conexión: se definen los parámetros de conexión a la base de datos: servidor, nombre de la base de datos, usuario y contraseña.
- Probar conexión: haciendo click en el botón 'Test' se realizan pruebas de conexión, indicando si las credenciales son correctas. Si la conexión se realiza con éxito, realiza pruebas adicionales de lectura, inserción y borrado de registros en las tablas de la base de datos, verificando el nivel de permiso del usuario indicado.
- Reconstruir BBDD: haciendo click en el botón 'Reconstruir', se renombran las tablas en uso actualmente por la integración, y se generan nuevas tablas limpias de datos.
- Borrar envíos LdM: haciendo click en el botón 'Envíos', se accede al formulario de visualización de envíos LdM realizados actualmente almacenados en la BBDD de la integración.



Paquete	ArchivoPadre	CodigoPadre	RevisionPadre	Proyecto	FechaEnvio
7	C:\SP_EZMA\Pruebas\Muestra-JUSTICIANARIAS - SolidEdge\Movil\Embellecedor\Tjera-EUN\Brazo-IZQ-JC.asm	EPDM-0001_1_TA005	G	<input checked="" type="checkbox"/>	06/05/2013 12:35
9	C:\SP_ASSY\cursoR\Solidworks\Ensamblaje3.SLDASM	PZ00001795	A	<input type="checkbox"/>	10/05/2013 13:27
10	C:\SP_ASSY\cursoR\Solidworks\Ensamblaje3.SLDASM	PZ00001795	B	<input checked="" type="checkbox"/>	10/05/2013 13:40
11	C:\SP_ASSY\cursoR\Solidworks\Ensamblaje3.SLDASM	PZ00001795	C	<input type="checkbox"/>	13/05/2013 8:11
3	C:\SP_EZMA\Pruebas\Muestra-JUSTICIANARIAS - SolidEdge\Movil\Embellecedor\Tjera-EUN\Brazo-IZQ-JC.asm	Z000000254	C	<input type="checkbox"/>	30/04/2013 12:55
8	C:\SP_EZMA\Pruebas\Muestra-JUSTICIANARIAS - SolidEdge\Movil\Embellecedor\Etiquetero-200-JC.asm	Z000000293	A	<input type="checkbox"/>	07/05/2013 9:13

Ilustración 4: formulario de visualización de envíos LdM

En el formulario se visualizan las propuestas de envío, las cuales vienen identificadas por el nombre de archivo del conjunto principal y su revisión, junto con la fecha en que se realizó el envío; adicionalmente, se indica si el envío se realizó sobre una estructura de fabricación o si por el contrario se trata de materiales de un proyecto.

Seleccionando una propuesta y haciendo clic en el botón 'Eliminar' se eliminan de la base de datos todos los registros con el número de paquete del envío a eliminar, tras la confirmación del usuario.

Configurar tipo de envío: en el apartado 'Destino estructura' se define el tipo de envío a realizar por la integración:

- **Arbol (LM):** las LdM se registran como estructuras de fabricación, con los materiales organizados en formato de esquema arborescente.
- **Plana (Proyecto):** las LdM se registran como listas de un único nivel, donde todos los materiales del ensamblaje del programa CAD pierden su organización jerárquica.
- **Mixto:** el tipo de envío es configurable, en función del tipo de componente asignado al ensamblaje del programa CAD (artículo o proyecto)

Además, se puede configurar la gestión de las operaciones por defecto a asignar a los componentes de la lista de materiales:

- **NO:** No se asignan operaciones a los componentes, será el ERP quien gestione la asignación de operaciones
- **AUTO. PLANTILLA:** las operaciones se asignan en función de la configuración de plantillas definida en la sección de configuración ERP. Puede definirse si copiar únicamente la primera operación de la plantilla, o todas.

Configurar datos LdM: en este apartado se definen los parámetros asignados a cada uno de los datos necesarios para la gestión de listas de materiales según su tipo de envío:

- **Configurar datos estructuras:** se asignan parámetros a los datos necesarios en los envíos a estructuras de fabricación.
- **Configurar datos proyectos:** se asignan parámetros a los datos necesarios en los envíos a listas de materiales de proyectos.

Configurar comportamiento LdM: en este apartado se definen opciones de comportamiento de la funcionalidad de envío de listas de materiales:

- Permitir el envío de LdM desde CAD: si se desactiva, no podrá lanzarse la funcionalidad de envío de listas de materiales desde el complemento del programa CAD correspondiente, únicamente desde la aplicación externa.
- Envío directo desde pieza CAD: Opción dependiente. Si se activa, los envíos LdM de piezas - consistentes en la propia y su materia prima - se realizarán de forma automática sin mostrar el formulario de envío, según las cantidades definidas en la funcionalidad de la integración 'Materia prima'.
- Reenviar estructuras: si se desactiva esta opción, no se permitirá enviar una lista de materiales si existe al menos una versión de la misma dada de alta en el ERP.
- Mantener posiciones en reenvíos: si se activa esta opción, los componentes de una propuesta de envío LdM mantendrán la misma posición que ocupaban en la versión previa en el ERP. A los componentes nuevos se les asignará nuevas posiciones.
- Permitir codificar durante el envío de LdM: si se desactiva esta opción, los componentes de la lista de materiales que no se correspondan con un artículo dado de alta en el ERP no podrán ser codificados desde el formulario de envío de listas de materiales en ningún momento.
- Codificar componentes nuevos al enviar la lista de materiales: si se activa esta opción, la aplicación de listas de materiales tratará de dar codificar los componentes que no se correspondan con ningún artículo dado de alta en el ERP, de forma automática al cargar el formulario de envío LdM.

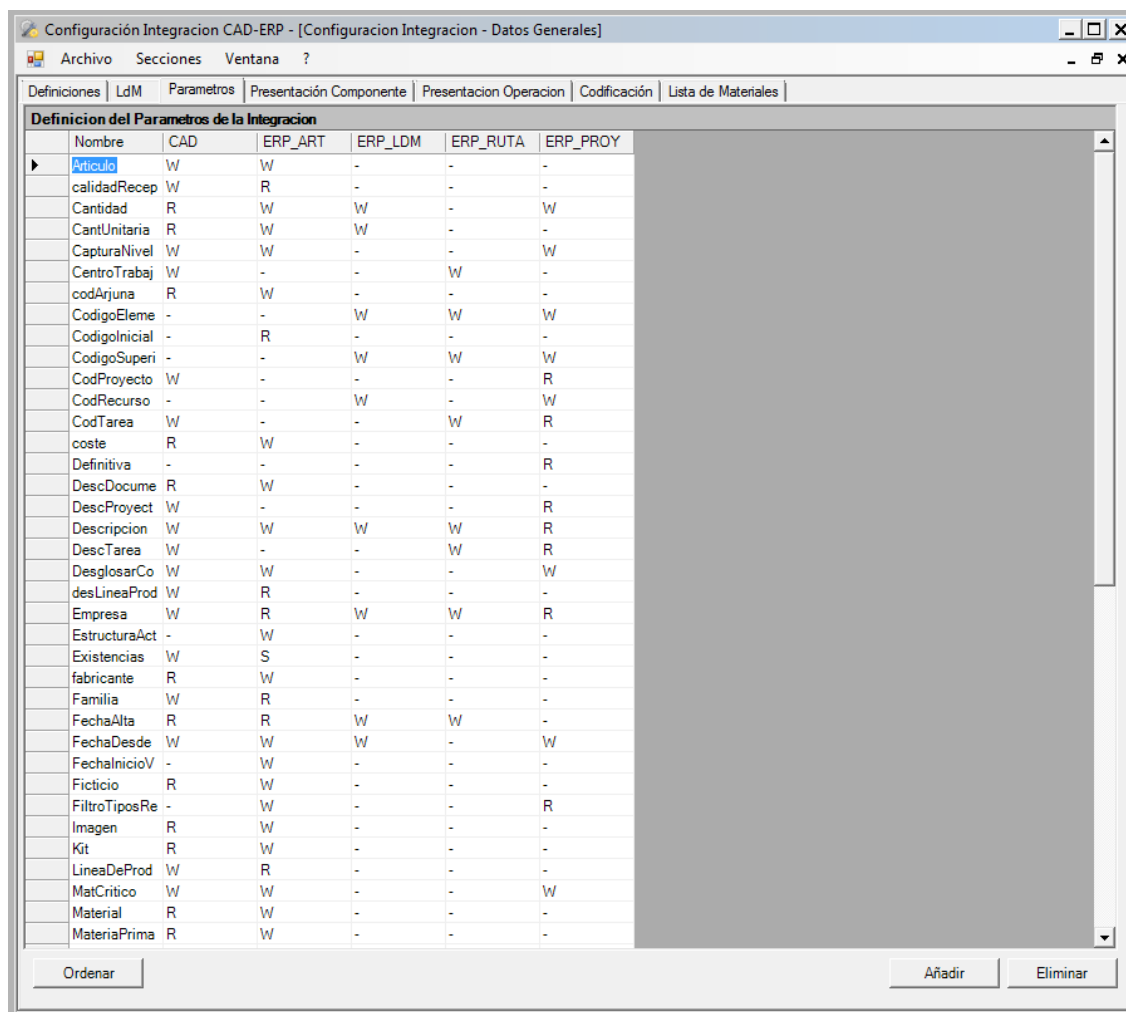


Ilustración 5: Pestaña del apartado de Parámetros

Configurar parámetros: En este apartado se accede a la configuración de los parámetros de la integración mediante las siguientes opciones:

- Crear parámetros: al hacer clic sobre el botón añadir, se pregunta al usuario por el nombre del nuevo parámetro. Tras introducir el nombre, el nuevo parámetro se introduce en la tabla, con relevancia inicial nula.
- Editar parámetros: mediante las celdas de las columnas CAD, ERP_ART, ERP_LDM, ERP_RUTA y ERP_PROY se define la relevancia del parámetro correspondiente en cada uno de los ámbitos de la integración:
 - CAD: define la relevancia del parámetro en el ámbito CAD
 - ERP_ART: define la relevancia del parámetro en el ámbito de Información de artículo.
 - ERP_LDM: define la relevancia del parámetro en el ámbito de Lista de Materiales - envío a estructuras de fabricación.
 - ERP_RUTA: define la relevancia del parámetro en el ámbito de Lista de Materiales - gestión de rutas

- ERP_PROY: define la relevancia del parámetro en el ámbito de Lista de Materiales - envío a proyectos.

La relevancia de cada parámetro viene indicada según el carácter en su correspondiente columna:

- : El parámetro no tiene relevancia en este ámbito

W: El parámetro se puede leer y modificar en este ámbito

R: El parámetro se puede leer siempre, y escribir al registrarse por primera vez en este ámbito.

S: El parámetro únicamente puede leerse en este ámbito.

La información de cada ámbito puede ser prioritaria sobre el resto de ámbitos, sobrescribiendo la información si el parámetro está definido en más de un ámbito, en función de su relevancia

(S >> R >> W).

- Eliminar parámetro: al hacer clic sobre el botón 'Eliminar' con un parámetro seleccionado en la tabla, y tras confirmarlo el usuario, se elimina el artículo de la lista, y con ello de todos los ámbitos donde tenía algún tipo de relevancia.

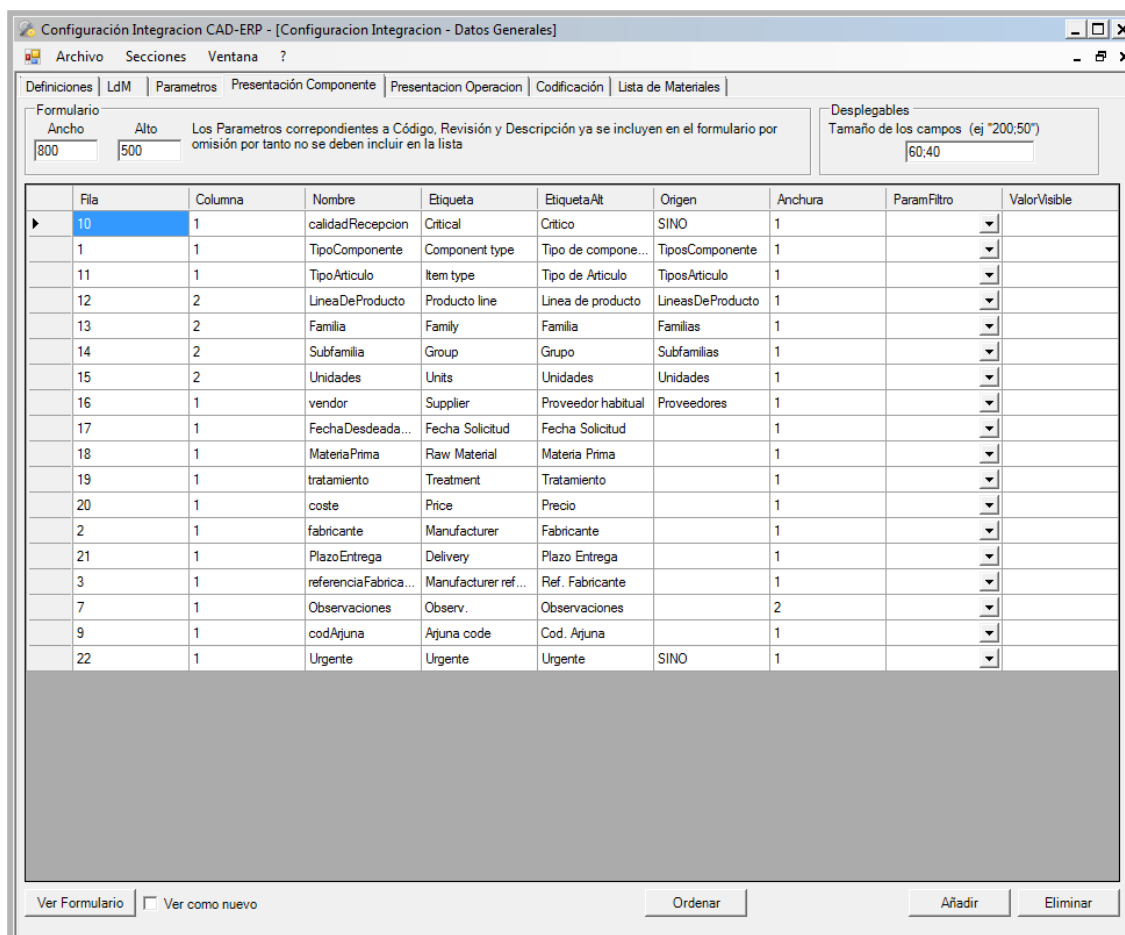


Ilustración 6: Pestaña del apartado de presentación de componente

Configurar presentación: En este apartado se definen los parámetros que se mostrarán en los formularios de ficha de artículo. La presentación de la ficha de operaciones se configura desde la pestaña 'Presentación operación' de forma análoga.

- **Añadir parámetro de presentación:** al hacer clic en botón 'Añadir', se presenta al usuario una lista con los parámetros con relevancia en ERP_ART (ERP_RUTA en el caso de operaciones). Tras escoger el usuario un parámetro, este se añade a la tabla con la fila y columna por defecto.
- **Editar parámetro de presentación:** mediante las celdas de la tabla, pueden editarse distintas propiedades de presentación del parámetro según su columna:
 - **Fila:** indica la fila en la que se presentará el parámetro.
 - **Columna:** indica la columna en la que se presentará el parámetro.
 - **Nombre:** el nombre del parámetro. Puede asignarse otro parámetro sin necesidad de añadir uno nuevo escribiendo el nombre de un parámetro válido.
 - **Etiqueta:** el texto con el que se presentará el parámetro en el idioma por defecto.
 - **EtiquetaAlt:** el texto con el que se presentará el parámetro cuando la aplicación se lance en un idioma distinto al idioma por defecto.
 - **Origen:** en el caso de parámetros cuyo valor se obtiene de una lista - o tabla de selección - indica el nombre de dicha lista o tabla de selección.
 - **Anchura:** define el ancho del parámetro en el formulario, medido en columnas (una o dos columnas)
 - **ParamFiltro:** en caso de parámetros cuya visibilidad pueda ser dependiente del valor de otro parámetro, se escoge el nombre del parámetro del que depende el parámetro actual, de entre una lista de parámetros de relevancia ERP_ART.
 - **ValorVisible:** en caso de parámetros cuya visibilidad pueda ser dependiente del valor de otro parámetro, se indica el valor del parámetro del que depende el parámetro actual
 - **Eliminar parámetros:** seleccionando un parámetro en la tabla, al hacer clic en el botón eliminar, se elimina el parámetro de la tabla, con lo que no se visualizará dicho parámetro en el formulario de artículo.
- **Ordenar:** ordena los parámetros en función de su fila y su columna asignadas
- **Ancho / Alto:** se definen la altura y la anchura en pixels del formulario al presentarlo en ventanas separadas.
- **Desplegables:** se definen la altura y la anchura de los controles desplegados (ORIGEN <> "") mediante el formato mediante el formato <ANCHURA>;<ALTURA>
- **Ver Formulario:** al hacer clic en el botón 'Ver formulario' se presenta un formulario según lo definido a fin de previsualizar la presentación.

En el caso de presentación de componente, los parámetros incluidos en la tabla serán además las columnas visibles en el árbol TR del formulario de envío de listas de materiales.

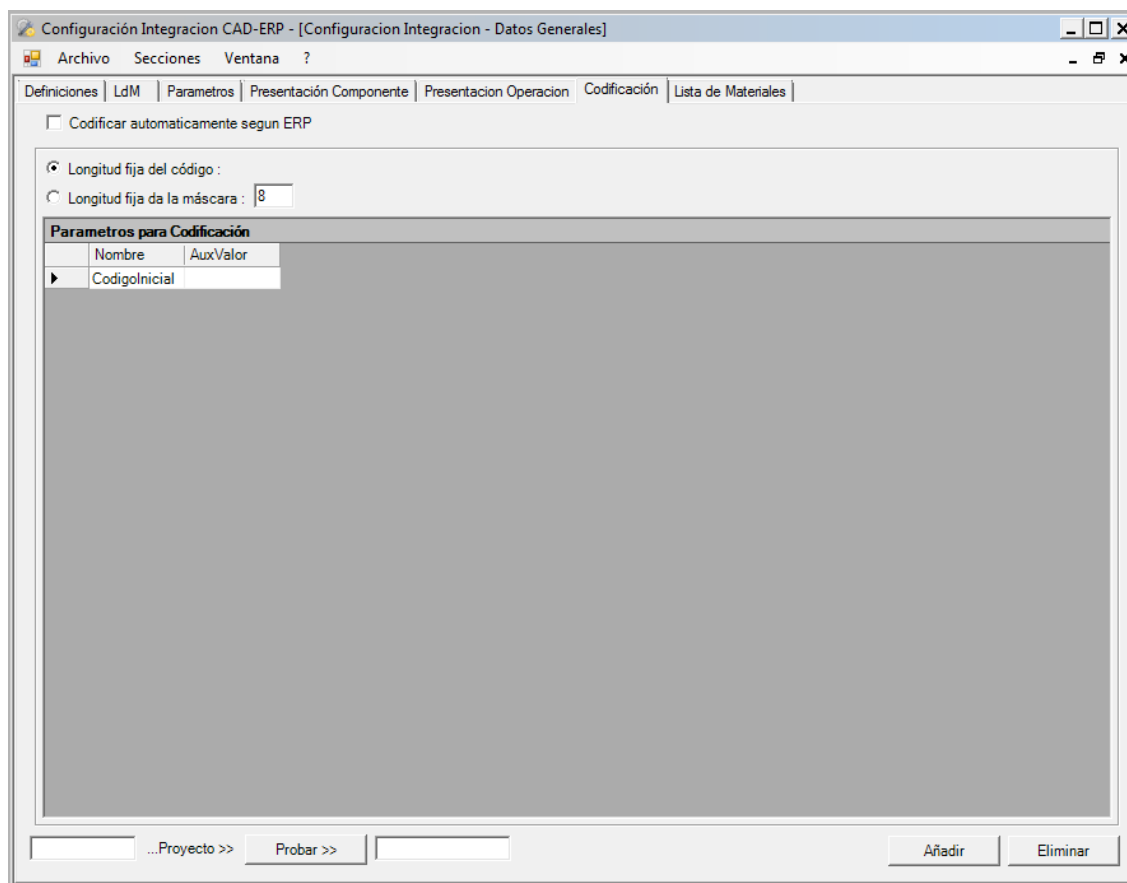
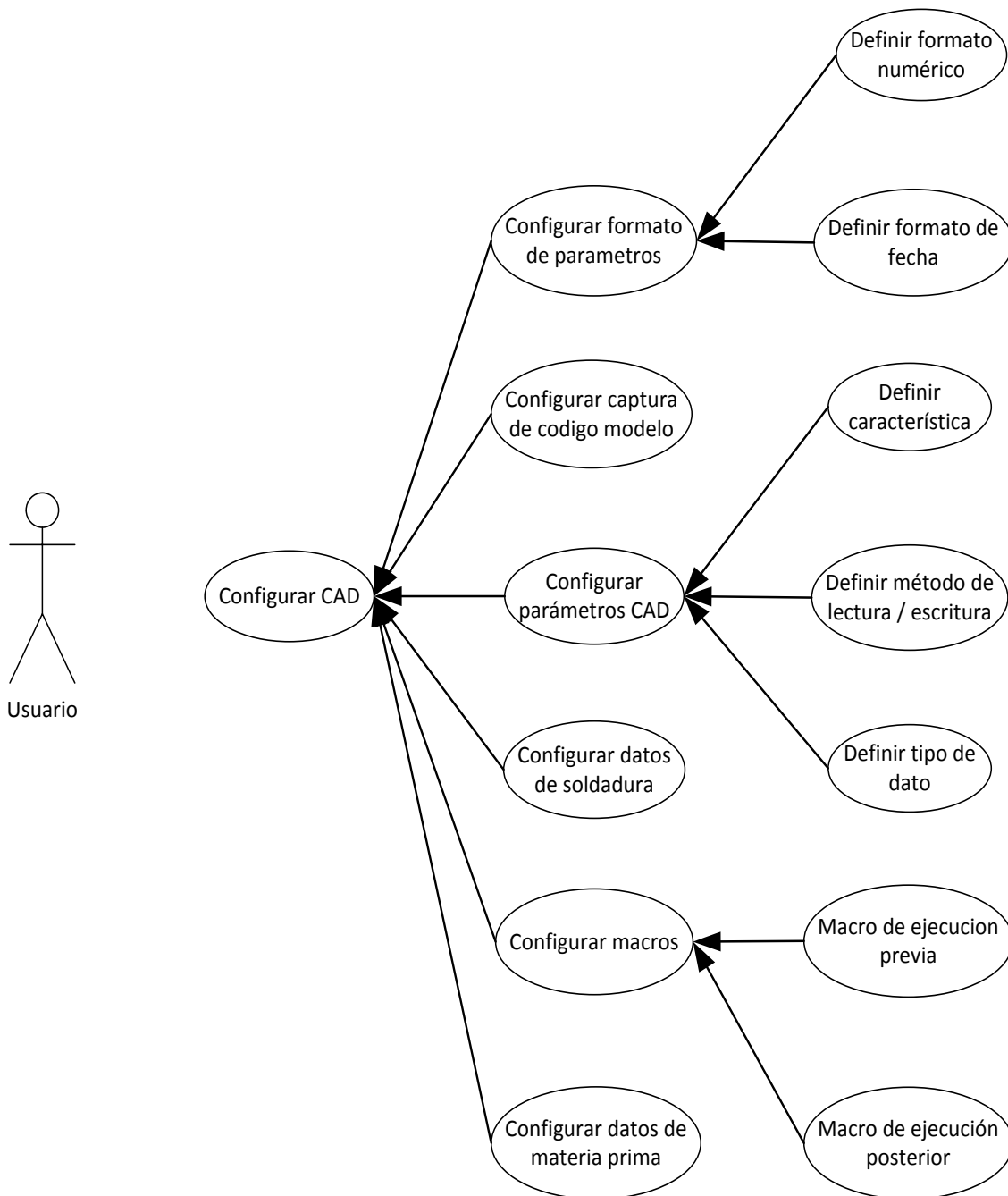


Ilustración 7: Pestaña del apartado de codificación

Configurar codificación: En este apartado se define la forma de codificar los artículos al darlos de alta en el ERP.

- **Codificar automáticamente según ERP:** si se activa esta opción, será el sistema ERP el que codifique el artículo independientemente del código suministrado por la integración. En caso de estar desactivado, el formato de código se configura por medio de los parámetros definidos en la sección inferior.

3.3.2 Configuración CAD



En esta sección se definen los parámetros que intercambian información con la aplicación CAD seleccionada.

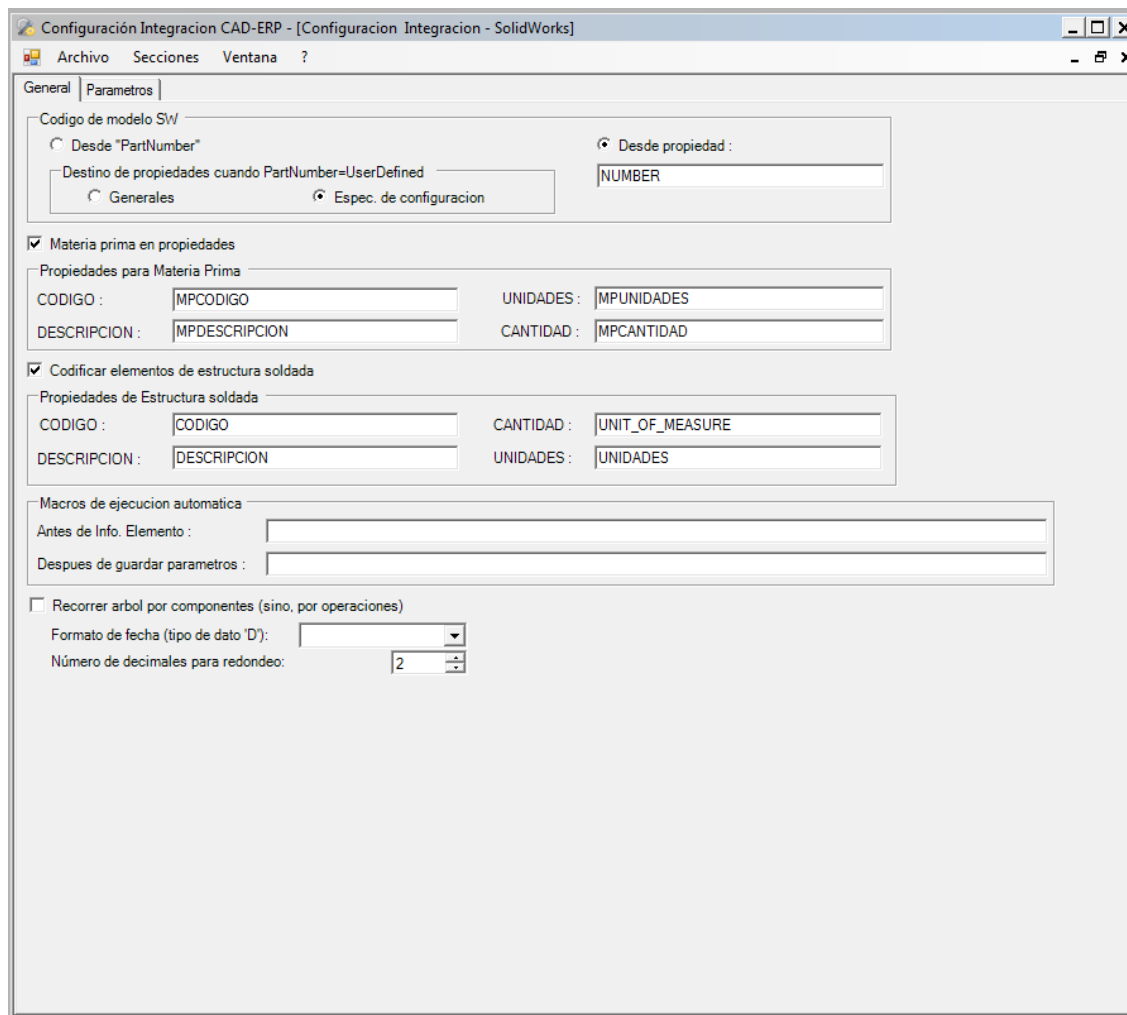


Ilustración 8: Pestaña principal del apartado de configuración CAD

Configurar captura de código modelo: En este apartado se define el modo de lectura / escritura del código de los componentes en CAD:

- Desde 'PartNumber': el código se extrae de la propiedad 'Part number' del modelo.
- Desde propiedad: el código se extrae y graba en las propiedades personalizadas del modelo, con el nombre de propiedad definido.

Configurar datos de materia prima: En este apartado puede definirse si los datos de materia prima se extraen desde las propiedades personalizadas de un modelo, y los nombres de las propiedades asociadas a cada uno de los datos necesarios para gestionar la materia prima:

- Código: código asignado a la materia prima
- Descripción: descripción de la materia prima
- Unidades: Unidad de medida de la materia prima
- Cantidad: Cantidad de materia prima asignada al componente, según la unidad definida.

Configurar datos de soldadura: En este apartado puede definirse si los datos de estructuras soldadas se extraen de entre las propiedades personalizadas de un modelo, y los nombres de las propiedades asociadas a cada uno de los datos necesarios para gestionar soldaduras en caso afirmativo. Las propiedades son análogas a las de la materia prima.

Configurar macros: En este apartado pueden indicarse las rutas donde se ubicarán los archivos de macros que se deseen ejecutar antes y/o después de la ejecución de alguna de las funcionalidades de la integración:

- Macro antes de: Especifica la macro que se ejecutará de forma previa a la ejecución de alguna funcionalidad de la integración.
- Macro después de: Especifica la macro que se ejecutará de forma posterior a la ejecución de alguna funcionalidad de la integración.

Configurar formato de parámetros: En este apartado puede definirse el formato de los parámetros CAD para el caso de que su tipo de dato subyacente sea numérico o de fecha:

- Número de decimales para redondeo: define el número de decimales al que se redondeará el valor de un parámetro cuyo tipo de dato subyacente sea numérico.
- Formato de fecha: define el formato de fecha que se empleará en el tratamiento de parámetros cuyo tipo de dato subyacente sea de tipo fecha.

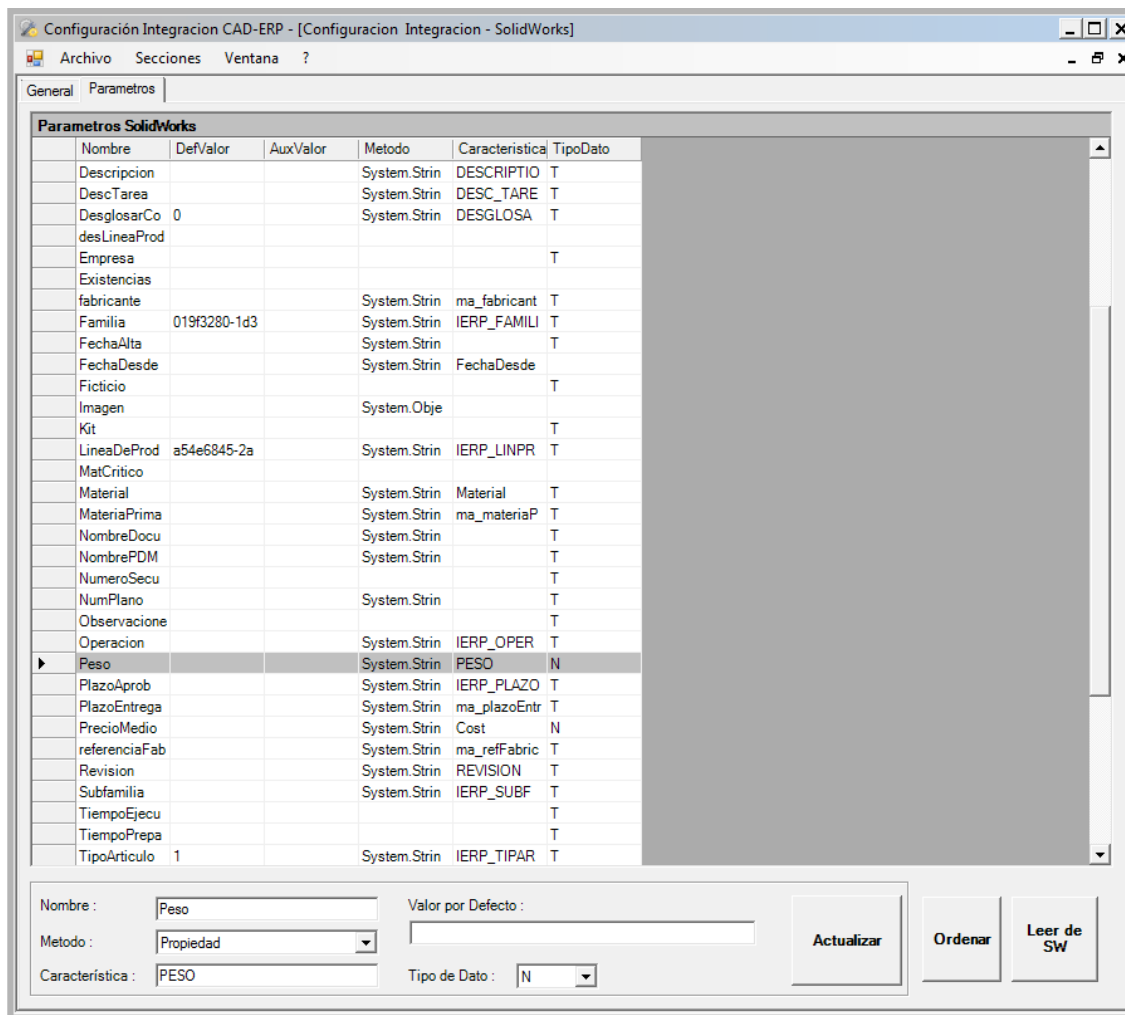
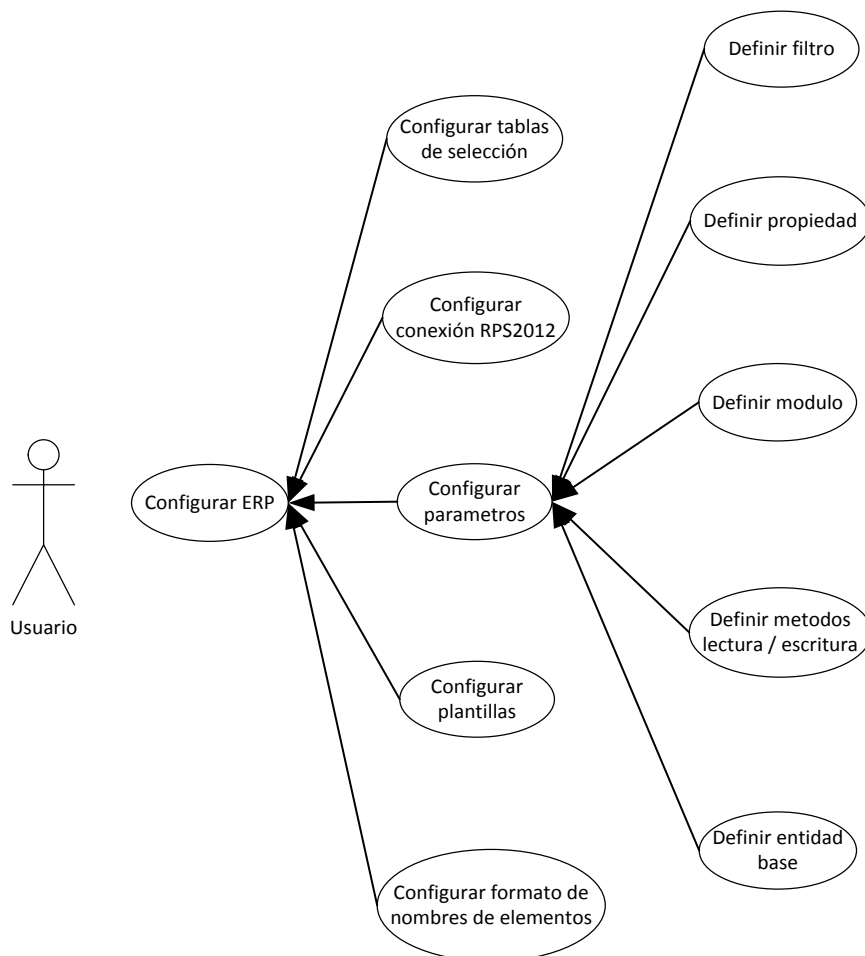


Ilustración 9: Pestaña de parámetros de la sección de configuración CAD

Configurar parámetros CAD: En este apartado se define la manera de leer / escribir cada uno de los parámetros de la integración con relevancia en CAD seleccionándolo en la tabla y editando los siguientes campos:

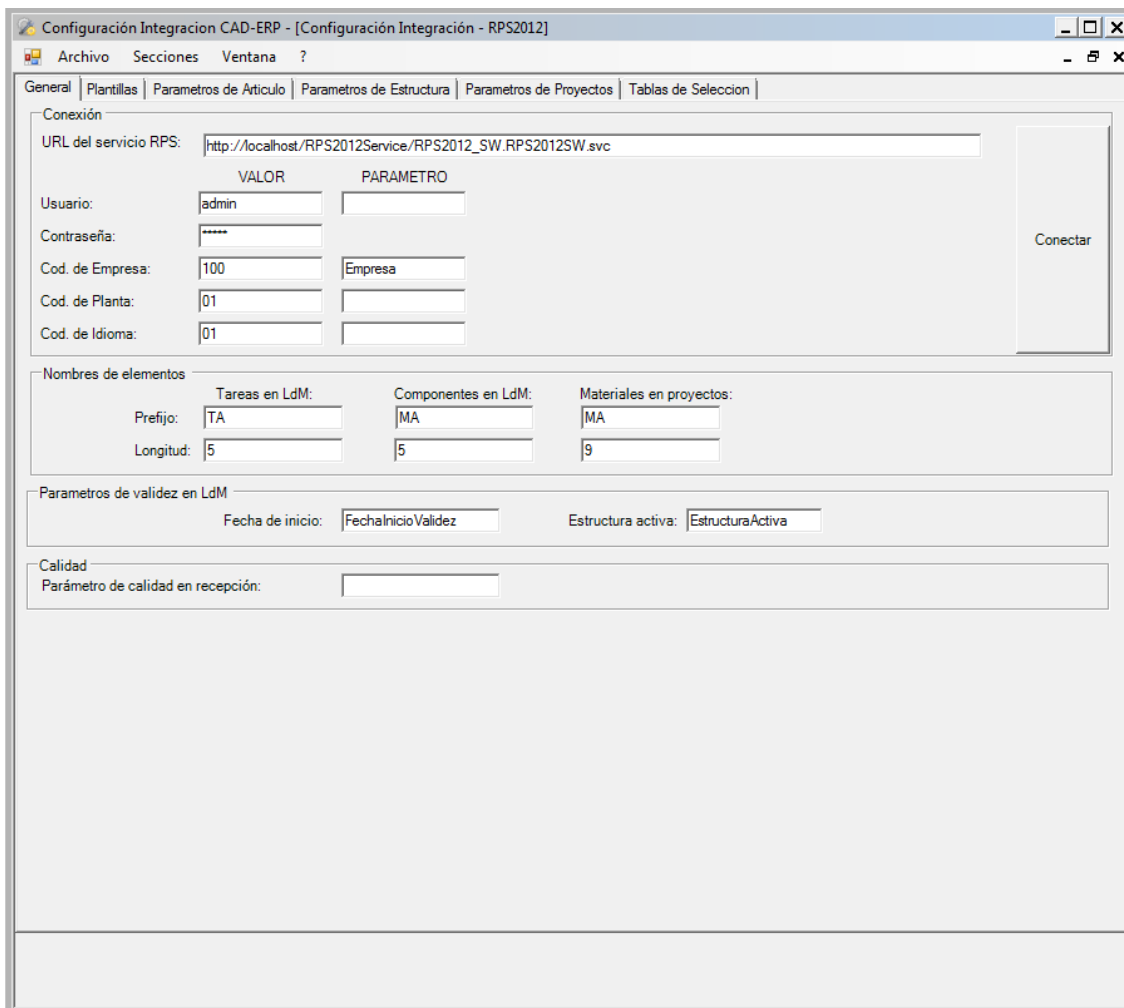
- Valor por defecto: en este campo puede indicarse un valor por defecto para el parámetro en caso de que no sea posible leer el valor correspondiente.
- Método de lectura / escritura: en el campo 'Metodo' puede asignarse alguno de los métodos públicos del módulo iCAD para la extracción / grabación del valor del parámetro correspondiente.
- Característica: puede indicarse alguna característica adicional para la lectura / escritura del valor del parámetro mediante el método, como puede ser el nombre de la propiedad a leer en el caso del método 'Propiedad'
- Tipo de dato: en esta columna se define el tipo de dato subyacente del valor del parámetro, ya que el tipo de dato *de facto* es siempre *String*:
 - T: Tipo de dato textual (String)
 - N: Tipo de dato numérico (Integer, Double)
 - D: Tipo de dato fecha: (Date, DateTime)

3.3.3 Configuración ERP (RPS2012)



En esta sección se definen los parámetros que intercambian información con la aplicación ERP. El adapter de RPS1012 permite el envío de listas de materiales de CAD teniendo como destino estructuras de fabricación o recursos materiales en proyectos. El diálogo con RPS2012 se hace por medio de un conector incluido en el módulo iRPS2012 y que expone las funciones de lectura / escritura definidas.

Este conector dialoga con RPS2012 a través de un servicio exclusivo instalado en el servidor de aplicaciones de RPS2012.



Configuración Integración CAD-ERP - [Configuración Integración - RPS2012]

Archivo Secciones Ventana ?

General | Plantillas | Parametros de Articulo | Parametros de Estructura | Parametros de Proyectos | Tablas de Seleccion

Conexión

URL del servicio RPS:

	VALOR	PARAMETRO
Usuario:	<input type="text" value="admin"/>	<input type="text"/>
Contraseña:	<input type="password" value="*****"/>	<input type="text"/>
Cod. de Empresa:	<input type="text" value="100"/>	<input type="text" value="Empresa"/>
Cod. de Planta:	<input type="text" value="01"/>	<input type="text"/>
Cod. de Idioma:	<input type="text" value="01"/>	<input type="text"/>

Conectar

Nombres de elementos

	Tareas en LdM:	Componentes en LdM:	Materiales en proyectos:
Prefijo:	<input type="text" value="TA"/>	<input type="text" value="MA"/>	<input type="text" value="MA"/>
Longitud:	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="9"/>

Parametros de validez en LdM

Fecha de inicio: Estructura activa:

Calidad

Parámetro de calidad en recepción:

Ilustración 10: Pestaña principal de la sección de configuración ERP (RPS2012)

Configurar conexión RPS2012: En este apartado se definen las credenciales empleadas para la conexión con el sistema RPS2012:

- URL del servicio: dirección en la que está hospedado el servicio. Las consultas a RPS2012 se realizan a través de los contratos de este servicio.
- Usuario: nombre de usuario en RPS2012
- Contraseña: contraseña del usuario en RPS2012
- Cod. de empresa: código de empresa en RPS2012
- Cod. de planta: código de la planta de la empresa sobre la que operar en RPS2012
- Cod. de idioma: código del idioma a emplear en la sesión RPS2012

Configurar formato de nombres de elementos: en este apartado se define el formato de los códigos a la hora de tratar materiales (artículos que forman parte de una lista de materiales) y tareas (operaciones asociadas a rutas en RPS2012):

- Prefijo del elemento
- Longitud del código

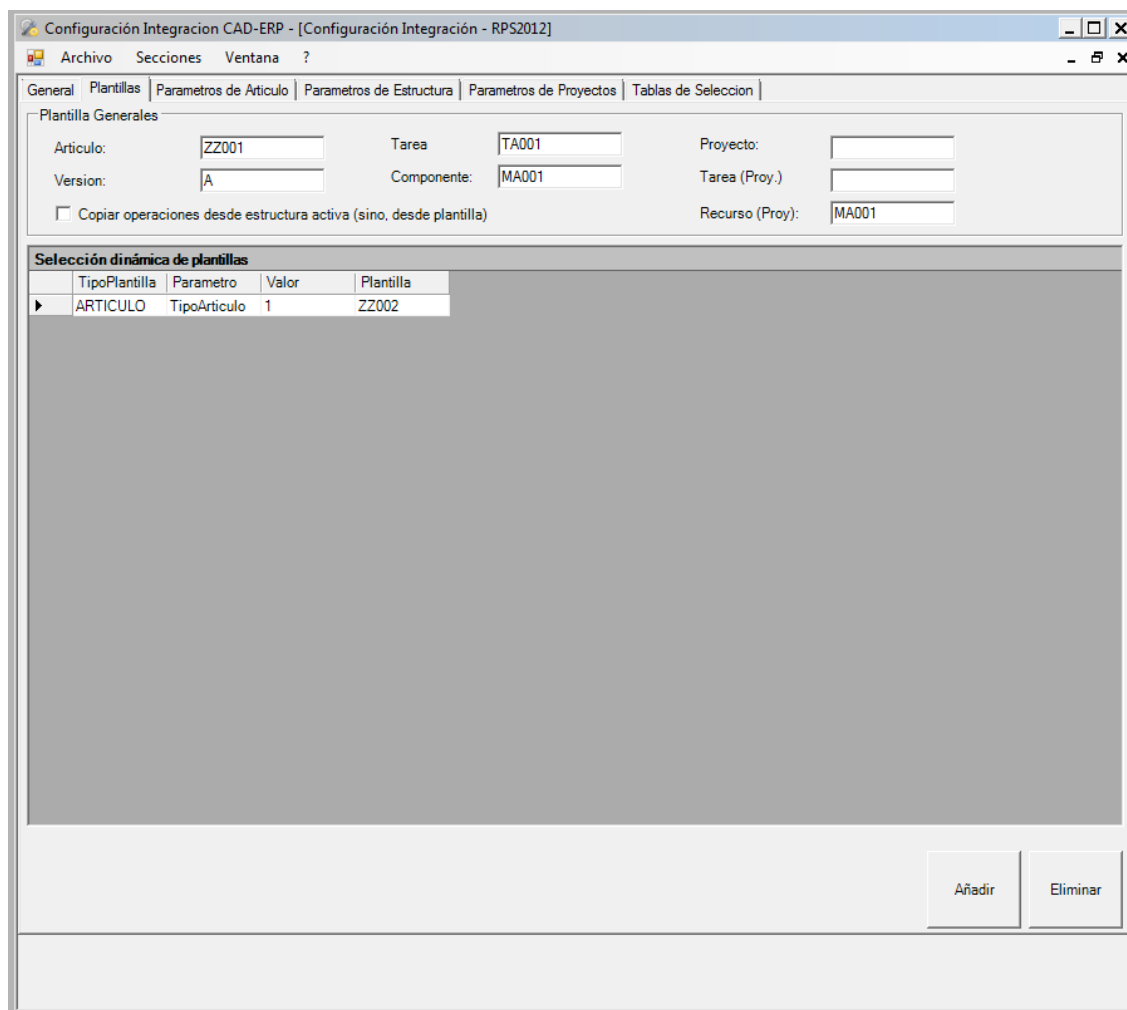


Ilustración 11: Pestaña del apartado de plantillas en configuración RPS2012

Configuración de plantillas: En este apartado se definen las plantillas a emplear en la creación de elementos en RPS2012:

- **Artículo:** El código de la plantilla empleada en la creación y modificación de artículos en RPS2012
- **Tarea:** El código de la plantilla empleada en la creación y modificación de tareas de estructura en RPS2012
- **Componente:** El código de la plantilla empleada en la creación y modificación de materiales de estructura en RPS2012
- **Proyecto:** El código de la plantilla empleada en la creación y modificación de proyectos en RPS2012
- **Tarea (Proy):** El código de la plantilla empleada en la creación y modificación de tareas de proyecto en RPS2012
- **Recurso (Proy):** El código de la plantilla empleada en la creación y modificación de materiales de proyecto en RPS2012

- Versión: El valor por defecto del campo 'Versión' en cualquiera de los ámbitos aplicables (estructura o proyecto)

Configurar parámetros (artículos)

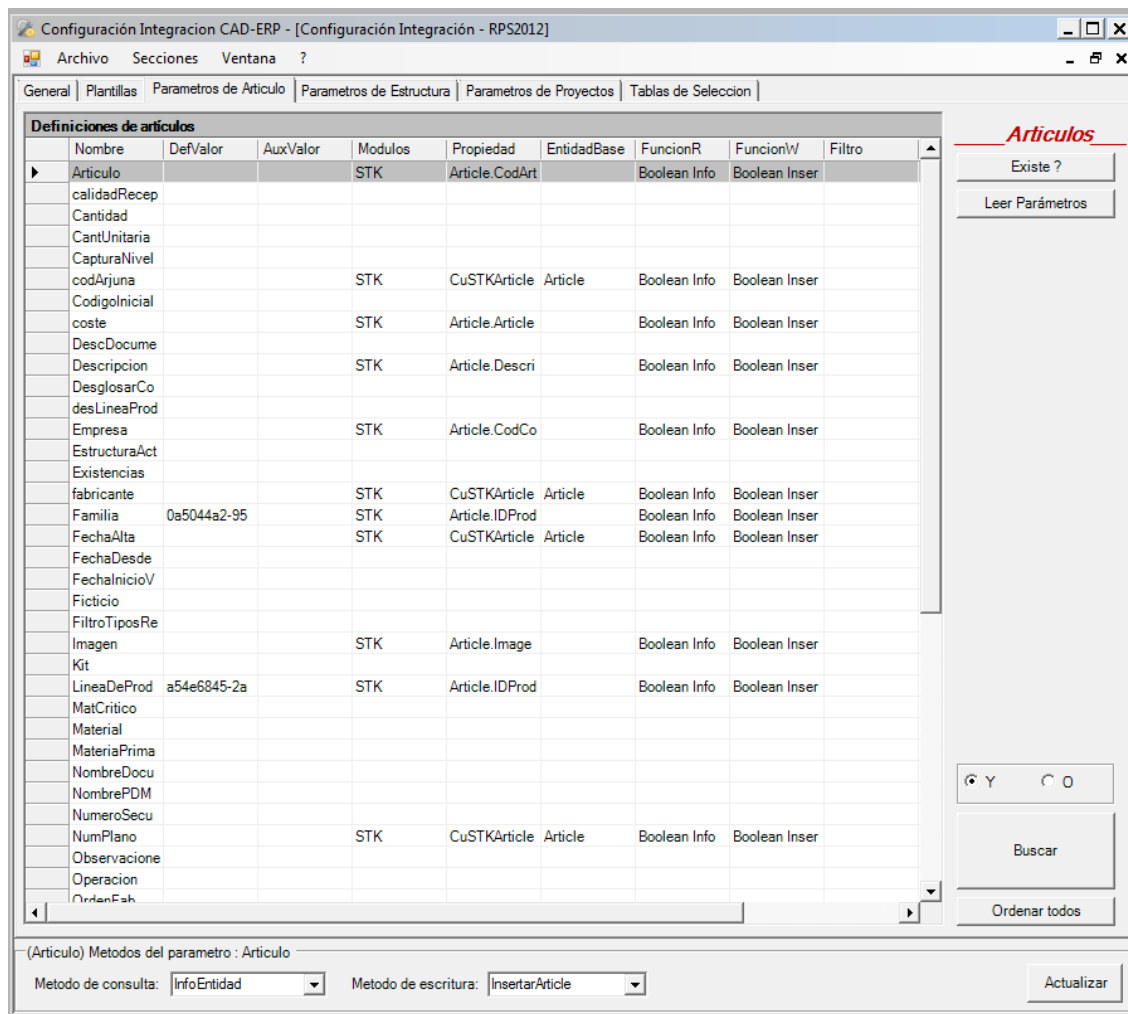


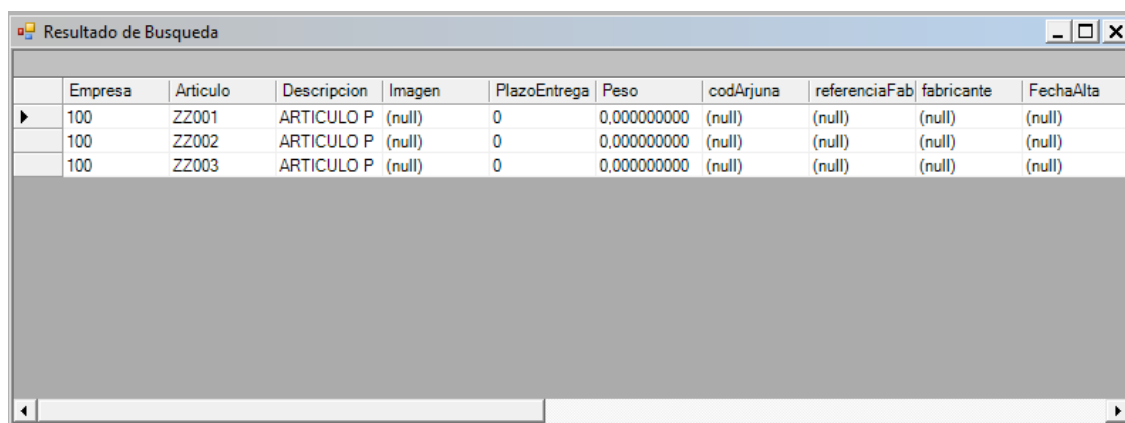
Ilustración 12: Pestaña del apartado de parámetros de artículos

En este apartado se definen las propiedades de entidades en RPS2012 a las que van enlazados los parámetros con relevancia ERP_ART de la integración, junto con sus métodos de lectura y escritura, valores por defecto y filtros:

- Definir módulo: el campo 'Módulo' hace referencia a uno de los ámbitos de funcionalidad de RPS2012, identificado por los 3 caracteres alfabéticos (STK --> Almacén de artículos, PRO --> Proyectos, ...) donde se emplaza la propiedad a la que va enlazada el parámetro.
- Definir propiedad: la columna 'Propiedad' identifica la tabla y la columna de la base de datos (denominados como entidad y propiedad de la entidad, respectivamente) de donde se extrae y se graba el valor del parámetro. Puede tratarse de una propiedad de una entidad extendida creada por un usuario de RPS2012, en cuyo caso hay que indicar en la columna 'Entidad base' el nombre de la entidad (o tabla de la base de datos) de RPS2012 original.

- Definir método de lectura: por medio del desplegable en la parte inferior, puede asignarse al parámetro el método de lectura para extraer su valor. Los métodos se exponen a través del adaptador del servicio que forma parte del módulo de tipo il_ERP iRPS2012.
- Definir método de escritura: por medio del desplegable en la parte inferior, puede asignarse al parámetro el método de escritura para grabar su valor. Los métodos se exponen a través del adaptador del servicio que forma parte del módulo de tipo il_ERP iRPS2012.
- Filtro: En caso de que se deseen extraer propiedades repetidas, se debe definir un filtro adecuado en el parámetro para desglosar la llamada al método correspondiente en varias llamadas separadas, extrayendo el valor pertinente de la propiedad en cada una. El formato a emplear en la generación de filtros es el siguiente:
 - <Propiedad>=<Parametro> Cuando el valor de la propiedad es dependiente del valor de otro parámetro distinto
 - <Propiedad>=<Valor> Cuando el valor de la propiedad depende de un valor fijo predefinido

Pueden realizarse consultas en RPS2012 mediante el botón 'Buscar', proporcionando valores de búsqueda en la columna 'AuxValor'. Los resultados se presentan en una tabla contenida en un formulario separado.



Empresa	Artículo	Descripción	Imagen	PlazoEntrega	Peso	codArjuna	referenciaFab	fabricante	FechaAlta
100	ZZ001	ARTICULO P	(null)	0	0,000000000	(null)	(null)	(null)	(null)
100	ZZ002	ARTICULO P	(null)	0	0,000000000	(null)	(null)	(null)	(null)
100	ZZ003	ARTICULO P	(null)	0	0,000000000	(null)	(null)	(null)	(null)

Los parámetros del resto de relevancia - aquellos con relevancia ERP_LDM, ERP_RUTA o ERP_PROY - se definen de forma análoga a aquellos con relevancia ERP_ART, mediante esta misma metodología

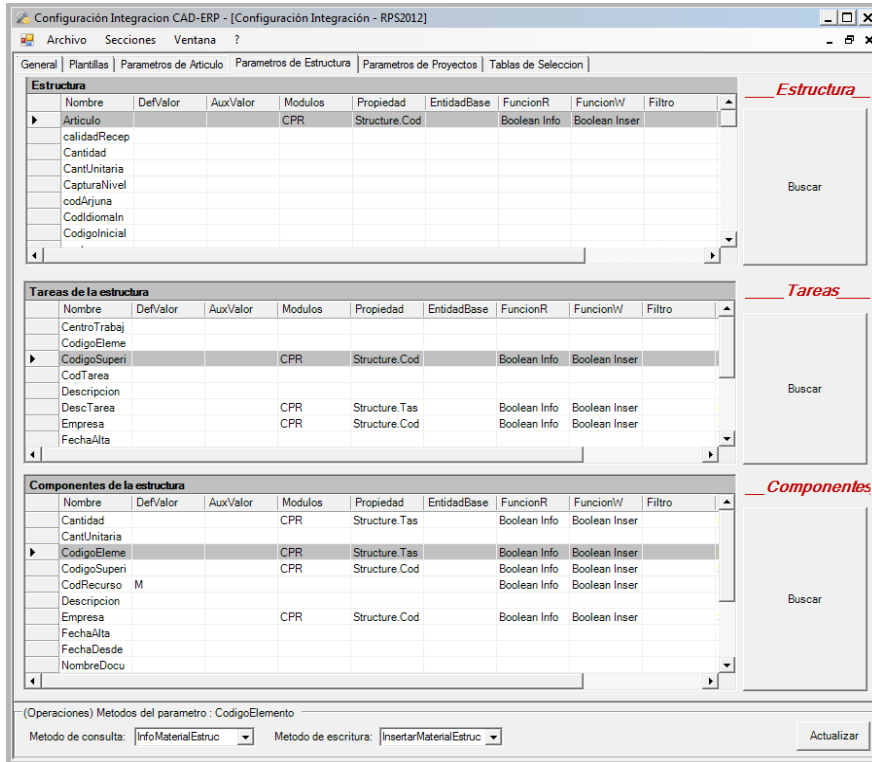


Ilustración 13: Pestaña del apartado de parámetros de estructura

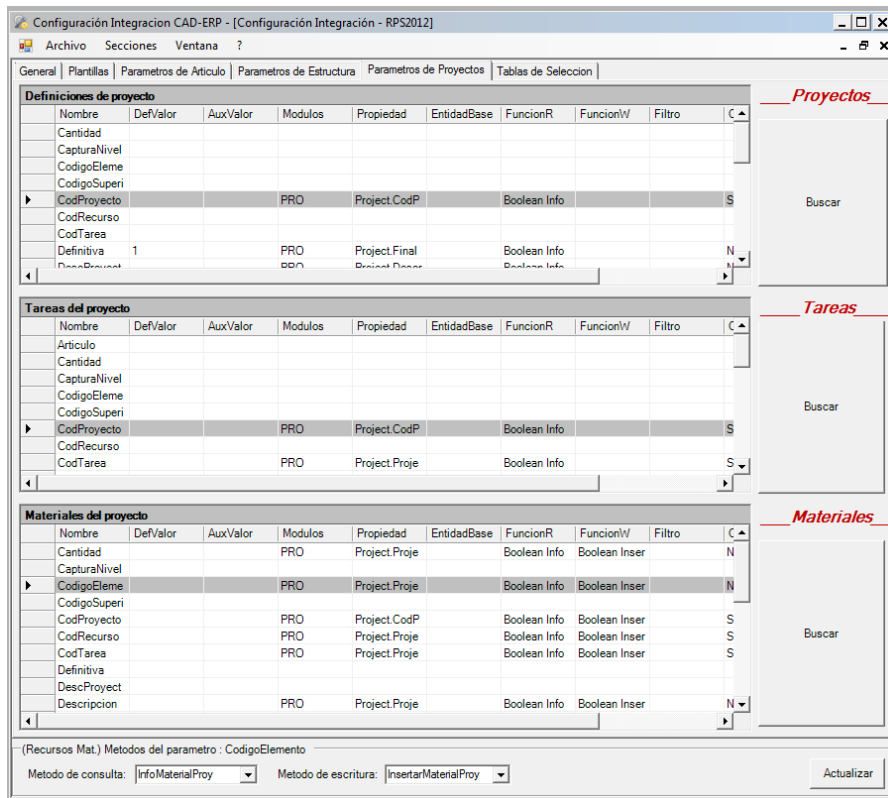


Ilustración 14: Pestaña del apartado de parámetros de proyectos

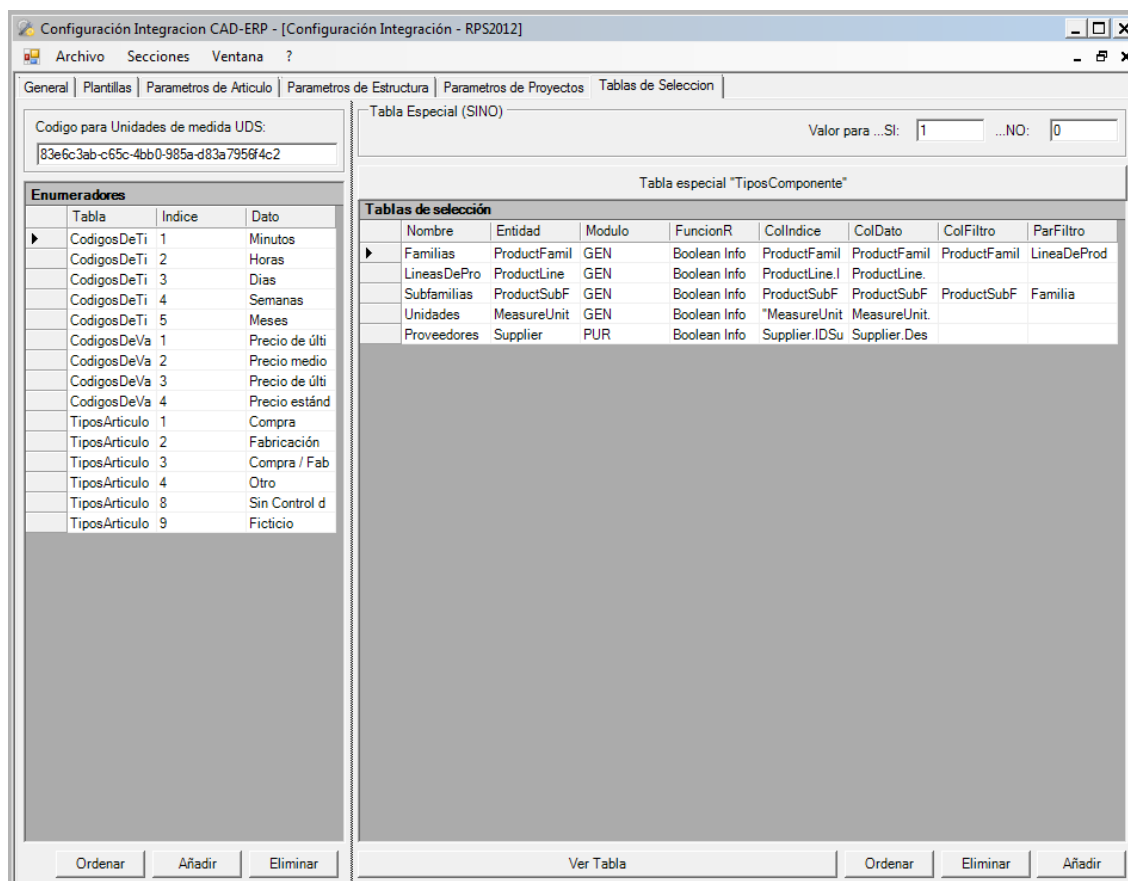
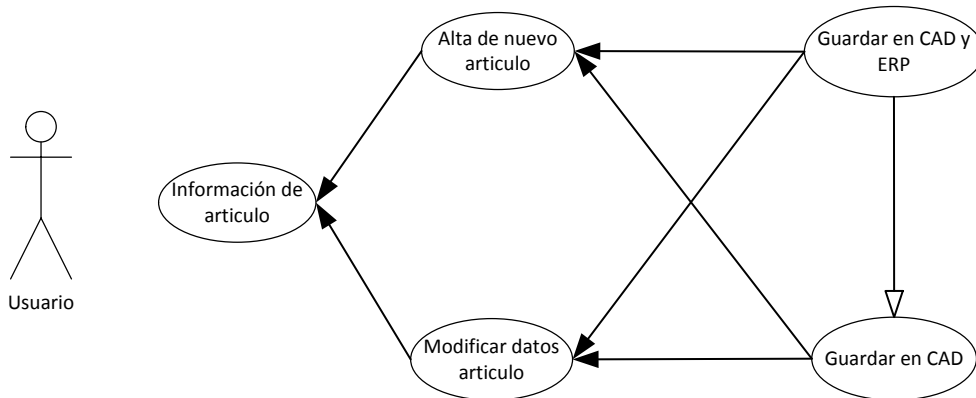


Ilustración 15: Pestaña del apartado de tablas de selección

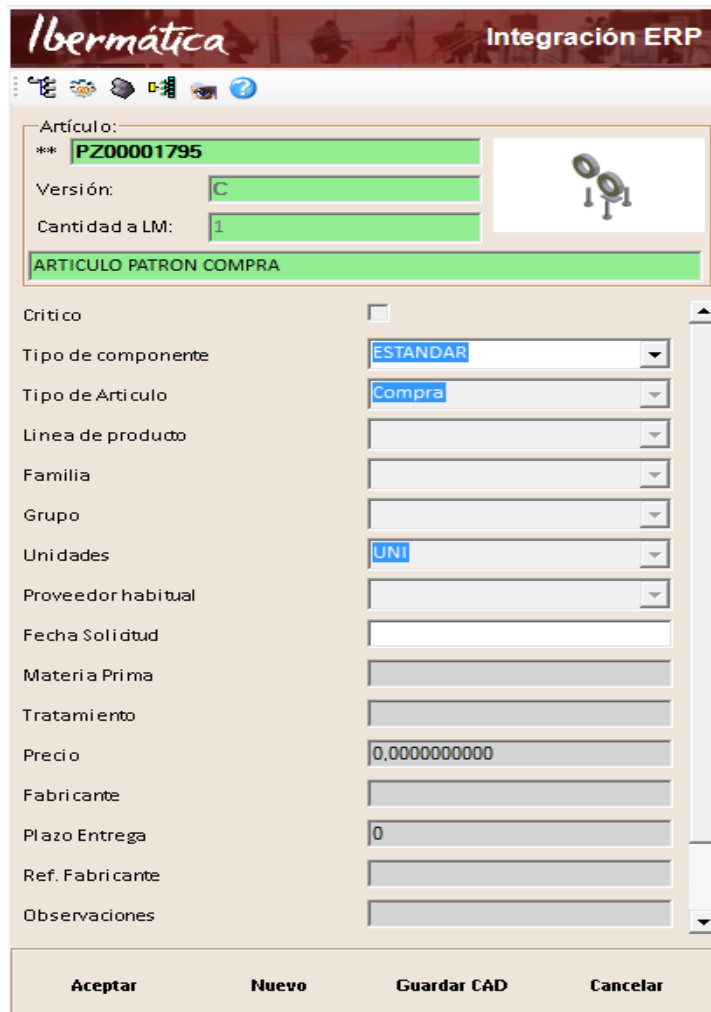
Configurar tablas de selección: en este apartado se configuran las tablas de selección y enumeradores de la integración para los parámetros que se alimentan de una lista de posibles valores. Los valores se asignan por medio de desplegable a 2 columnas presentados en los distintos formularios de la integración; estos desplegable cuentan con una columna con el valor que alimenta el parámetro, y otro valor descriptivo que se presenta para facilitar la lectura y selección de valores por parte del usuario.

- Enumeradores: se definen los valores de un enumerador en la tabla indicando el nombre del enumerador junto con el valor índice (valor que alimenta el parámetro) y el valor dato (valor descriptivo para facilitar la selección.)
- Tablas de selección: se definen de forma análoga a los parámetros de RPS2012, enlazando los valores a propiedades de entidades en RPS2012, mediante la metodología de asignación de módulos, entidades y métodos. El valor índice se enlaza a alguna propiedad compatible de la entidad (código, ID, ...) y el valor dato se enlaza a otra propiedad de naturaleza descriptiva (Nombre, abreviatura, descripción, ...). Adicionalmente, pueden definirse filtros para las tablas, de modo que los valores dependan del valor de otro parámetro; la columna ColFiltro indica la propiedad de RPS2012 empleada a modo de filtro, y la columna ParFiltro indica el parámetro de la integración cuyo valor se emplea en el filtrado.

3.3.4 Información de artículo



Este es el proceso que se realiza cada vez que se quiere dar de alta un nuevo artículo en el sistema ERP. También se utiliza para modificar los valores que previamente se han adjudicado a un determinado artículo.



Ibermática Integración ERP

Artículo: ** PZ00001795

Versión: C

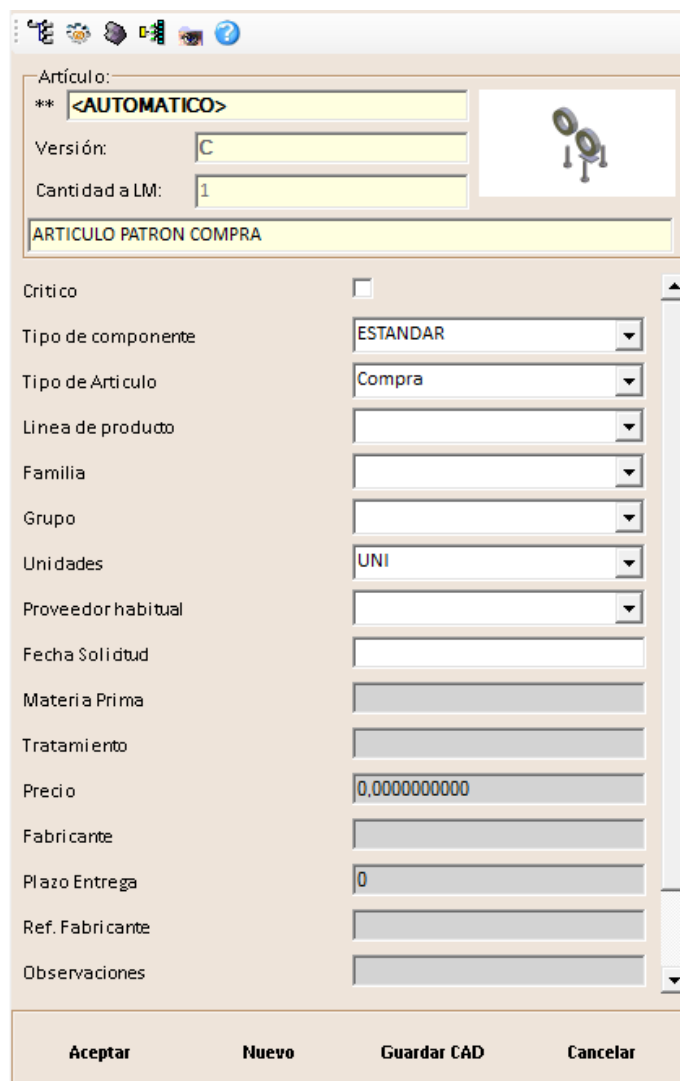
Cantidad a LM: 1

ARTICULO PATRON COMPRA

Crítico
 Tipo de componente: ESTANDAR
 Tipo de Artículo: Compra
 Línea de producto:
 Familia:
 Grupo:
 Unidades: UNI
 Proveedor habitual:
 Fecha Solicitud:
 Materia Prima:
 Tratamiento:
 Precio: 0,0000000000
 Fabricante:
 Plazo Entrega: 0
 Ref. Fabricante:
 Observaciones:

Ilustración 16: Formulario de información de artículo

Alta de nuevo artículo: En caso de que el componente seleccionado en el programa CAD no esté dado de alta en el sistema ERP, se mostrará el formulario de información de artículo con el código automático calculado por la integración, y los parámetros extraídos exclusivamente desde CAD; los datos clave (Código, Versión, Cantidad a LdM y Descripción) se muestran en fondo blanco. El mismo formulario se muestra para un artículo ya codificado en el ERP al pulsar el botón 'Nuevo'.



Artículo: ** <AUTOMATICO>
 Versión: C
 Cantidad a LM: 1
 ARTICULO PATRON COMPRA

Critico
 Tipo de componente: ESTANDAR
 Tipo de Artículo: Compra
 Línea de producto:
 Familia:
 Grupo:
 Unidades: UNI
 Proveedor habitual:
 Fecha Solicitud:
 Materia Prima:
 Tratamiento:
 Precio: 0,0000000000
 Fabricante:
 Plazo Entrega: 0
 Ref. Fabricante:
 Observaciones:

Aceptar Nuevo Guardar CAD Cancelar

Ilustración 17: Formulario de información de nuevo artículo

El usuario puede modificar los valores de los parámetros presentados en el formulario escribiendo los valores o seleccionando del desplegable en el caso de parámetros asociados a tablas de selección. Los parámetros que se presentan son los configurados según el caso de uso 'Configurar presentación'; aquellos parámetros con relevancia de lectura (R o S) no podrán ser editados, y se presentan en gris, únicamente para su visualización

Además de los valores de los parámetros, puede asignarse al componente CAD el código de un artículo previamente registrado en el ERP, tecleando su código en el campo correspondiente, o realizando una búsqueda de artículos en el ERP (doble clic en el campo de código)

Integración CAD-ERP

Ibermática Busqueda de artículo

Empresa	Artículo	Descripcion	Imagen	PlazoEntrega	Peso	codArjuna	referenciaFab
100	PZ00001507	MORDAZA DELANTERA	(null)	0	0.000000000	(null)	(null)
100	PZ00001153	PERFIL BASE 22 ESTACIONES	Matriz Byte[]	0	0.000000000	(null)	(null)
100	PZ00000816	LARGERO	Matriz Byte[]	0	0.000000000	(null)	(null)
100	PZ00001487	PUNTERA	(null)	0	0.000000000	(null)	(null)
100	MA00000035	BARRERA DE SEGURIDAD (EMISOR)	(null)	0	0.000000000	2060-99049	1022230
100	MA00000057	BARRERA DE SEGURIDAD (RECEPTOR)	(null)	0	0.000000000	2060-9021612	1027480
100	MA00000993	CUERPO LED AMARILLO	(null)	0	0.000000000	2040-80058	ZB4-Bv10B55
100	MA00000644	ARANDELA GROWER DIN 127 M8	Matriz Byte[]	0	0.000000000	7015-53287	(null)
100	PZ00001663	SOPORTE RODAMIENTO INFERIOR	Matriz Byte[]	0	0.000000000	(null)	(null)
100	MA00000190	Caracter TEE AR1-MB01-P	(null)	0	0.000000000	7056-54208	AR1-MB01-P
100	MA00000925	PASADOR CILINDRICO CON ROSCA INT	Matriz Byte[]	0	0.000000000	7013-55810	(null)
100	MA00000175	Caracter TEE AR1-MA01-2	(null)	0	0.000000000	7056-53747	AR1-MA01-2
100	ES00000121	GASTO GENERICO EXENTO FACTURA	(null)	0	0.000000000	(null)	(null)
100	PZ00001064	TUBO BOTONERA	(null)	0	0.000000000	(null)	(null)
100	PZ00001065	ARTICULACION BOTONERA	(null)	0	0.000000000	(null)	(null)
100	PZ00001723	EJE VERTICAL	Matriz Byte[]	0	0.000000000	AI-208-02	(null)
100	PZ00000758	DISTANCIADOR PERFILES BASE	Matriz Byte[]	0	0.000000000	(null)	(null)
100	PZ00000795	UÑA ANCLAJE FIJO TRASERO	Matriz Byte[]	0	0.000000000	(null)	(null)
100	MA00000670	TUERCA HEXAGONAL M14 DIN 934	Matriz Byte[]	0	0.020000000	7011-53315	(null)
100	MA00000839	Tuerca PG13 metalico	(null)	0	0.000000000	7053-57081	(null)
100	MA00000837	Tuerca PG11 metalico	(null)	0	0.000000000	7053-57080	(null)
100	MA00000037	BARRERA DE SEGURIDAD (EMISOR)	(null)	0	0.000000000	2060-95192	1022238
100	MA00000664	TUERCA HEXAGONAL M4 DIN 934	Matriz Byte[]	0	0.000000000	7011-53307	(null)
100	MA00000647	ARANDELA GROWER DIN 127 M14	Matriz Byte[]	0	0.000000000	7015-53290	(null)
100	PZ00001173	CINTAS TRANSFER FLUJO LIBRE LAZPI	Matriz Byte[]	0	0.000000000	(null)	(null)
100	MA00000677	TUERCA CAJEAADA R10 M6	Matriz Byte[]	0	0.000000000	7011-56184	3842 530 285

Buscar Hecho Criterio de busqueda en campos múltiples: Y O Limpiar Cancelar

Ilustración 18: Formulario MatchCode para búsqueda en ERP

Al asignar código de esta manera, los datos del artículo seleccionado se copiarán sobre los datos del nuevo artículo, según la relevancia establecida en cada parámetro.

Artículo: **MA00000035**

Versión: **C**

Cantidad a LM: **1**

BARRERA DE SEGURIDAD (EMISOR)


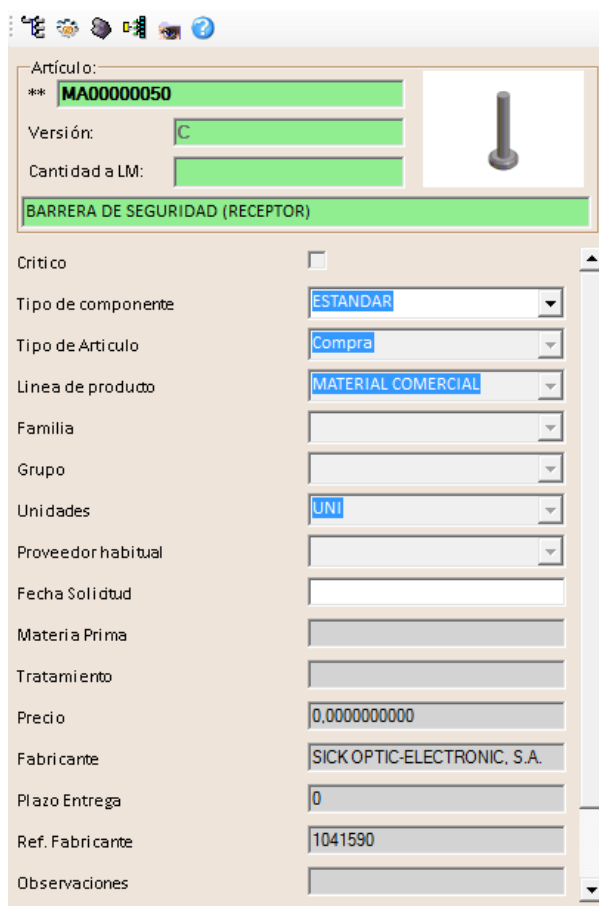


Ilustración 19: Código asignado desde el formulario de búsqueda

Modificar datos artículos: Si el artículo seleccionado ya se encuentra dado de alta en el sistema ERP, se mostrarán su código y datos asociados importados desde el ERP o el programa CAD, según la relevancia establecida de cada parámetro; los datos clave (Código, Versión, Cantidad a LdM y Descripción) se muestran en fondo verde.



Artículo: ** MA00000050
 Versión: C
 Cantidad a LM:
 BARRERA DE SEGURIDAD (RECEPTOR)

Critico
 Tipo de componente: ESTANDAR
 Tipo de Artículo: Compra
 Línea de producto: MATERIAL COMERCIAL
 Familia:
 Grupo:
 Unidades: UNI
 Proveedor habitual:
 Fecha Solicitud:
 Materia Prima:
 Tratamiento:
 Precio: 0,0000000000
 Fabricante: SICK OPTIC-ELECTRONIC, S.A.
 Plazo Entrega: 0
 Ref. Fabricante: 1041590
 Observaciones:

Ilustración 20: Formulario de información de artículo registrado en ERP

De forma análoga al caso de nuevos artículos, el usuario puede modificar los valores de los parámetros presentados en el formulario, editando los valores de los parámetros en función de la configuración de cada parámetro, o asignando el código y datos de un artículo ya existente en el ERP.

Guardar datos en CAD: Al pulsar el botón 'Guardar CAD' se graba el valor de los parámetros del artículo con relevancia R (sólo si se trata de un nuevo artículo) o W en el ámbito CAD, según el método de escritura establecido para cada parámetro según el caso de uso 'Configuración de parámetros CAD'.

Guardar datos en CAD y ERP: Al pulsar el botón 'Aceptar', se guardan los datos en CAD según el caso de uso 'Guardar datos en CAD', y a continuación se procede a grabar los datos del artículo correspondiente en el ERP, generando un nuevo artículo sobre el que se guardan los datos en el caso de 'Alta de nuevo artículo'.

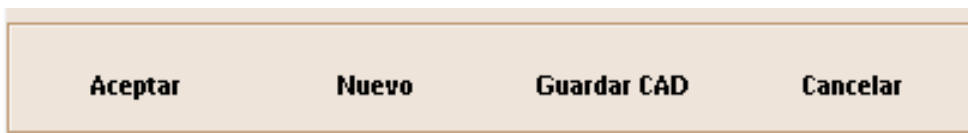


Ilustración 21: Botones del formulario de información de artículo

En caso de que el elemento seleccionado en el programa CAD no sea el documento activo, sino uno de sus componentes, no se podrán guardar los datos en ERP, tan sólo en CAD. En este caso, no se mostrarán el botón 'Aceptar' y 'Nuevo'

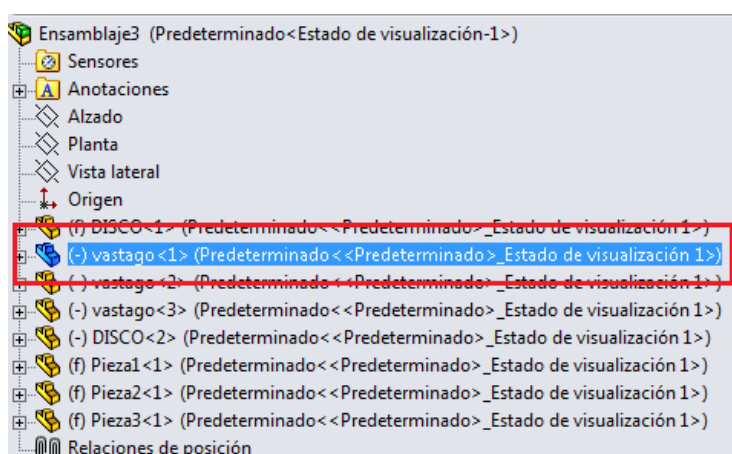


Ilustración 22: Componente seleccionado en SolidWorks

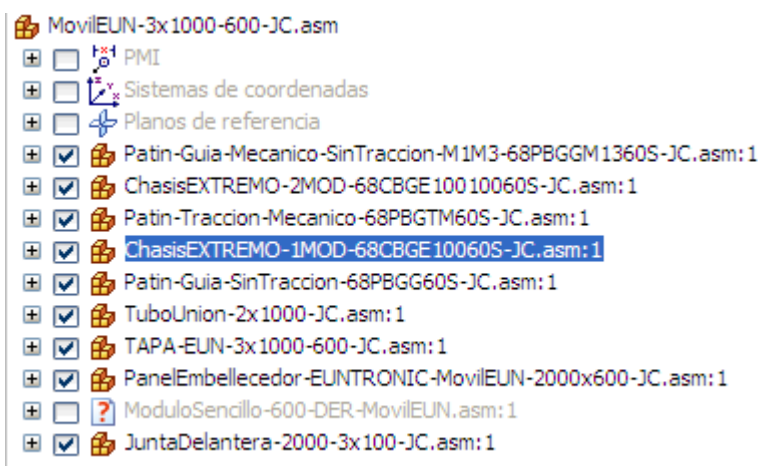
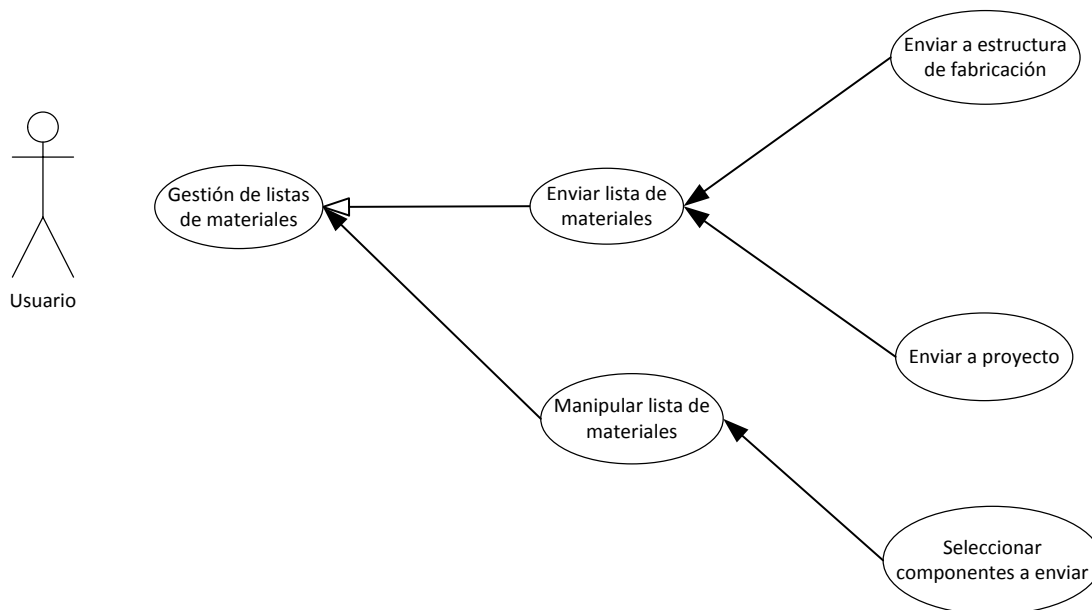


Ilustración 23: Componente seleccionado en Solid Edge

3.3.5 Enviar lista de materiales



Mediante este proceso se revisan todos los artículos que componen un conjunto y se da de alta una nueva lista de materiales en el ERP. Dependiendo de la configuración, se permite lanzar la misma lista varias veces; el sistema alertará sobre las variaciones que contiene la lista respecto al último envío, informando sobre si un componente esta dado de alta o no.

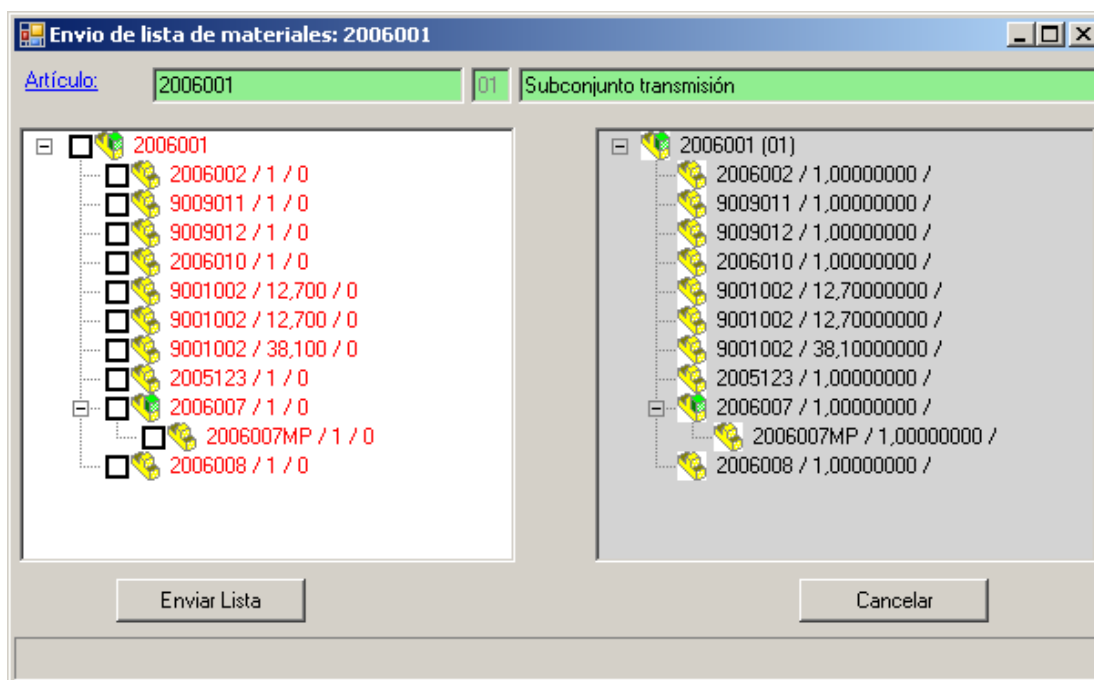


Ilustración 24: Formulario de envío de listas de materiales

Gestión de listas de materiales: al lanzar el formulario de envío de listas de materiales, se extrae la estructura de componentes del documento activo en el programa CAD, representando en cada nodo el código del componente, cantidad y código de operación de ruta por defecto. En la cabecera del formulario se muestra el código, la versión y una descripción del conjunto principal asociado a la lista de materiales.

En el caso de generar una propuesta para una lista de materiales previamente enviada, se muestra en el árbol de la izquierda la estructura de la lista según está dada de alta en el ERP, representando en cada nodo el código del componente, cantidad y código de operación de ruta asignado al componente.

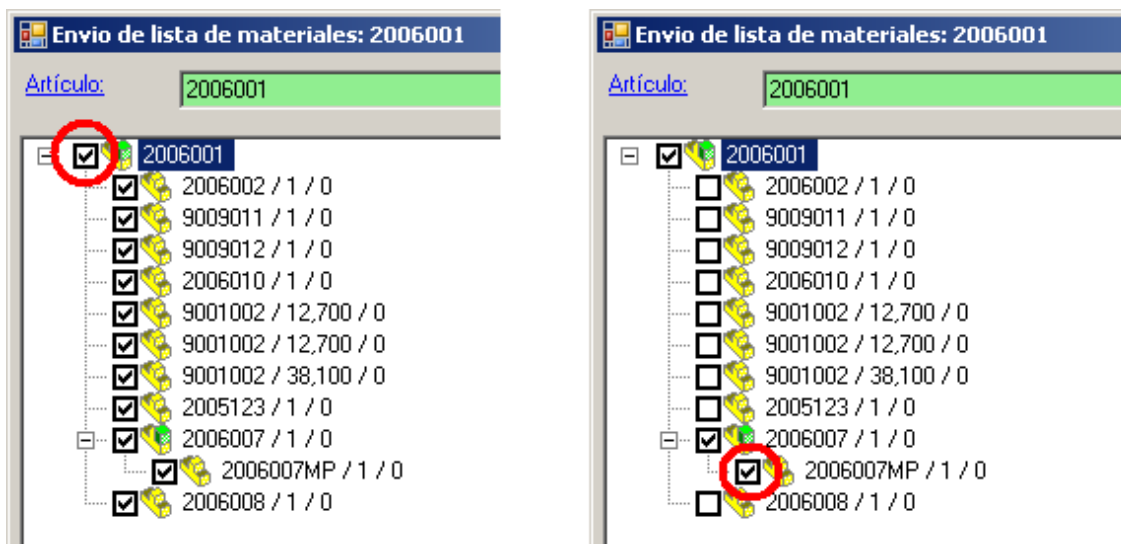


Ilustración 25: Selección de componentes a enviar

Manipular lista de materiales: el árbol CAD permite, por cada componente, incluirlo o excluirlo de la propuesta de envío, propagándose la inclusión / exclusión a los componentes hijos en caso de marcar un componente con subestructura.

Envío a estructuras: Al pulsar el botón 'Aceptar', en el caso de que el conjunto principal esté identificado como un artículo por la integración, se generará una nueva estructura de fabricación en el ERP, cuya lista de materiales estará formada por los componentes extraídos desde el programa CAD, con las cantidades especificadas por el usuario. La gestión de las rutas la efectúa el sistema ERP.

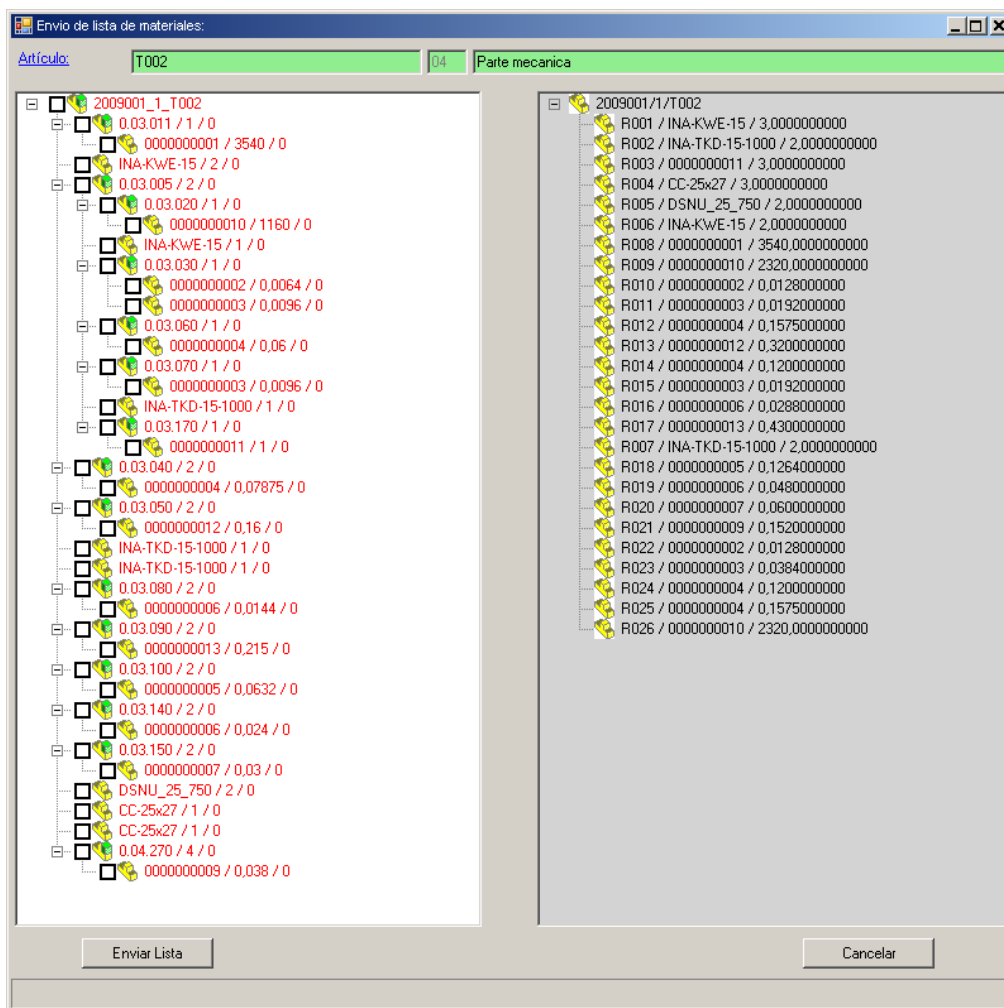


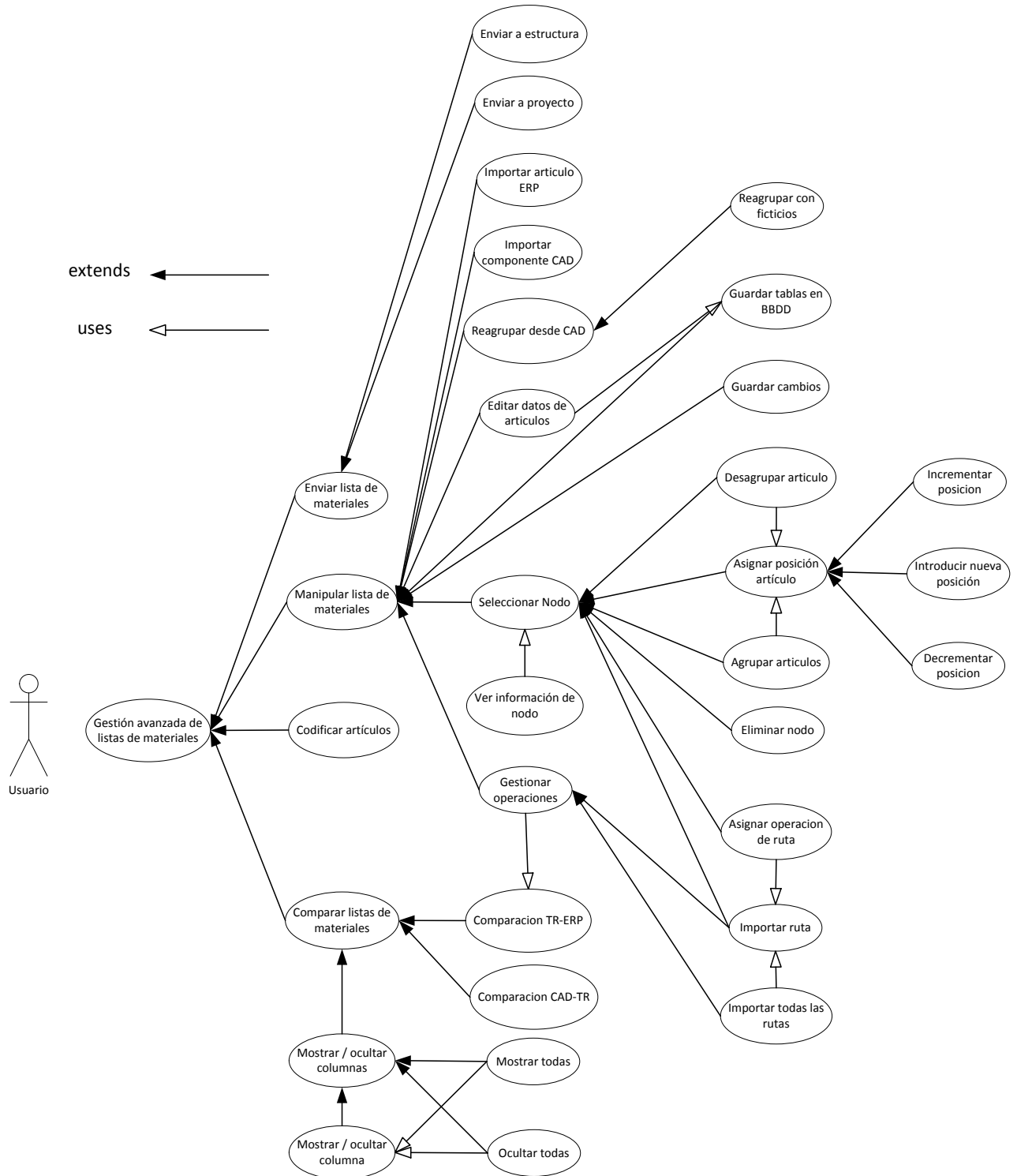
Ilustración 26: Envío de lista de materiales de tarea de proyecto

Envío a proyectos: Al pulsar el botón 'Aceptar', en caso de que el conjunto principal esté identificado como tarea de versión de proyecto por la integración, se añadirán los componentes extraídos desde el programa CAD a la lista de materiales del proyecto correspondiente en ERP. La estructura arborescente extraída desde CAD se "aplana", estando todos los materiales al mismo nivel.

4. Diseño

4.1 Gestión avanzada de listas de materiales

Extiende el caso de uso gestión de listas de materiales



Descripción de los casos de uso:

4.1.1 Reagrupar desde CAD:

El evento se lanza la primera vez que se va realizar un envío LdM de un conjunto, y al hacer clic en el botón Reagrupar (btnReagrupar_Click).

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación; hay datos en el árbol CAD
- Flujo de eventos:
 1. El usuario lanza el caso de uso Crear propuesta de envío LdM para un ensamblaje por primera vez.
 2. El formulario y sus controles se inicializan; árbol CAD muestra la estructura según se refleja en el programa CAD.
 3. El árbol de transformación se carga mediante una agrupación del árbol CAD.
 4. *El usuario manipula la propuesta de LdM en el árbol TR.
 5. *El usuario hace clic en el botón Reagrupar
 6. *El árbol de transformación se carga mediante una agrupación del árbol CAD
- Postcondiciones: El árbol de transformación muestra todos los componentes del árbol CAD agrupados según el criterio configurado.

4.1.2 Importar componente CAD:

Transfiere el componente seleccionado del árbol CAD al árbol de transformación.

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación; hay datos en el árbol CAD
- Flujo de eventos:
 1. El usuario selecciona un componente en el árbol CAD
 2. El usuario hace clic en el botón Pasar Elemento
 3. El componente se añade a la propuesta de envío en el árbol de transformación
- Postcondiciones: El árbol de transformación incluye el componente seleccionado en el árbol CAD

4.1.3 Importar material desde ERP:

Lanza un formulario de búsqueda para insertar un nuevo componente en el árbol de transformación.

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación
- Flujo de eventos:
 1. El usuario selecciona el elemento bajo el cual insertar el nuevo artículo.
 2. El usuario hace clic en el botón Insertar Elemento
 3. Se muestra el formulario de búsqueda de artículos MatchCode
 4. El usuario realiza una búsqueda de artículo en el formulario MatchCode
 5. El usuario selecciona un artículo en la tabla de resultados.
 6. El usuario pulsa el botón Aceptar del formulario MatchCode
 7. Se muestra un formulario iMsgBox preguntando al usuario la cantidad del artículo a insertar

8. El usuario introduce la cantidad y pulsa el botón Aceptar
 9. Se inserta el artículo buscado con la cantidad introducida bajo el componente seleccionado en el árbol de transformación
- Postcondiciones: El árbol de transformación incluye el artículo buscado bajo el componente seleccionado en la primera posición libre

4.1.4 Editar datos de artículos:

Modifica los datos editables de los artículos representados por los componentes del árbol de transformación en los ámbitos CAD y ERP

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación
- Flujo de eventos:
 1. El usuario hace clic sobre el botón Activar Modo Edición
 2. El formulario LdM oculta los paneles laterales y muestra todas las columnas asociadas a parámetros con relevancia en ERP_ART
 3. El formulario LdM habilita las celdas para edición en función de la configuración de cada parámetro y el estado de cada componente (editable o no)
 4. El usuario modifica valores en las celdas editables.
- Postcondiciones:

4.1.5 Subir posición de material:

Incrementa en uno la posición del componente seleccionado en el árbol de transformación.

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación
- Flujo de eventos:
 1. El usuario selecciona un nodo en el árbol de transformación
 2. El usuario hace clic sobre el botón Subir Posición
 3. El formulario intercambia la posición del componente seleccionado con el componente cuya posición es inmediatamente superior dentro de la estructura de su componente padre
- Postcondiciones:

4.1.6 Bajar posición de material:

Decrementa en uno la posición del componente seleccionado en el árbol de transformación.

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación
- Flujo de eventos:
 1. El usuario selecciona un nodo en el árbol de transformación
 2. El usuario hace clic sobre el botón Bajar Posición
 3. El formulario intercambia la posición del componente seleccionado con el componente cuya posición es inmediatamente inferior dentro de la estructura de su componente padre
- Postcondiciones:

4.1.7 Introducir nueva posición manual:

- Asigna una nueva posición introducida por el usuario al componente seleccionado.
- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación
- Flujo de eventos:
 1. El usuario selecciona un componente en el árbol de transformación
 2. El usuario hace clic sobre el botón Asignar Posición
 3. Se muestra un formulario iMsgBox preguntado por la posición a asignar al componente
 4. El usuario introduce la posición y pulsa el botón Aceptar
 5. El formulario modifica el campo PosicionEnArbolTR del componente y refleja el cambio en el árbol de transformación.
- .Postcondiciones:

4.1.8 Desagrupar materiales:

Fragmenta un componente en dos componentes cuya suma de cantidades es igual a la suma del componente original

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo transformación
- Flujo de eventos:
 1. El usuario selecciona un componente en el árbol de transformación
 2. El usuario hace clic sobre el botón Fragmentar Posición
 3. Se muestra un formulario iMsgBox preguntando por la cantidad en la que separar el componente.
 4. El usuario introduce la cantidad y pulsa el botón Aceptar
 5. Se genera un nuevo componente con la cantidad introducida en la primera posición libre y se le substraee al componente seleccionado la cantidad introducida; se reflejan los cambios en el árbol de transformación.
- Postcondiciones:

4.1.9 Agrupar materiales:

Fusiona dos componentes con el mismo código en un nuevo componente, cuya cantidad es la suma de las cantidades de los componentes originales.

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo de transformación
- Flujo de eventos:
 1. El usuario selecciona dos componentes válidos
 2. El usuario hace clic sobre el botón Fusionar Posiciones
 3. Se elimina uno de los componentes, y su cantidad se agrega a la del componente restante.
- Postcondiciones:

4.1.10 Eliminar material:

Elimina un componente de la propuesta de envío LdM

- Actores: Usuario de CAD

- Precondiciones: Formulario activado en modo de transformación
- Flujo de eventos:
 1. El usuario selecciona un componente en el árbol de transformación
 2. El usuario hace clic sobre el botón Eliminar Material
 3. El componente se elimina del árbol de transformación
- Postcondiciones:

4.1.11 Importar ruta:

Obtiene las operaciones asignadas de los componentes hijos del componente seleccionado, desde el ERP.

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo de comparación TR-ERP
- Flujo de eventos:
 1. El usuario selecciona un componente en el árbol de transformación
 2. El usuario hace clic en el botón Importar Rutas
 3. Se lanza una consulta en el ERP para obtener las operaciones del componente
 4. Se asignan las operaciones a los hijos del componente, reflejando los cambios en el árbol de transformación.
- Postcondiciones:

4.1.12 Importar todas las rutas:

Obtiene las operaciones asignadas de todos los componentes en la propuesta de envío LdM.

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo de comparación TR-ERP
- Flujo de eventos:
 1. El usuario hace clic sobre el botón Importar todas las rutas
 2. El formulario carga las operaciones asignadas o por defecto para todos los componentes de la propuesta.
- Postcondiciones:

4.1.13 Asignar operación de ruta a material

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo de comparación TR-ERP
- Flujo de eventos:
 1. El usuario selecciona un componente en el árbol de transformación
 2. El usuario hace clic sobre el botón Asignar Operación
 3. Se importa la ruta del componente seleccionado
 4. Se muestra un formulario iMsgBox con las operaciones obtenidas desde el ERP
 5. El usuario selecciona la operación a asignar al componente y pulsa el botón aceptar
 6. El formulario asigna la operación elegida al componente seleccionado y actualiza el árbol de transformación.
- Postcondiciones:

4.1.14 Guardar cambios (ejecución en CAD)

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo de edición; ejecución en CAD
- Flujo de eventos:
 1. El usuario ha editado datos de uno o más componentes en el árbol de transformación
 2. El usuario hace clic sobre el botón Guardar Cambios
 3. Por cada componente editado, se registran los cambios en el ERP y en el programa CAD
 4. El formulario vuelve al modo de transformación
- Postcondiciones:

4.1.15 Guardar cambios (ejecución sin CAD)

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo de edición; ejecución sin CAD
- Flujo de eventos:
 1. El usuario ha editado datos de uno o más componentes en el árbol de transformación
 2. El usuario hace clic sobre el botón Guardar Cambios
 3. Por cada componente editado, se registran los cambios en el ERP y en la herramienta PDM
- Postcondiciones:

4.1.16 Guardar tablas en BBDD

- Actores: Usuario de CAD
- Precondiciones: Lista de materiales enviada, o formulario cerrándose
- Flujo de eventos:
 1. La propuesta se ha enviado al ERP correctamente
-o-
El usuario cierra el formulario
 2. Se muestra un iMsgBox preguntando al usuario si desea guardar la propuesta en la base de datos de la integración.
 3. El usuario hace clic sobre el botón Aceptar
 4. Se guardan las tablas de árboles CAD y transformación, y la tabla de operaciones que corresponden a la propuesta actual.
- Postcondiciones:

4.1.17 Comparación CAD-TR

- Actores: Usuario de CAD
- Precondiciones: Formulario activado en modo de transformación
- Flujo de eventos:
 1. El usuario hace clic sobre el botón Ver Diferencias
 2. Se oculta el panel derecho con el árbol ERP, y se ocultan las filas equivalentes en los árboles CAD y transformación, mostrando únicamente los materiales con diferencias
- Postcondiciones:

4.1.18 Codificar artículos

- Actores: Usuario de CAD
- Precondiciones: Formulario activado; hay componentes sin codificar en la propuesta
- Flujo de eventos:
- Postcondiciones:

4.1.19 Enviar lista de materiales a estructura

- Actores: Usuario de CAD
- Precondiciones: Formulario activado; todos los componentes están codificados
- Flujo de eventos:
- Postcondiciones:

4.1.20 Enviar lista de materiales a proyecto

- Actores: Usuario de CAD
- Precondiciones: Formulario activado; todos los componentes están codificados
- Flujo de eventos:
- Postcondiciones:

4.2 Multilinguaje

Extiende el caso de uso Configurar aplicación. El análisis se recoge en el anexo A1.1, y el resumen de la implementación en el anexo A2.1

4.3 Gestión de calidad

Extiende el caso de uso Información de artículo. El análisis se recoge en el anexo A1.2, y el resumen de la implementación en el anexo A2.2

4.4 Archivo asociado en PDM

Extiende los siguientes casos de uso:

- Configurar aplicación (configurar parámetro CAD - definir método de lectura / escritura)
- Información de artículo
- Enviar lista de materiales

El análisis se recoge en el anexo A1.3, y el resumen de la implementación en el anexo A2.3

4.5 Mantenimiento de operaciones en lista de materiales

Extiende el caso de uso Enviar lista de materiales. El análisis se recoge en el anexo A1.4, y el resumen de la implementación en el anexo A2.4

4.6 Entidades extendidas

Extiende los siguientes casos de uso:

- Configurar aplicación (configurar parámetro ERP [RPS2012])
- Información de artículo
- Enviar lista de materiales

El análisis se recoge en el anexo A1.5

4.7 Llamadas múltiples a métodos del servicio RPS2012

Extiende los siguientes casos de uso:

- Configurar aplicación (configurar parámetro ERP [RPS2012] - definir filtros)
- Información de artículo
- Enviar lista de materiales

El análisis se recoge en el anexo A1.6

4.8 Siguiete revisión en PDM

Extiende los siguientes casos de uso:

- Configurar aplicación (configurar parámetro CAD - seleccionar método de lectura / escritura)
- Información de artículo

El análisis se recoge en el anexo A1.7, y el resumen de la implementación en el anexo A2.5

4.9 Codificación manual RPS2012

Extiende el caso de uso Información de artículo. El análisis se recoge en el anexo A1.8, y el resumen de la implementación en el anexo A2.7

4.10 Filtrar parámetros en ficha de artículo

Extiende los siguientes casos de uso:

- Configurar aplicación (presentación componente)
- Información de artículo

El análisis se recoge en el anexo A1.9, y el resumen de la implementación en el anexo A2.6

4.11 Multiempresa en RPS2012

Extiende los siguientes casos de uso:

- Información de artículo
- Enviar lista de materiales

El análisis se recoge en el anexo A1.10, y el resumen de la implementación en el anexo A2.8

4.12 Comprobación de licencia para servicio RPS2012

El análisis se recoge en el anexo A1.11, y el resumen de la implementación en el anexo A2.9

4.13 Módulo iWIN

Nueva implementación de la interfaz il_PDM. El análisis y el desglose de métodos a implementar se recogen en el anexo A1.12

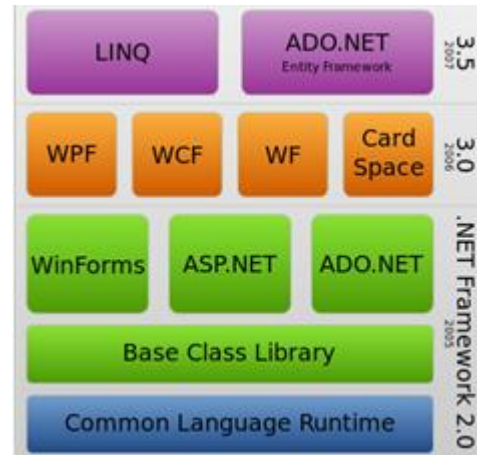
4.14 Módulo iSE

Nueva implementación de la interfaz il_CAD. El análisis y el desglose de métodos a implementar se recogen en el anexo A1.13

5. Implementación

5.1 Tecnologías

La aplicación se ha desarrollado por medio del framework .NET, empleando gran variedad de las tecnologías que allí se incluyen para implementar las capas de presentación, lógica de negocio.



5.1.1 Presentación

Para la presentación de los casos de uso - formularios y controles - se ha empleado la tecnología de Microsoft *WinForms*, parte del framework .NET



5.1.2 Lógica de negocio

La lógica de negocio de las funcionalidades se ha implementado con VB.NET y C# como lenguajes de programación.



5.1.3 Acceso a datos

Se ha empleado para el almacenamiento de datos en la aplicación OLEDB y Transact-SQL de Microsoft, con SQL-Server como servidor de base de datos



5.2 Software empleado

El software utilizado en el desarrollo del proyecto ha sido proporcionado en su totalidad por la empresa.

5.2.1 Microsoft Visual Studio



Visual Studio 2010



Visual Studio 2008

Se han implementado las funcionalidades en el lenguaje de programación VisualBasic.NET, con el entorno de desarrollo de Microsoft, Visual Studio. Esta herramienta se ha empleado principalmente en la implementación de las capas de presentación y lógica de negocio.

5.2.2 Microsoft SQL-Server



Microsoft SQL Server 2008 R2

Para el almacenamiento de datos propio de la aplicación se ha utilizado el motor de bases de datos SQL-Server de Microsoft. Esta herramienta se ha empleado principalmente para la capa de acceso a datos.

5.2.3 Internet Information Services



Microsoft IIS Server 7

Para la comunicación con el sistema ERP RPS2012, se ha empleado un servicio web de tipo WCF, que interactúa con el modelo de entidades alojado en la base de datos RPS2012. Este servicio se hospeda en un servidor IIS de la máquina a utilizar a modo de servidor.

5.2.4 VMWare Player



VMWare Player 5

Para el desarrollo de la aplicación, se han empleado varias máquinas virtuales de forma local. En estas máquinas se han probado distintos entornos de la integración, con distintos programas CAD y ERP.

5.2.5 SolidWorks



SolidWorks 2012

Uno de los programas CAD con los que trabaja la integración.

5.2.6 Enterprise PDM



SolidWorks Enterprise PDM 2012

La herramienta de gestión documental PDM incluida con SolidWorks.

5.2.7 Solid Edge



Solid Edge ST3

Uno de los programas CAD con los que trabaja la integración.

5.2.8 RPS2012



RPS 2012

Sistema ERP de Ibermática; uno de los programas ERP con los que trabaja la integración.

5.3 Gestión avanzada de listas de materiales - arquitectura de la funcionalidad

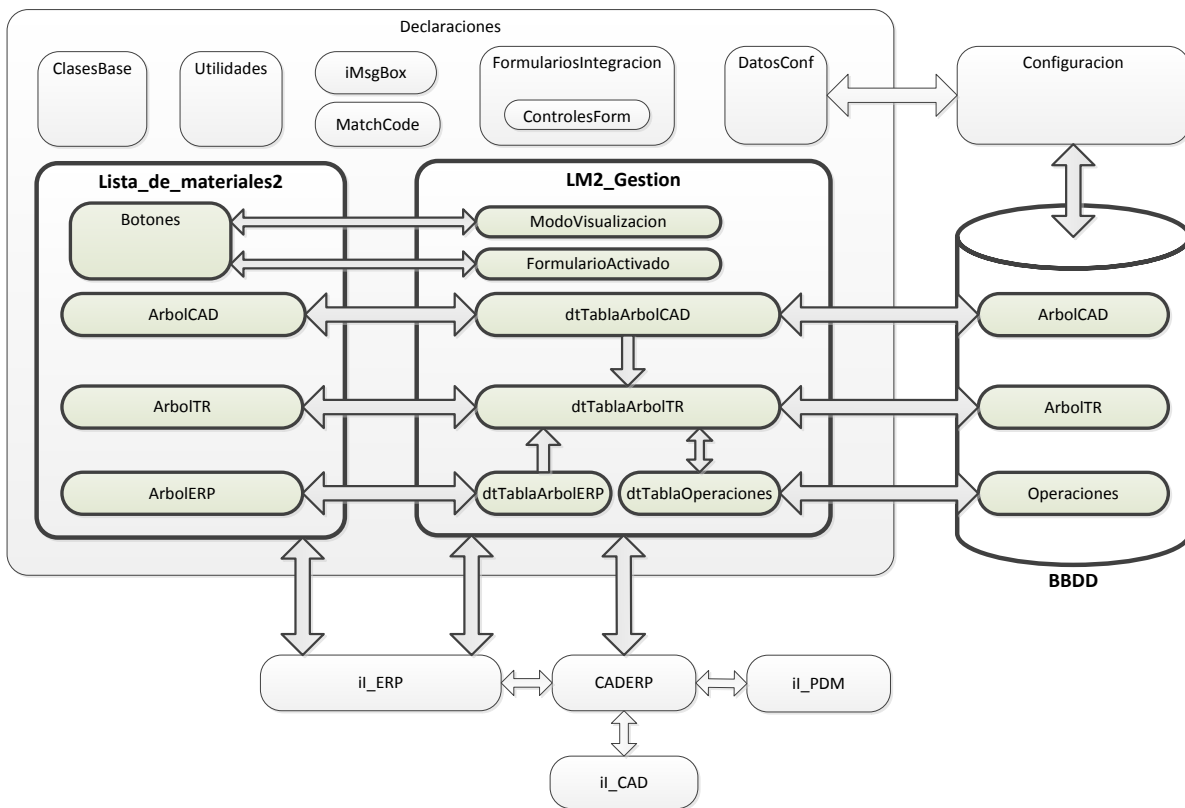


Ilustración 27: Arquitectura de la nueva funcionalidad de listas de materiales

Lista_de_materiales2:

La capa de presentación: un nuevo formulario implementado mediante Windows Forms, diseñado para reflejar la imagen y el diseño corporativos del resto de formularios de la integración

LM2_Gestion:

La capa de lógica de negocio: una clase que se comunica con el módulo CADERP para acceder a una serie de funcionalidades de la integración, y permite acceder a una base de datos para gestionar y almacenar información de envíos de listas de materiales.

Base de datos:

Fuente de datos de la integración para la nueva funcionalidad, donde se almacenan las distintas propuestas realizadas y versiones de envíos de listas de materiales enviadas.

5.4 Descripción de las nuevas clases

5.4.1 Lista_de_materiales2

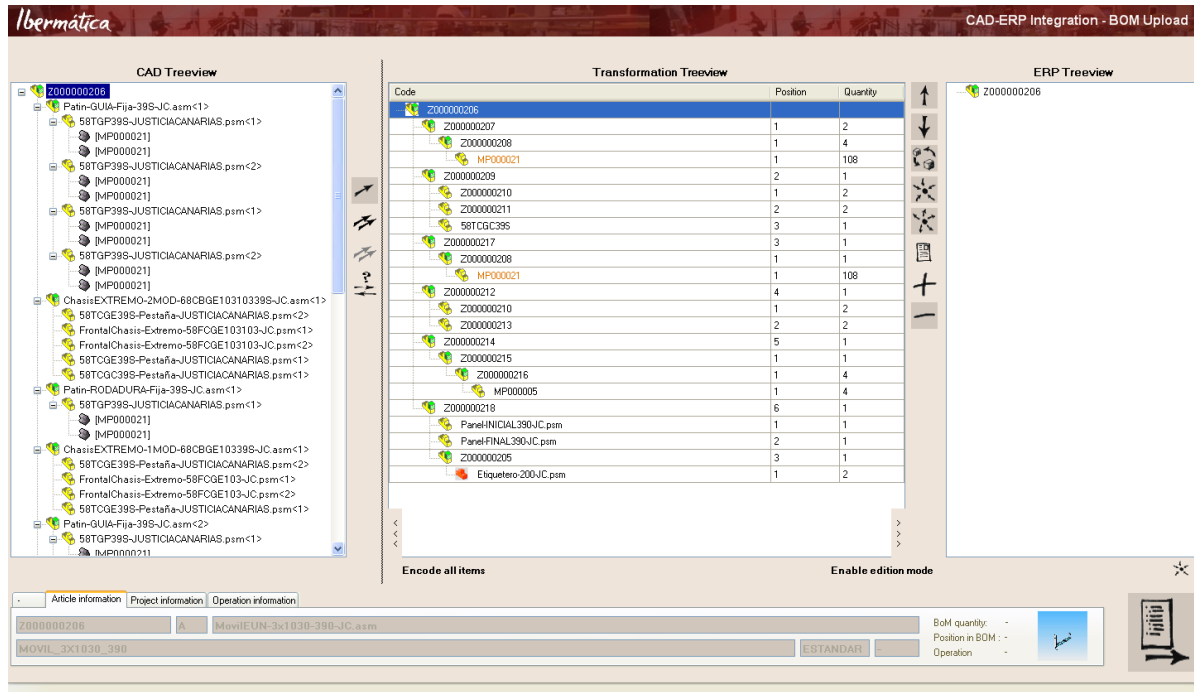
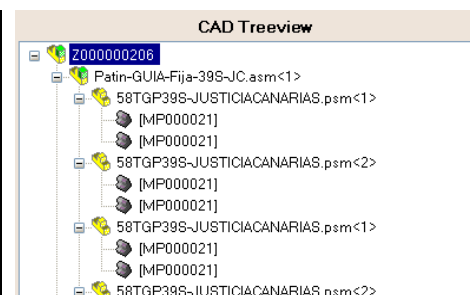


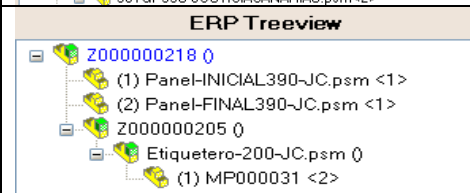
Ilustración 28: Nuevo formulario de listas de materiales

El nuevo formulario de envío de listas de materiales que se presenta al usuario de la aplicación cuenta con 3 estructuras de árbol para la visualización de la lista de materiales en los ámbitos CAD y ERP, junto con la transformación propuesta para el envío:

ArbolCAD: Muestra la lista de materiales según se extraída desde el programa CAD, identificando el tipo de componente por medio de distintos iconos. Si el formulario se lanza sin ejecutar el programa CAD, el árbol se carga con los datos de la propuesta almacenada en la BBDD de la integración, identificada por la revisión del conjunto principal.



ArbolERP: Muestra la versión activa de la lista de materiales asociada al conjunto principal extraída desde el ERP.



ArbolTR (árbol de transformación): Muestra la propuesta de envío inicial agrupada desde CAD según el criterio de agrupación configurado si se trata de un primer envío, y con la propuesta de envío anterior extraída de la BBDD de la integración si se trata de un reenvío o de una propuesta no enviada pero cuyas modificaciones se han guardado. Se trata de un control personalizado TreeGridView que conjuga la funcionalidad de un control DataGridView de visualización y edición de datos en formato tabla, con la representación arborescente de un control TreeView como el resto de árboles del formulario

Transformation Treeview		
Code	Position	Quantity
Z000000206		
Z000000207	1	2
Z000000208	1	4
MP0000021	1	108
Z000000209	2	1
Z000000210	1	2
Z000000211	2	2
581CGC395	3	1
Z000000217	3	1
Z000000208	1	1
MP0000021	1	108

Los iconos en cada nodo de los árboles indican el tipo de componente que representan, y su estado respecto al sistema ERP:



Representa un ensamblaje con revisión en el árbol CAD, y un artículo con estructura dado de alta en los árboles de transformación y ERP.



Representa un ensamblaje sin revisión en el árbol CAD, o una estructura sin codificar el árbol de transformación. No se utiliza en el árbol ERP



Representa una pieza en el árbol CAD, o un artículo sin estructura dado de alta en los árboles de transformación y ERP.



Representa un artículo sin estructura no codificado en el árbol de transformación. No se emplea en los árboles CAD y ERP



Representa un componente ficticio en el árbol CAD. No se emplea en los árboles de transformación y ERP.



Representa un componente de materia prima en los árboles CAD. No se emplea en los árboles de transformación y ERP.



Representa un nodo de operación en los árboles de transformación y ERP (modo de comparación TR-ERP). No se emplea en el árbol CAD










Además de los iconos, se emplea un código de colores en los textos de los nodos para constatar las diferencias entre los componentes de los árboles CAD y de transformación:












- **Color verde:** el elemento existe en uno de los árboles, pero no en el otro.
- **Color naranja:** el elemento existe en ambos árboles, pero las cantidades no coinciden
- **Color azul:** el elemento existe en ambos árboles, pero sus datos no coinciden
- **Color negro:** el elemento existe en ambos árboles; las cantidades y los datos coinciden

El mismo código se emplea en la comparación entre los árboles de transformación y ERP, con la adición de un nuevo código de color:

- **Color morado:** el elemento existe en ambos árboles, pero están asignados a diferentes operaciones de ruta.

Una serie de botones ubicados en el formulario permiten la ejecución de los distintos casos de uso extendidos:

<p>Botón Reagrupar: Descarta la propuesta actual en el árbol de transformación, y realiza una agrupación de la estructura de CAD con el criterio configurado</p>	
<p>Botón Reagrupar con ficticios: Descarta la propuesta actual en el árbol de transformación y realiza una agrupación de la estructura de CAD con el criterio configurado, ignorando en la agrupación los componentes ficticios</p>	
<p>Botón Pasar Elemento: Transfiere el elemento seleccionado del árbol CAD al árbol de transformación. Si no hay ningún elemento seleccionado el botón permanece desactivado</p>	
<p>Botón Ver Diferencias: Entra en el modo de comparación CAD-TR, ocultando el panel derecho que aloja el árbol ERP y botones de manipulación, y extendiendo el árbol TR para ocupar su espacio; se ocultan los componentes que tengan correspondencia en su contrapartida en los arboles CAD y TR respectivamente. Las diferencias entre componentes se representan con un código de colores en el texto del nodo</p>	
<p>Botón Ocultar panel izquierdo: Entra en el modo de comparación TR-ERP, ocultando el panel izquierdo que aloja el árbol CAD y botones de agrupación, e intercalando nodos de operaciones entre niveles de la estructura de componentes. Se desactivan en el panel derecho los botones de manipulación, y se activan a su vez los botones de gestión rutas. Las diferencias entre componentes se representan con el mismo código de colores que en la comparación CAD-TR</p>	
<p>Botón Ocultar panel derecho: Oculta el panel derecho que aloja el árbol ERP y botones de manipulación para facilitar una visualización holgada de la propuesta TR. No oculta los nodos iguales en ambos árboles</p>	
<p>Botón Codificar Todos: Realiza un alta de artículo para cada componente en la propuesta de envío que no esté codificado en el ERP (identificado por icono rojo). Al finalizar se refrescan los datos de los componentes en el árbol de transformación y se muestra el resultado de la operación. Si el botón se pulsa en el modo de edición, descarta las modificaciones de los parámetros que hayan podido realizarse y vuelve al modo de transformación</p>	
<p>Botón Activar Modo Edición: Entra en el modo de edición, ocultando los paneles izquierdo y derecho, junto con el resto de botones del formulario. Se muestran todas las columnas del árbol TR. Los botones Codificar Todos y Activar Modo Edición cambian su texto a 'Cancelar' y 'Guardar Cambios'. Si el botón se pulsa en el modo de edición, guarda los datos modificados de los artículos en el ERP, y a continuación en el programa CAD si existe una instancia del programa CAD en ejecución, o a través del programa de gestión documental PDM en caso contrario</p>	
<p>Botón Subir Posición: Intercambia la posición del componente seleccionado con la del componente inmediatamente superior dentro de la estructura de su componente padre en la propuesta de envío. Si el componente seleccionado es el primero dentro de la estructura del padre, se muestra un mensaje de error. Si no hay ningún componente seleccionado el botón permanece desactivado</p>	

<p>Botón Bajar Posición: Intercambia la posición del componente seleccionado con la del componente inmediatamente inferior dentro de la estructura de su componente padre en la propuesta de envío. Si el componente seleccionado es el último dentro de la estructura del padre, se muestra un mensaje de error. Si no hay ningún componente seleccionado el botón permanece desactivado</p>	
<p>Botón Asignar Posición: Asigna una nueva posición al componente seleccionado dentro de la estructura de su componente padre en la propuesta de envío. Si la posición introducida por el usuario no es válida, se muestra un mensaje de error.</p>	
<p>Botón Agrupar Elementos: Realiza una agrupación de los componentes seleccionados, si estos tienen el mismo código, según el criterio de agrupación establecido.</p>	
<p>Botón Fragmentar Elemento: Desagrupa el componente seleccionado en 2 componentes con el mismo código, cuya suma de cantidades es igual a la cantidad del componente original.</p>	
<p>Botón Ver Ficha Artículo: Lanza el caso de uso Información de artículo para el componente seleccionado; no se podrán modificar datos del artículo ni realizar altas desde la ficha creada, se muestra únicamente a fin de visualizar los datos</p>	
<p>Botón Insertar Elemento: Lanza el formulario de búsqueda de artículos MatchCode, y se inserta una instancia del artículo buscado bajo la estructura del componente seleccionado, con la cantidad especificada por el usuario</p>	
<p>Botón Eliminar Elemento: Elimina el elemento seleccionado de la propuesta de envío. Si el componente seleccionado tiene componentes hijos a su vez, estos también se eliminan</p>	
<p>Botón Importar Ruta: Realiza una consulta en el ERP de las operaciones asignadas al componente seleccionado y asocia estas operaciones a sus hijos, representando la asociación por medio de nodos de operación intercalados entre componentes de distinto nivel. Si alguno de los componentes hijos tiene componentes hijos a su vez, se realiza la misma operación sobre cada uno de ellos de forma recursiva</p>	
<p>Botón Importar Todas: Realiza la operación de Importar Ruta sobre el conjunto principal, trayéndose así todas las operaciones para todos los componentes de la propuesta</p>	<p>Import all routes</p>
<p>Botón Asignar Operación: Realiza la operación de Importar Ruta sobre el componente padre del seleccionado, y posteriormente permite al usuario seleccionar la operación a asignar al componente seleccionado de entre una lista de las operaciones obtenidas. El resto de componentes hermanos del seleccionado se asocian a la operación por defecto configurada</p>	
<p>Botón Refrescar ERP: Realiza una consulta al sistema ERP de la última versión activa de lista de materiales asociada al conjunto principal, y la presenta refrescando el árbol ERP</p>	
<p>Botón Enviar: Realiza el envío de la propuesta de lista de materiales activa en el árbol TR, generando en el ERP una nueva versión de estructura de fabricación, o agregando materiales a una tarea de versión de proyecto</p>	

Además de los botones, existe una serie de controles destinados a proporcionar información visual al usuario:

Barra de estado: Va presentando información al usuario sobre el progreso de determinadas operaciones que estén siendo realizadas. Los errores encontrados durante la ejecución se informan aquí, sobre fondo rojo.

Pestañas de información de nodos: Presentan información visual útil en función del nodo seleccionado:

Información de artículo: Si el nodo seleccionado corresponde a un componente, se presentan su código, revisión, descripción, tipo de componente, cantidad asignada, unidad de medida, posición en la lista de materiales, operación asignada y vista en miniatura.

Información de ruta: Si el nodo seleccionado corresponde a una operación, se presentan su código y su descripción.

Información de proyecto: Si el nodo seleccionado es el nodo principal y se corresponde con un envío a proyecto ERP, se presentan su código, versión, descripción, código de la tarea y descripción de la tarea.

En caso de ejecución sin CAD, el formulario muestra un menú en la parte superior, para especificar las credenciales de conexión al sistema de gestión documental PDM, que será el que almacene los cambios que se realicen sobre los datos de los artículos.

5.4.2 LM2_Gestion

La clase LM2_Gestion se encarga de la lógica de negocio de la funcionalidad de listas de materiales. Interactúa con el formulario Lista_de_materiales2 por medio de eventos lanzados desde el formulario, que se recogen en LM2_Gestion. Además, el formulario puede acceder a los siguientes objetos para cargar datos en los controles de presentación:

dtTablaArbolCAD (DataTable) : tabla con los datos de la estructura LdM según se extrae desde el programa CAD. Alimenta el control Treeview ArbolCAD; cada fila corresponde a la relación entre dos nodos padre-hijo. Cuenta con los siguientes campos:

- Indice: Columna de identidad (autoincremental)
- Paquete: Identificador de la propuesta de lista de materiales
- PadreArchivo: ruta local del componente padre
- PadreRev: revisión del componente padre
- PadreCod: código ERP del componente padre
- PadreDesc: descripción del componente padre
- PadreImg: cadena de bytes que componen la imagen en miniatura del componente padre
- PadreDatos: colección de datos del padre serializada en una cadena de caracteres
- HijoArchivo: ruta local del componente hijo
- HijoInstancia: número de instancia del componente hijo
- HijoRev: revisión del componente hijo
- HijoCod: código ERP del componente hijo
- HijoDesc: descripción del componente hijo
- HijoUnid: código ERP de la unidad de medida del componente hijo
- HijoImg: cadena de bytes que componen la imagen en miniatura del componente hijo
- HijoDatos: cadena de bytes que componen la imagen en miniatura del componente hijo
- Cantidad: cantidad con la que interviene el componente hijo en la lista de materiales
- PosicionEnArbolTR: posición que ocupa el componente hijo en el árbol de transformación
- TipoProyecto: 1 si se trata de un envío a proyectos, 0 en caso contrario

dtTablaArbolTR (DataTable): tabla con los datos de la propuesta de envío de lista de materiales. Alimenta el control personalizado TreeGridView ArbolTR; cada fila corresponde a la relación entre dos componentes padre-hijo. Cuenta con los siguientes campos:

- Indice: Columna de identidad (autoincremental)
- Paquete: Identificador de la propuesta de lista de materiales
- PadreCod: código ERP del componente padre
- PadreRev: revisión del componente padre
- PadreArchivo: ruta local de archivo del componente padre
- PadreDatos: colección de datos del padre serializada en una cadena de caracteres
- HijoPos: posición del componente hijo en la lista de materiales del componente padre
- HijoCod: código ERP del componente hijo
- HijoUnid: código ERP de la unidad de medida del componente hijo
- HijoArchivo: ruta local de archivo del componente hijo
- HijoDatos: colección de datos del hijo serializada en una cadena de caracteres

- Operacion: código de la operación asignada al componente hijo
- Cantidad: cantidad con la que interviene el componente hijo en la lista de materiales
- Observaciones: información adicional sobre el componente hijo
- Procedencia:
- FechaEnvio: fecha envío de la lista de materiales; si se trata de una propuesta no enviada, el campo contiene NULL
- TipoReagrupacion: 1 en el caso de envío a proyectos, 0 en caso contrario

dtTablaArbolERP (DataTable): tabla con los datos de la lista de materiales activa en el ERP. Alimenta el control TreeView ArbolERP; cada fila corresponde a la relación entre dos componentes padre-hijo. Cuenta con los siguientes campos:

- CantUnitaria: Cantidad con la que interviene el componente hijo dentro de la lista de materiales de su componente padre
- Cantidad: Cantidad con la que interviene el componente hijo dentro de la lista de materiales del conjunto principal.
- CodRecurso: Código del componente hijo dentro de la lista de materiales del conjunto principal
- CodigoElemento: Código del artículo representado por el componente hijo
- CodigoSuperior: Código del artículo representado por el componente padre
- Descripcion: Descripción del componente hijo
- Empresa: Código de la empresa en el sistema ERP
- FechaAlta: Fecha de alta del componente en la lista de materiales del conjunto principal
- NombreDocumento: Nombre o ruta de archivo de un documento asociado al componente hijo
- NombreFicPDF: Nombre o ruta de archivo de un fichero PDF asociado al componente hijo
- NumeroSecuencia: Código de material (posición en LdM) del componente hijo en la lista de materiales del componente padre.
- Observaciones: Observaciones
- Operacion: Operación asignada al componente hijo entre las operaciones del componente padre.
- Revision: Versión de la lista de materiales del componente padre.
- UnidadesLM: Código de la unidad de medida en el ERP

dtTablaOperaciones (DataTable): tabla con los datos de las operaciones asignadas a los componentes de la propuesta de lista de materiales; cada fila corresponde a una operación de la ruta de un componente.

- Indice: Columna de identidad (autoincremental)
- Paquete: Identificador de la propuesta de lista de materiales
- PadreCod: código del componente padre del que cuelga la operación
- Operacion: código de la operación
- DescripcionOp: descripción de la operación
- OperDatos: colección de datos de la operación serializada en una cadena de caracteres

ModoVisualizacion: Enumerador de los posibles modos de visualización del formulario de envío de listas de materiales:

- 0: Modo de transformación; el formulario se carga en este modo al inicializarse, mostrando los paneles laterales. En este modo se permite modificar la estructura de la propuesta de envío LdM
- 1: Modo de edición; el formulario oculta los paneles laterales con los árboles CAD y ERP, y habilita las celdas del árbol de transformación para ser editadas. En este modo se permite modificar los datos de los componentes de la lista de materiales y actualizar los datos de los artículos que representan en ERP y de los archivos 3D de CAD
- 2: Modo de comparación CAD-TR; el formulario oculta el panel ERP derecho, junto con los botones de manipulación y de gestión de rutas.
- 3: Modo de comparación TR-ERP; el formulario oculta el panel CAD izquierdo, junto con los botones de agrupación, y se desactivan los botones de manipulación, quedando activados los de gestión de rutas. Tanto en los árboles de transformación como ERP, se intercalan nodos de operación entre componentes padre-hijo de la estructura, cargados desde dtTablaOperaciones

FormularioActivado: Booleano que indica si los botones del formulario están activados o desactivados. Los botones se desactivan en los siguientes casos:

- La versión de lista de materiales actual (revisión del conjunto principal) ya ha sido enviada al ERP
- El conjunto principal o alguno de los subensamblajes extraídos desde CAD no tiene revisión asignada (no se puede generar la lista de materiales de ese conjunto)
- No hay componentes en la estructura extraída desde CAD

5.4.3 Base de datos

La base de datos se emplea para almacenar versiones y propuestas de envíos de listas de materiales. Está gestionada por un motor SQL Server de Microsoft, y se puede acceder a ella mediante el protocolo Oledb, presentando las credenciales del servidor.

Cuando se lanza la aplicación ListaDeMateriales en una ejecución independiente del programa CAD, se almacenan las siguientes tablas que alimentan los objetos de LM2_Gestion:

ArbolCAD		ArbolTR		Operaciones	
<input type="checkbox"/>	Indice	<input type="checkbox"/>	Indice	<input type="checkbox"/>	Indice
<input type="checkbox"/>	Paquete	<input type="checkbox"/>	Paquete	<input type="checkbox"/>	Paquete
<input type="checkbox"/>	PadreArchivo	<input type="checkbox"/>	PadreCod	<input type="checkbox"/>	PadreCod
<input type="checkbox"/>	PadreRev	<input type="checkbox"/>	PadreRev	<input type="checkbox"/>	Operacion
<input type="checkbox"/>	PadreCod	<input type="checkbox"/>	PadreArchivo	<input type="checkbox"/>	DescripcionOp
<input type="checkbox"/>	PadreDesc	<input type="checkbox"/>	PadreDatos	<input type="checkbox"/>	OperDatos
<input type="checkbox"/>	PadreImg	<input type="checkbox"/>	HijoPos		
<input type="checkbox"/>	PadreDatos	<input type="checkbox"/>	HijoCod		
<input type="checkbox"/>	HijoArchivo	<input type="checkbox"/>	HijoUnid		
<input type="checkbox"/>	HijoInstancia	<input type="checkbox"/>	HijoArchivo		
<input type="checkbox"/>	HijoRev	<input type="checkbox"/>	HijoDatos		
<input type="checkbox"/>	HijoCod	<input type="checkbox"/>	Operacion		
<input type="checkbox"/>	HijoDesc	<input type="checkbox"/>	Cantidad		
<input type="checkbox"/>	HijoUnid	<input type="checkbox"/>	Observaciones		
<input type="checkbox"/>	HijoImg	<input type="checkbox"/>	Procedencia		
<input type="checkbox"/>	HijoDatos	<input type="checkbox"/>	FechaEnvio		
<input type="checkbox"/>	Cantidad	<input type="checkbox"/>	TipoReagrupacion		
<input type="checkbox"/>	PosicionEnArbolTR				
<input type="checkbox"/>	TipoProyecto				

ArbolCAD --> almacena las estructuras extraídas desde CAD, y alimenta la tabla dtTablaArbolCAD cuando se lanza la aplicación ListaDeMateriales sin ninguna instancia del programa CAD en ejecución.

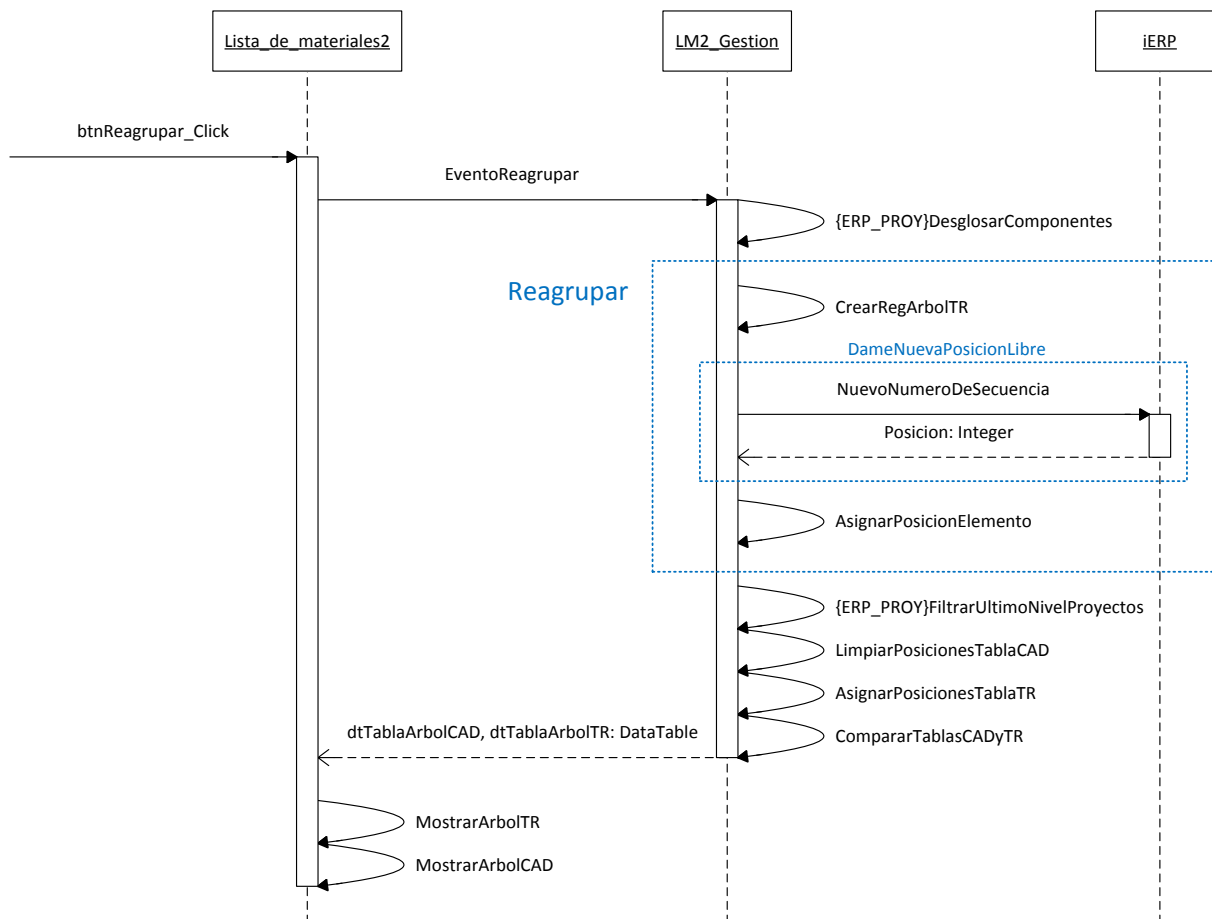
ArbolTR --> almacena las propuestas de envío de lista de materiales, y alimenta la tabla dtTablaArbolTR al cargar una nueva versión de un envío de lista de materiales ya realizado.

Operaciones --> almacena las operaciones asignadas a los componentes desde el formulario de lista de materiales, y alimenta la tabla dtTablaOperaciones al cargar una nueva versión de un envío de lista de materiales ya realizado

- Indice
- Paquete
- PadreCod
- Operacion
- DescripcionOp
- OperDatos

5.5 Diagramas de secuencia

5.5.1 Caso de uso Reagrupar desde CAD



Métodos:

btnReagrupar_Click: Controla el evento de clic sobre el botón btnReagrupar

- Precondiciones: El formulario de envío LdM está cargado, con un conjunto a enviar válido
- Entrada:
- Salida:
- Postcondiciones: Se ha lanzado el evento *Reagrupar*, a capturar por el objeto LM2_Gestion

EventoReagrupar: Controla el evento Reagrupar lanzado desde el clic del botón de reagrupar.

- Precondiciones: Hay nodos en ArbolCAD; la tabla de datos dtTablaArbolCAD tiene al menos una fila.
- Entrada:
- Salida: Si alguna precondición no se cumple devuelve mensaje de error correspondiente en la barra de estado del formulario.

- Postcondiciones: Se ha reagrupado ArbolTR según el criterio de agrupación configurado, con los datos de la estructura importada desde el programa CAD.

DesglosarComponentes: Mueve todos los nodos de nivel superior a 1 al nivel 2 del árbol CAD.

- Precondiciones: El nodo principal está identificado como proyecto (ERP_PROY), y está habilitado el envío a proyectos en la configuración.
- Entrada:
 - TablaIntermedia (DataTable) --> copia de dtTablaArbolCAD, la tabla que alimenta el árbol CAD
- Salida:
- Postcondiciones: Los subcomponentes cuelgan directamente del conjunto principal en dtTablaArbolCAD

Reagrupar: Método recursivo que recorre los nodos del árbol CAD y agrupa los componentes representados por sus nodos hijos, según el criterio de agrupación configurado, en la tabla dtTablaArbolTR.

- Precondiciones: El nodo a tratar tiene nodos hijos a su vez.
- Entrada:
 - CodPadre (String) --> Código del componente representado por el nodo a tratar.
 - CodPadreFicticio (String; Opcional) --> Código del componente a tratar, si está habilitada la agrupación con ficticios en la configuración, y el nodo representa un ficticio.
 - ArchivoPadreFicticio (String; Opcional) --> Ruta de archivo del componente a tratar, si está habilitada la agrupación con ficticios en la configuración, y el nodo representa un ficticio
 - CantidadFicticio (String; Opcional) --> Cantidad del componente a tratar, si está habilitada la agrupación con ficticios en la configuración, y el nodo representa un ficticio
 - StrDatosPadreFicticio (String; Opcional) --> Colección de datos del componente a tratar serializada en una cadena de caracteres, si está habilitada la agrupación con ficticios en la configuración, y el nodo tiene hijos un ficticio
- Salida:
- Postcondiciones: La tabla de datos dtTablaArbolTR ha sido recreada con sus componentes agrupados en función del criterio establecido (caso de uso 'Configurar tipo de agrupación')

CrearRegArbolTR: Crea una fila de datos con los datos clave proporcionados desde CAD para insertar en dtTablaArbolTR.

- Precondiciones:
- Entrada:
 - drDatosNodoCAD (DataRow): fila de datos con los datos extraídos del árbol CAD.
 - CodigoDistintoPadre (String; Opcional): código del componente estándar a asignar como padre, si está habilitada la agrupación con ficticios en la configuración, y el componente padre es ficticio.

- ArchivoDistintoPadre (String; Opcional): Ruta de archivo del componente estándar a asignar como padre, si está habilitada la agrupación con ficticios en la configuración, y el componente padre es ficticio.
- CantidadMulFicticio (Integer; Opcional): Cantidad con la que interviene el componente padre, si está habilitada la agrupación con ficticios en la configuración, y el componente padre es ficticio.
- StringDatosPadre (String; Opcional): Colección del componente estándar a asignar como padre serializada en una cadena de caracteres, si está habilitada la agrupación con ficticios en la configuración, y el componente padre es ficticio
- Salida: fila de datos (DataRow)
- Postcondiciones: la fila de datos resultante tiene los datos clave del componente CAD pasado por parámetro y el formato adecuado para ser insertado en dtTablaArbolTR.

DameNuevaPosicionLibre: Devuelve una nueva posición en el árbol TR bajo la estructura de un componente, comprobando si dicha posición está bloqueada en la lista de materiales del ERP o ya ha sido asignada en dtTablaArbolTR.

- Precondiciones:
- Entrada:
 - CodigoPadre (String) --> Código del componente bajo cuya estructura estamos consultando la siguiente posición libre.
 - TablaFilasBloqueadas (DataTable) --> Tabla de datos con los componentes cuya posición está bloqueada.
- Salida: la siguiente posición libre relativa a la estructura del componente padre indicado, o -1 en caso de no haberse encontrado ninguna posición (Integer).
- Postcondiciones:

NuevoNumeroDeSecuencia: Busca la siguiente posición sin asignar en la versión activa mas reciente de la lista de materiales del ERP.

- Precondiciones: El objeto de tipo il_ERP (miERP) está inicializado y conectado al ERP.
- Entrada:
 - TipoDeIntervalo (String) --> "P" en el caso de envío a proyectos, "L" en el caso de envío a estructuras, para buscar la posición en estructuras o en proyectos.
 - CodigoPadre (String; Opcional) --> Código del componente bajo cuya estructura se consulta por la posición libre. si se omite se busca bajo el conjunto principal.
 - PosicionAnterior (Integer; Opcional) --> Si no se omite, la función devuelve el siguiente número entero al proporcionado en este parámetro.
- Salida: La siguiente posición libre en la lista de materiales (Integer).
- Postcondiciones:

AsignarPosicionElemento: Devuelve la posición del elemento en la lista de materiales, si tenía posición asignada en el árbol ERP; sino, devuelve una nueva posición libre.

- Precondiciones:
- Entrada:

- drDatos (DataRow) --> fila de datos clave del elemento
- TablaFilasBloqueadas (DataTable; Opcional) --> Si no se omite, se leen las posiciones bloqueadas en esta tabla.
- Salida: posición a asignar al elemento en la lista de materiales (Integer)
- Postcondiciones:

FiltrarUltimoNivelProyectos: Extrae los componentes de último nivel del árbol y reconstruye la tabla dtTablaArbolTR únicamente con estos componentes.

- Precondiciones: El nodo principal está identificado como proyecto (ERP_PROY), y está habilitado el envío a proyectos en la configuración.
- Entrada:
- Salida:
- Postcondiciones: la tabla dtTablaArbolTR contiene únicamente los componentes de último nivel (aquellos sin hijos), anotados como hijos del conjunto principal.

LimpiarPosicionesTablaCAD: Método recursivo que pone a cero el campo PosicionEnArbolTR de la tabla dtTablaArbolCAD a los hijos del componente especificado, para su posterior asignación de posiciones

- Precondiciones:
- Entrada:
 - CodigoPadre (String; Opcional) --> Si se omite, se limpia la posición de los hijos del conjunto principal.
- Salida:
- Postcondiciones: Los componentes tienen el campo PosicionEnArbolTR de la tabla dtTablaArbolCAD a 0

AsignarPosicionesTablaTR: Realiza la asignación de posiciones a los componentes de la lista de materiales en dtTablaArbolTR y a sus elementos asociados en dtTablaArbolCAD, previa limpieza de posiciones.

- Precondiciones:
- Entrada:
 - CodPadre (String) --> El código del componente a cuyos hijos se va a asignar posición.
- Salida:
- Postcondiciones: Los componentes tienen el campo PosicionEnArbolTR de la tabla dtTablaArbolCAD asignado correctamente.

ComparaTablasCADyTR: Asigna el estado de los componentes en las tablas de datos dtTablaArbolCAD y dtTablaArbolTR para asignarles el color adecuado en los árboles según el esquema.

- Precondiciones:
- Entrada:
 - colDatosNodoPrincipal (Collection; Opcional) → colección de datos del conjunto principal
- Salida:

- Postcondiciones:

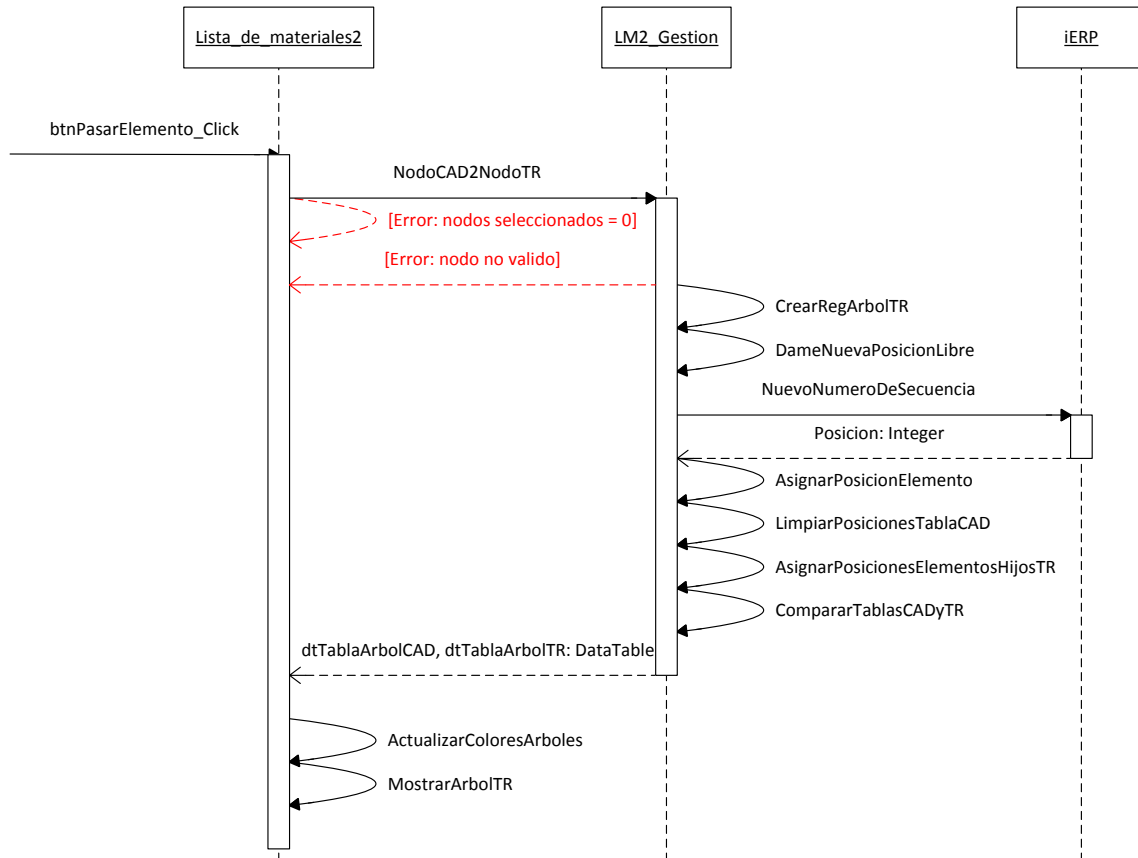
MostrarArbolCAD: Carga los nodos del árbol CAD a partir de los datos en dtTablaArbolCAD y refleja los cambios en ArbolCAD

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

MostrarArbolTR: Carga las celdas del árbol de transformación a partir de los datos en dtTablaArbolTR y refleja los cambios en ArbolTR

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

5.5.2 Caso de uso Importar componente CAD



Métodos

btnPasarElemento_Click: Controla el evento de click sobre el botón Pasar Elemento

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

NodoCAD2NodoTR: Realiza las operaciones necesarias para adecuar el nodo del árbol CAD al formato del árbol de transformación

- Precondiciones:
- Entrada:
 - CodigoPadreSeleccionado (String) → código del padre del componente seleccionado
 - CodigoSeleccionado (String) → código del componente seleccionado
 - Instancia (String) → número de instancia del componente en caso de que haya más de una instancia en la lista de materiales

- CodigoPadreTR (String) → código del padre del componente n el árbol de transformación
- ArchivoPadreTR (String) → ruta de archivo del componente padre del seleccionado
- CantidadPadreFicticio (Integer) → cantidad con la que interviene el componente padre en la lista de materiales si el padre es ficticio
- StringDatosPadreTR (String) → Colección de datos del padre serializada en una cadena de caracteres
- Salida:
- Postcondiciones:

CrearRegArbolTR: ver Caso de uso Reagrupar desde CAD

DameNuevaPosicionLibre: ver Caso de uso Reagrupar desde CAD

NuevoNumeroDeSecuencia: ver Caso de uso Reagrupar desde CAD

LimpiarPosicionesTablaCAD: ver Caso de uso Reagrupar desde CAD

AsignarPosicionesElementosHijosTR:

- Precondiciones:
- Entrada:
 - CodigoPade (String) → código del padre del componente
- Salida:
- Postcondiciones:

CompararTablasCADyTR: ver Caso de uso Reagrupar desde CAD

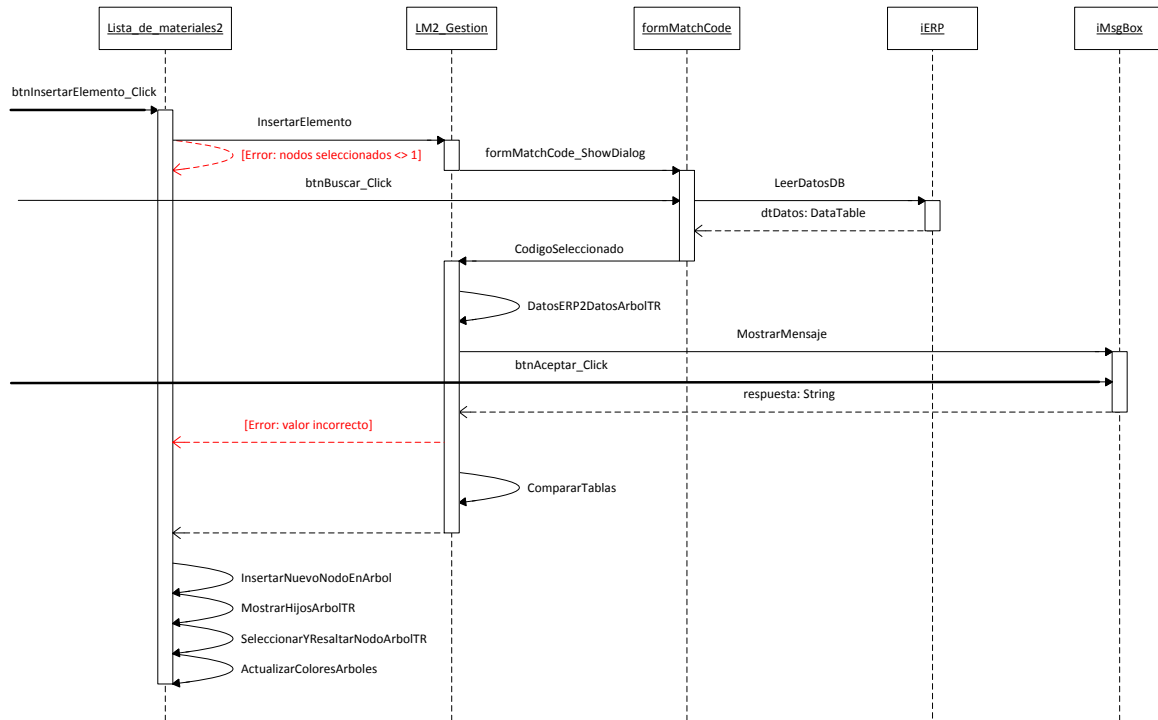
ActualizaColoresArboles: Actualiza el color de los nodos en los árboles CAD, transformación y ERP, en función del modo de visualización del formulario y el estado de los nodos

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

MostrarArbolTR: Recarga el árbol de transformación con los datos almacenados en dtTablaArbolTR

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

5.5.3 Caso de uso Importar material desde ERP



Métodos

btnInsertarElemento_Click: Controla el evento de click sobre el botón Insertar Elemento, y lanza el evento 'InsertarElemento' a capturar por LM2_Gestion

- Precondiciones:
- Entrada:
- Salida:
 - Si no se ha seleccionado ningún nodo (o se ha seleccionado más de uno) en el árbol CAD o de transformación, se muestra un mensaje de error en la barra de estado.
- Postcondiciones:

InsertarElemento (LM2_Gestion): Controla el evento 'InsertarElemento' lanzado desde Lista_de_materiales2

- Precondiciones:
- Entrada:
 - CodigoPadre (String) --> código del componente seleccionado bajo el cual se inserta el nuevo elemento
 - PosicionPadre (Integer) --> posición que ocupa el padre (componente seleccionado) en la lista de materiales
 - EsRaiz (Boolean) --> indica si el componente seleccionado es el conjunto principal
- Salida:
- Postcondiciones:

formMatchCode_ShowDialog: Muestra el formulario MatchCode para la búsqueda de artículos

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

btnBuscar_Click: Controla el evento de click sobre el botón 'Buscar' en el formulario MatchCode, y llama a LeerDatosDB del objeto iERP con los parámetros y condiciones introducidos en el formulario.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

LeerDatosDB: Ejecuta una consulta de artículos al ERP, construida a partir de los parámetros de búsqueda introducidos en el formulario.

- Precondiciones:
- Entrada:
 - colParametros (Collection) --> contiene los parámetros de artículo con los valores de búsqueda introducidos por el usuario
 - misCondiciones (Collection) --> contiene condiciones para los parámetros introducidos (mayor que, similar a, ...)
- Salida:
- Tabla de datos con los artículos que satisfacen los parámetros y condiciones de búsqueda
- Postcondiciones:

CodigoSeleccionado: Controla el evento de clic sobre el botón 'Hecho' del formulario MatchCode, indicando que se ha seleccionado un artículo entre los resultados de la búsqueda

- Precondiciones:
- Entrada:
- Salida:
 - Colección de parámetros de artículo con los valores del artículo seleccionado en el formulario MatchCode
- Postcondiciones:

DatosERP2DatosArbolTR: Construye una fila de datos para la tabla dtTablaArbolTR a partir de la colección de parámetros obtenida del formulario MatchCode

- Precondiciones:
 - Entrada:
 - DatosERP (Collection) --> Parámetros de artículo con los valores del artículo buscado en el ERP
 - CodigoPadre (String) --> código del componente en el árbol de transformación bajo el cual insertar el nuevo elemento.

- PosicionPadre (Integer) --> posición del componente en la lista de materiales bajo el cual insertar el nuevo elemento.
- EsRaiz (Boolean) --> Indica si el componente bajo el que insertar el nuevo componente es el conjunto principal
- Salida:
- Fila de datos con los datos del artículo buscado y los datos de su componente padre, con el formato adecuado para ser insertada en dtTablaArbolTR (DataRow)
- Postcondiciones:

MostrarMensaje: Muestra el formulario iMsgBox con el formato indicado para que el usuario pueda introducir un valor alfanumérico.

- Precondiciones:
- Entrada:
 - Mensaje (String) --> Texto que se presenta al usuario en el cuerpo del formulario
 - Titulo (String; Opcional) --> Texto a presentar en la cabecera del formulario (por defecto, cadena vacía)
 - Botones (Integer; Opcional) --> Entero que representa el tipo de botones a incluir en el formulario (por defecto, botón Aceptar)
 - TipoMensaje (Integer; Opcional) --> Entero que representa el tipo de mensaje del formulario (por defecto, mensaje de información)
 - PorDefecto (String; Opcional) --> Valor por defecto a utilizar si el usuario no introduce nada (por defecto, cadena vacía).
- Salida:
 - Valor alfanumérico introducido por el usuario, o el valor por defecto especificado (String)
- Postcondiciones:

btnAceptar_Click: Controla el evento de clic sobre el botón 'Aceptar' del formulario iMsgBox

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

CompararTablas: Realiza una comparación entre tablas, especificada por los argumentos pasados al método

- Precondiciones:
- Entrada:
 - CompararCADyTR (Boolean) --> indica si se realizará una comparación entre las tablas dtTablaArbolCAD y dtTablaArbolTR (por defecto, TRUE)
 - ComparaTRYERP (Boolean) --> indica si se realizará una comparación entre las tablas dtTablaArbolTR y dtTablaArbolERP (por defecto TRUE)
- Salida:
- Postcondiciones:

InsertarNuevoNodoEnArbol: Inserta un nuevo nodo en el árbol de transformación

- Precondiciones:
- Entrada:
 - `dataRowNuevo (DataRow)` --> fila de datos a partir de la que construir el nuevo nodo.
- Salida:
- Postcondiciones:

MostrarHijosArbolTR: Refresca los nodos en el árbol de transformación bajo el nodo de componente indicado, a partir de los datos en `dtTablaArbolTR`

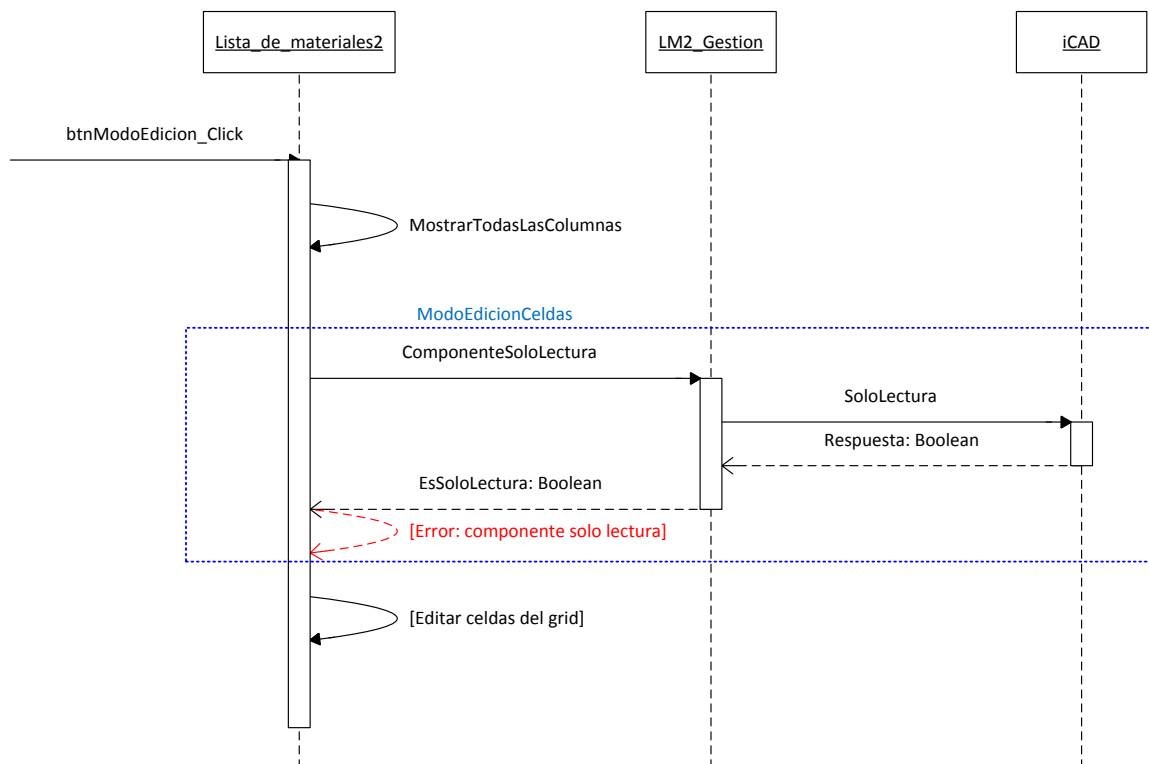
- Precondiciones:
- Entrada:
 - `NodoPadre (TreeGridNode)` --> Nodo padre bajo el que refrescar los datos
 - `HayNuevos (Boolean)` --> indica si hay nuevos nodos que insertar bajo el componente indicado
 - `TablaFilasBloqueadas (DataTable; Opcional)` --> Si no se omite, las posiciones bloqueadas se buscarán en esta tabla en lugar de consultarlas al ERP
- Salida:
- Postcondiciones:

SeleccionarYResaltarNodoArbolTR: selecciona y resalta todos los nodos equivalentes (aquellos que representan el mismo artículo) en los árboles CAD y de transformación

- Precondiciones:
- Entrada:
 - `CodigoPadre (String)` --> El código del componente padre del componente a resaltar
 - `Posicion (Integer)` --> La posición que ocupa el componente a resaltar dentro de la estructura de su componente padre.
- Salida:
- Postcondiciones:

ActualizarColoresArboles: ver Caso de uso Importar componente CAD

5.5.4 Caso de uso Editar datos de artículos



Métodos

btnModoEdicion_Click: Controla el evento de clic sobre el botón Activar Modo Edición

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

MostrarTodasLasColumnas: Muestra todas las columnas de parámetros de artículos (ERP_ART) en el árbol de transformación

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

ModoEdicionCeldas: Habilita o no para la edición las celdas de nodos hijos del componente indicado en el árbol de transformación, de forma recursiva.

- Precondiciones:
- Entrada:
 - NodoPadre (TreeGridNode) --> El nodo padre de los componentes cuyas celdas serán tratadas

- PermitirEdicion (Boolean) --> Indica si habilitar la celda o no, en función del parámetro representado por la celda y el estado del componente.
- Salida:
- Postcondiciones:

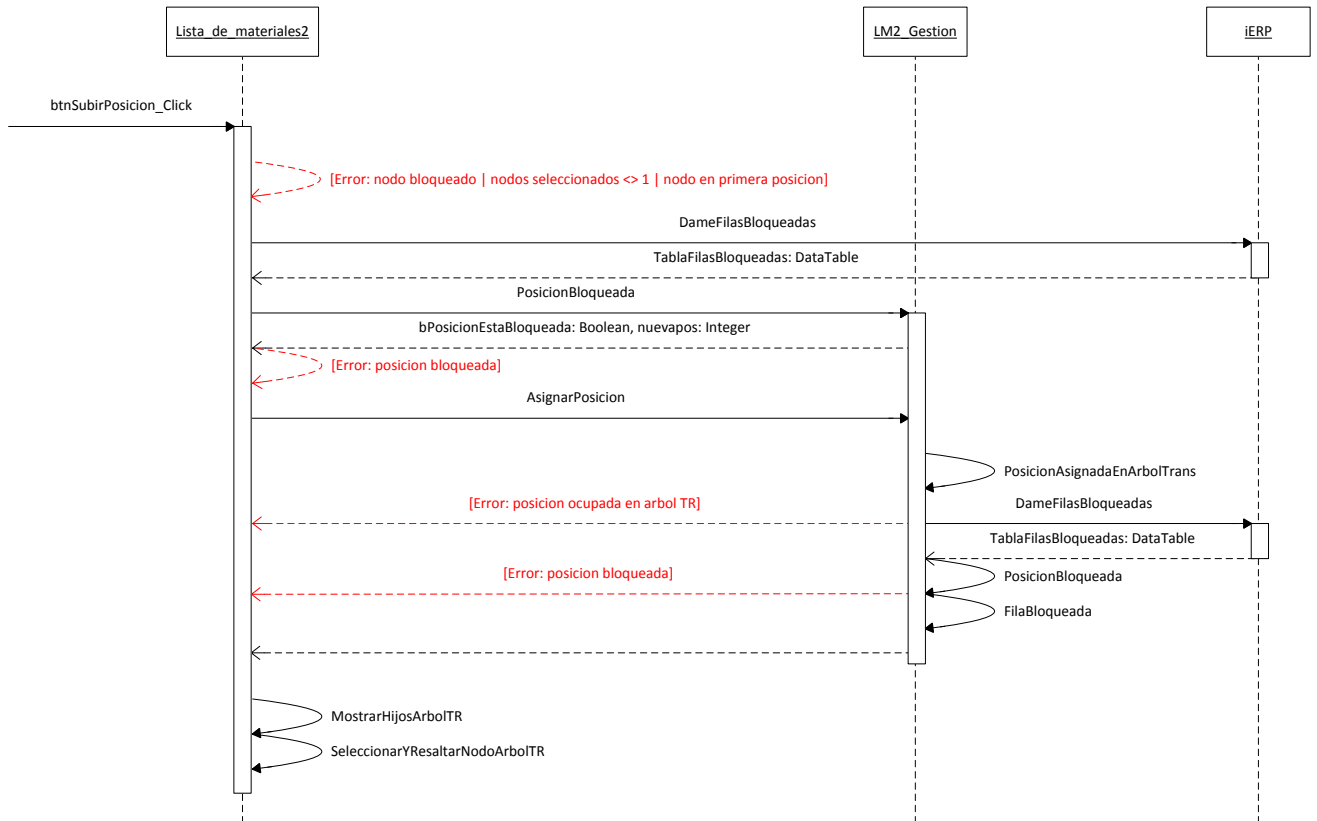
ComponenteSoloLectura: Evento que se lanza por cada componente del árbol de transformación para saber si sus datos pueden ser editados o no.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

SoloLectura: Indica si el componente especificado se encuentra en estado de sólo-lectura

- Precondiciones:
- Entrada:
 - Componente (Object) --> Puntero al modelo CAD representado por el componente cuyo estado queremos averiguar.
- Salida:
- Postcondiciones:

5.5.5 Caso de uso Subir posición de material



Métodos

btnSubirPosicion_Click: Controla el evento de clic sobre el botón Subir Posición

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

DameFilasBloqueadas: ver caso de uso Reagrupar desde CAD

PosicionBloqueada: Determina si la posición indicada bajo el componente indicado se encuentra bloqueada.

- Precondiciones:
- Entrada:
 - CodigoPadre (String) --> código del componente padre bajo cuya estructura se comprueba la posición.
 - Posicion (Integer) --> Posición bajo el componente indicado a comprobar
 - TablaFilasBloqueadas (DataTable) --> Tabla de datos donde se comprueba si la posición se encuentra bloqueada
- Salida:

- Verdadero si la posición está bloqueada, falso en caso contrario (Boolean)
- Postcondiciones:

AsignarPosicion: Asigna una nueva posición al elemento seleccionado; si existe un componente en la nueva posición, intercambia la posición del componente seleccionado con la del componente que ocupa dicha posición

- Precondiciones:
- Entrada:
 - **CodigoPadreSeleccionado (String)** --> código del padre del componente seleccionado
 - **PosicionSeleccionado (Integer)** --> posición actual del componente seleccionado dentro de la estructura de su componente padre
 - **NuevaPosicion (Integer)** --> nueva posición del componente dentro de la estructura de su componente padre.
- Salida:
- Postcondiciones:

PosicionAsignadaEnArbolTrans: Comprueba si la posición indicada bajo el componente indicado se encuentra ocupada por un elemento.

- Precondiciones:
- Entrada:
 - **CodigoPadre (String)** --> código del componente bajo cuya estructura se comprueba la posición
 - **Posicion (Integer)** --> posición a comprobar bajo la estructura del componente indicado.
- Salida:
 - Verdadero si la posición está asignada a un componente, falso en caso contrario (Boolean)
- Postcondiciones:

DameFilasBloqueadas: ver caso de uso Reagrupar desde CAD

PosicionBloqueada: ver caso de uso Reagrupar desde CAD

FilaBloqueada: Indica si la fila del árbol de transformación se encuentra bloqueada y no se puede cambiar su posición

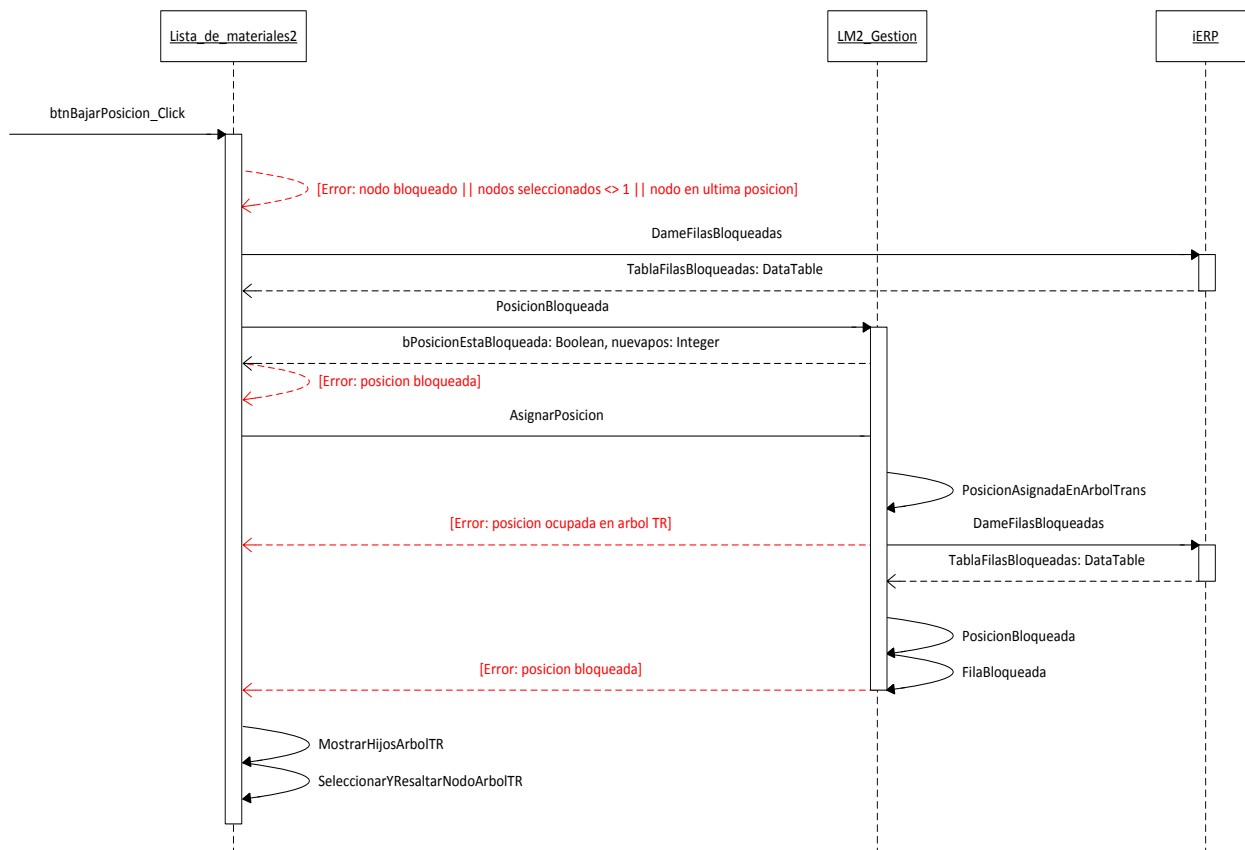
- Precondiciones:
- Entrada:
 - **CodigoPadre (String)** --> código del componente bajo cuya estructura se comprueba la fila
 - **CodigoHijo (String)** --> código del componente cuya fila se comprueba
 - **Posicion (Integer)** --> Posición bajo el componente padre cuya fila se comprueba
 - **TablaFilasBloqueadas (DataTable)** --> tabla de datos que contiene las filas bloqueadas
- Salida:

- Verdadero si la fila identificada por los argumentos se encuentra bloqueada, y falso en caso contrario (Boolean)
- Postcondiciones:

MostrarHijosArbolTR: ver caso de uso Importar material desde ERP

SeleccionarYResaltarNodosArbolTR: ver caso de uso Importar material desde ERP

5.5.6 Caso de uso Bajar posición de material



Métodos

btnBajarPosicion_Click: Controla el evento de clic sobre el botón Bajar Posición

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

DameFilasBloqueadas: ver caso de uso Reagrupar desde CAD

AsignarPosicion: ver caso de uso Subir posición de material

PosicionBloqueada: ver caso de uso Reagrupar desde CAD

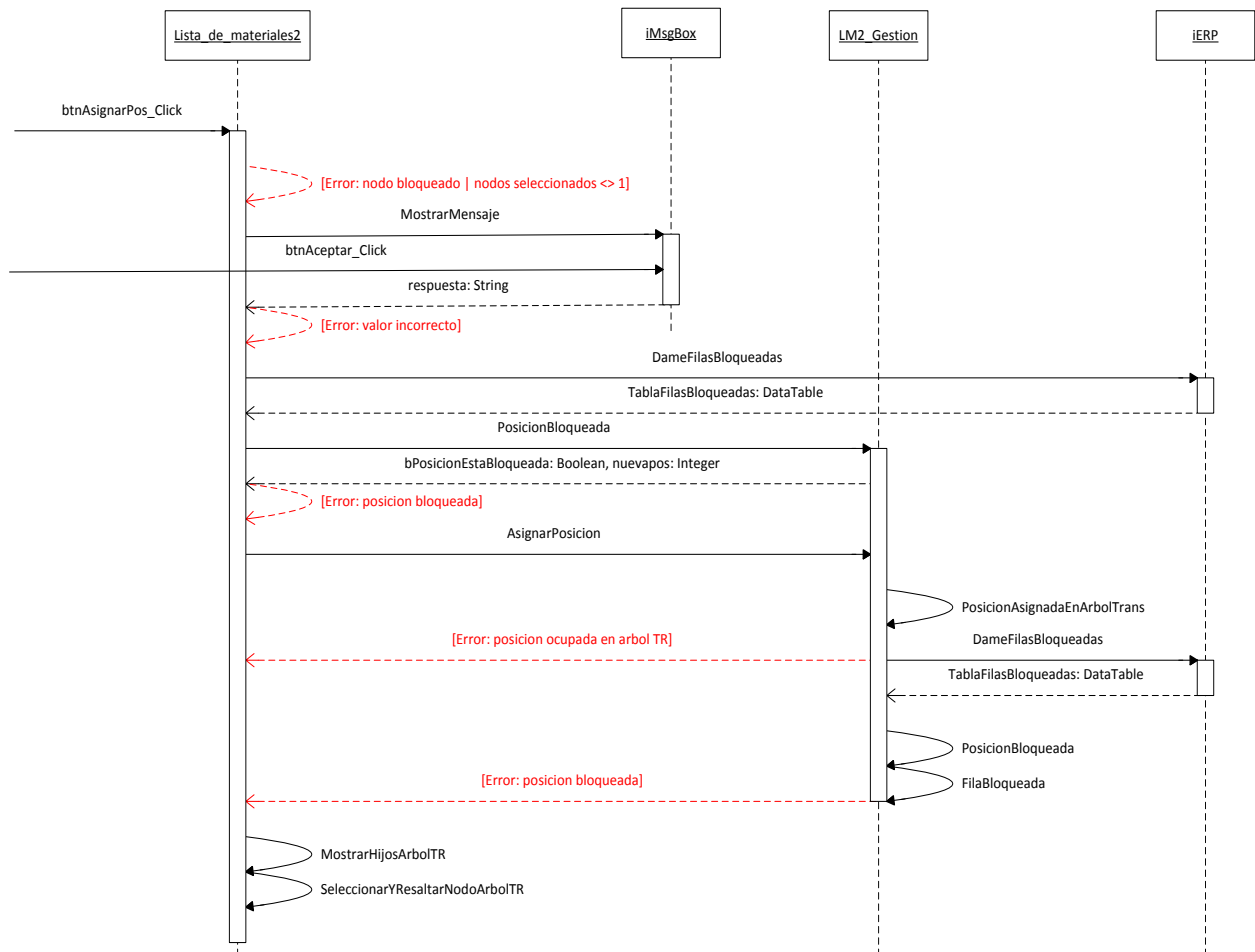
PosicionAsignadaEnArbolTrans: ver caso de uso Subir posición de material

FilaBloqueada: ver caso de uso Subir posición de material

MostrarHijosArbolTR: ver caso de uso Importar material desde ERP

SeleccionarYResaltarNodosArbolTR: ver caso de uso Importar material desde ERP

5.5.7 Caso de uso Introducir nueva posición manual



Métodos

btnAsignarPos_Click: Controla el evento de clic sobre el botón Asignar Posición

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

MostrarMensaje: ver caso de uso Importar material desde ERP

DameFilasBloqueadas: ver caso de uso Reagrupar desde CAD

PosicionBloqueada: ver caso de uso Reagrupar desde CAD

AsignarPosicion: ver caso de uso Subir posición de material

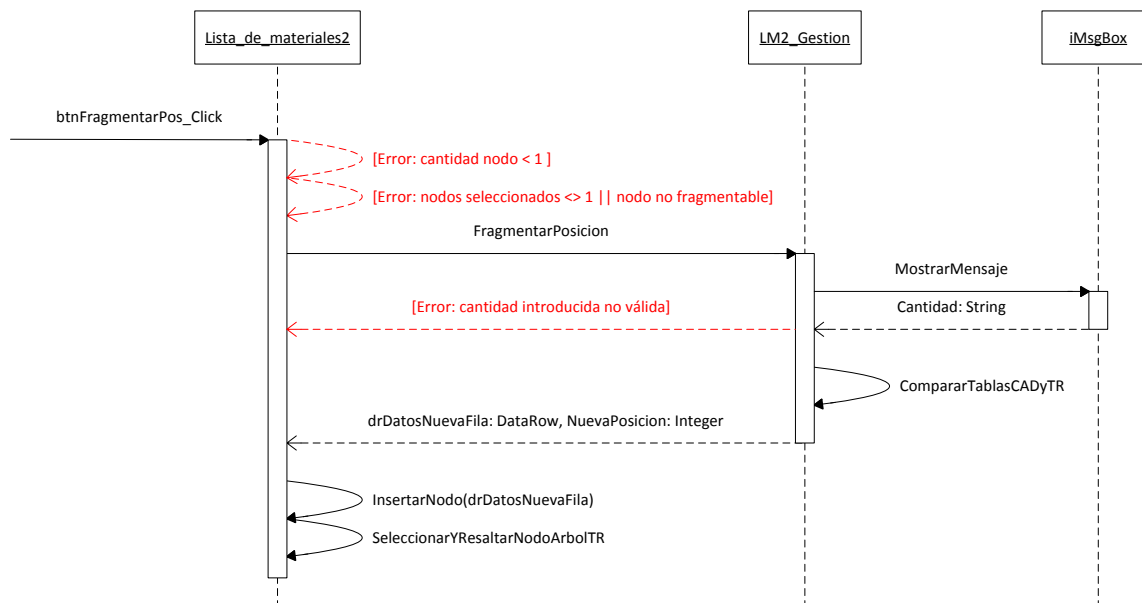
PosicionAsignadaEnArbolTrans: ver caso de uso Subir posición de material

FilaBloqueada: ver caso de uso Subir Posición material

MostrarHijosArbolTR: ver caso de uso Importar material desde ERP

SeleccionarYResaltarNodosArbolTR: ver caso de uso Importar material desde ERP

5.5.8 Caso de uso Desagrupar materiales



Métodos

btnFragmentarPos_Click: Controla el evento de clic sobre el botón Fragmentar Posición

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

FragmentarPosicion: Divide el nodo seleccionado en dos nodos cuyas cantidades suman la cantidad del nodo original seleccionado.

- Precondiciones:
- Entrada:
 - CodigoPadreSeleccionado (String) --> código del componente padre del componente seleccionado
 - PosicionSeleccionado (Integer) --> posición del componente seleccionado dentro de la estructura del componente padre
 - NuevaPosicion (Integer; Retorno)
 - drDatosNuevaFila (DataRow; Retorno)
- Salida:
 - NuevaPosicion (Integer; Retorno) --> posición en la que se ha insertado el nuevo componente resultado de la fragmentación.
 - drDatosNuevaFila (DataRow; Retorno) --> fila de datos con los datos del componente resultado de la fragmentación
- Postcondiciones:

MostrarMensaje: ver caso de uso Importar material desde ERP

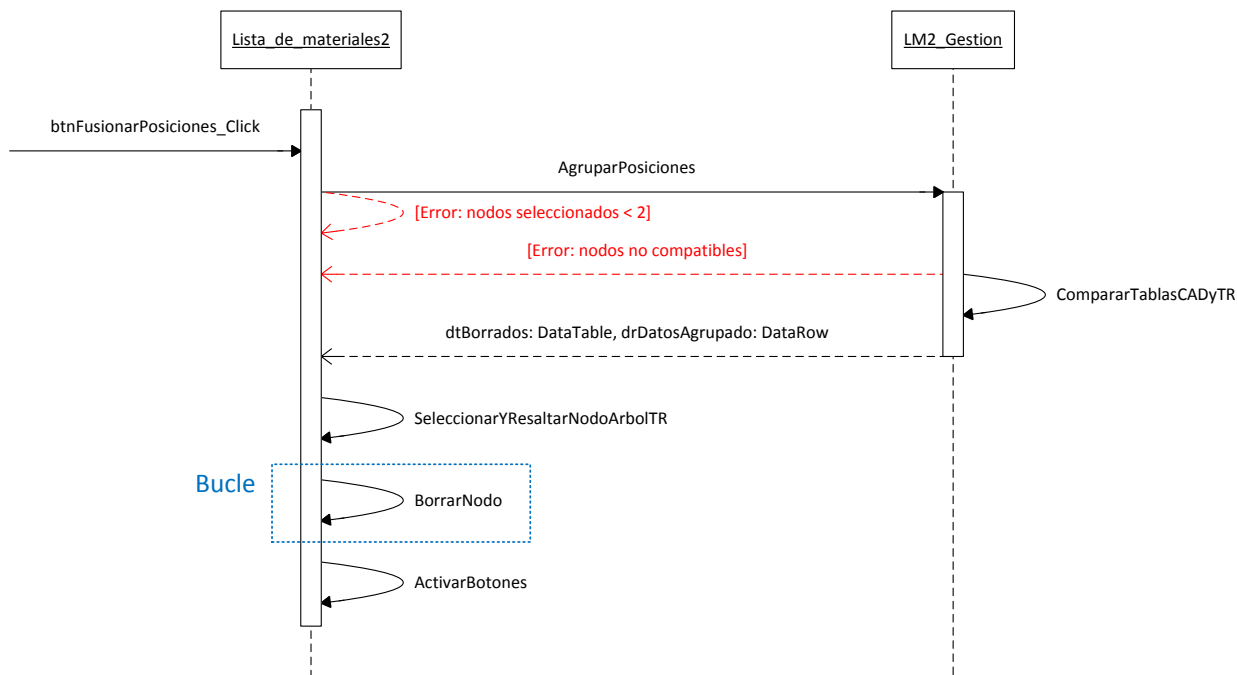
ComparaTablasCADyTR: ver caso de uso Reagrupar desde CAD

InsertarNodo: Genera un nodo en el árbol de transformación a partir de los datos en la fila de datos pasada como parámetro de la función

- Precondiciones:
- Entrada:
 - dataRowActual (DataRow) --> fila de datos a partir de la cual se construye el nodo en ArbolTR
 - NodoPadre (TreeGridNode) --> Nodo del árbol de transformación bajo el cual insertar el nuevo nodo
 - TablaFilasBloqueadas (DataTable; Opcional) --> Tabla de datos con las filas bloqueadas en ArbolTR. Si no se omite, se buscan aquí las filas bloqueadas en lugar de consultar al ERP
- Salida:
 - Nuevo nodo a insertar en el árbol de transformación (TreeGridNode)
- Postcondiciones:

SeleccionarYResaltarNodosArbolTR: ver caso de uso Importar material desde ERP

5.5.9 Caso de uso Agrupar materiales



Métodos

btnFusionarPosiciones_Click: Controla el evento de clic sobre el botón Fusionar Posiciones

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

AgruparPosiciones: Fusiona dos filas de datos en una sola siempre que sus datos sean compatibles

- Precondiciones:
- Entrada:
 - DataTableSeleccionado (DataTable) --> Tabla de datos a la que pertenecen ambas filas de datos
 - DataTableBorrados (DataTable; Retorno) --> Tabla de datos con los elementos que han sido borrados de la tabla seleccionada
 - drDatosAgrupados (DataRow; Retorno) --> Fila de datos con los datos resultantes de la fusión.
- Salida:
- Postcondiciones:

CompararTablasCADyTR: ver caso de uso Reagrupar desde CAD

SeleccionarYResaltarNodosArbolTR: ver caso de uso Importar material desde ERP

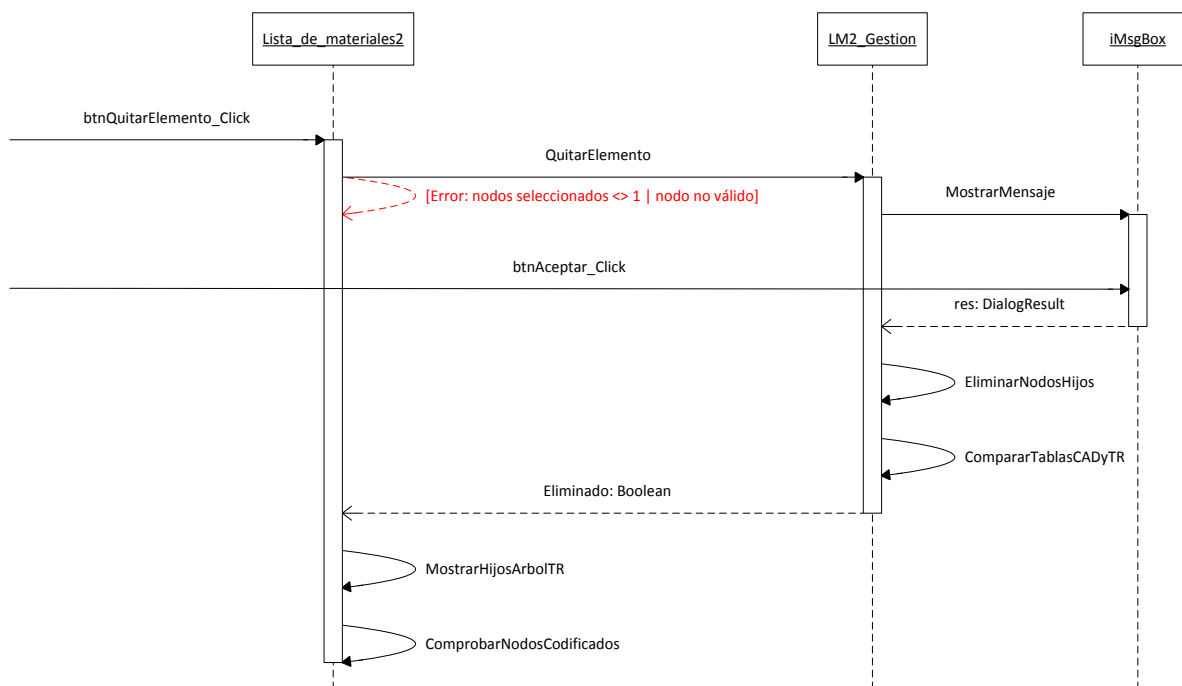
BorrarNodo: Elimina un nodo en el árbol de transformación según la posición indicada y el nodo padre indicado

- Precondiciones:
- Entrada:
 - NodoPadre (TreeGridNode) --> Nodo del árbol de transformación bajo el cual se desea borrar el nodo.
 - Posicion (Integer) --> Posición bajo el nodo padre indicado en la que se halla el nodo a borrar.
- Salida:
- Postcondiciones:

ActivarBotones: Habilita los botones del formulario en función del modo de visualización y el nodo actualmente seleccionado.

- Precondiciones:
- Entrada:
 - ArbolSeleccionado (String) --> nombre del árbol cuyos botones se desean activar o desactivar
 - Nodo (TreeNode; Opcional) --> nodo seleccionado en el árbol
- Salida:
- Postcondiciones:

5.5.10 Caso de uso Eliminar material



Métodos

btnQuitarElemento_Click: Controla el evento de clic sobre el botón Eliminar Elemento

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

QuitarElemento: Elimina el componente seleccionado de la propuesta de lista de materiales en el árbol de transformación

- Precondiciones:
- Entrada:
 - CodigoPadreSeleccionado (String) --> código del componente padre del componente seleccionado
 - PosicionSeleccionado (Integer) --> posición del componente dentro de la estructura del componente padre.
 - EsRaiz (Boolean) --> indica si el componente seleccionado es el conjunto principal o no
 - Eliminado (Boolean; Retorno)
- Salida:
 - Eliminado (Boolean; Retorno) --> Verdadero si el componente se ha eliminado de la propuesta, y falso en caso contrario.
- Postcondiciones:

MostrarMensaje: ver caso de uso Importar material desde ERP

btnAceptar_Click: ver caso de uso Importar material desde ERP

EliminarNodosHijos: Elimina los nodos hijos del componente indicado en el árbol de transformación

- Precondiciones:
- Entrada:
 - CódigoPadre (String) --> código del componente cuyos hijos se van a eliminar del árbol de transformación
- Salida:
- Postcondiciones:

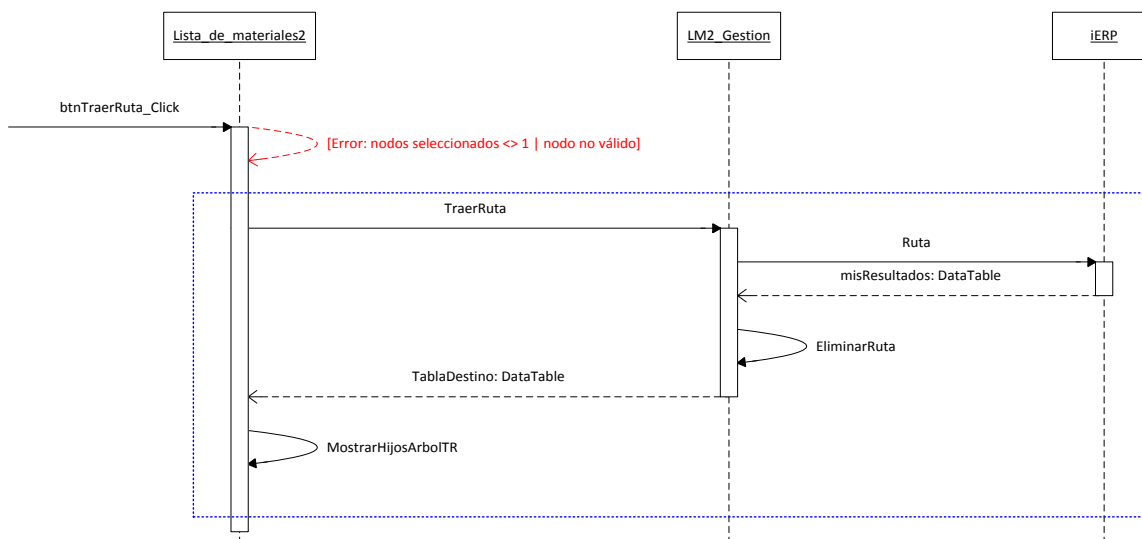
CompararTablasCADyTR: ver caso de uso Reagrupar desde CAD

MostrarHijosArbolTR: ver caso de uso Importar material desde ERP

ComprobarNodosCodificados: Método recursivo que recorre el los componentes hijos del nodo pasado como argumento comprobando si los componentes están dados de alta en el ERP y asignándoles el icono correspondiente

- Precondiciones:
- Entrada:
 - NodoPadre (TreeGridNode) --> Nodo del árbol de transformación cuyos hijos se van a comprobar
- Salida:
 - Verdadero si hay componentes sin codificar bajo el nodo indicado, y falso en caso contrario (Boolean)
- Postcondiciones:

5.5.11 Casos de uso: Importar ruta / Importar todas las rutas



Métodos

btnTraerRuta_Click: Controla el evento de clic sobre el botón Traer Ruta

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

TraerRuta: Realiza una consulta en el ERP de la ruta asignada al componente seleccionado y asocia las componentes hijos con su operación correspondiente en función del tipo de artículo de cada uno. El método es recursivo, y se vuelve a llamar para cada componente hijo del componente cuyo código se pasa como argumento. El caso de uso 'Importar todas las rutas' llama a esta función pasando como argumento el código del conjunto principal.

- Precondiciones:
- Entrada:
 - CodigoArticulo (String) --> código del artículo cuya ruta se consulta
 - TablaDestino (DataTable; Retorno)
 - RutaPlantillaAlternativa (Boolean) --> indica si las operaciones de la ruta deben importarse desde una plantilla de artículo distinta a la plantilla de artículo por defecto
- Salida:
 - TablaDestino (DataTable; Retorno) --> Tabla de datos con las operaciones de la ruta asociada al artículo cuyo código se pasa como argumento.
- Postcondiciones:

Ruta: Busca la ruta del artículo identificado por los datos que se pasan como argumento

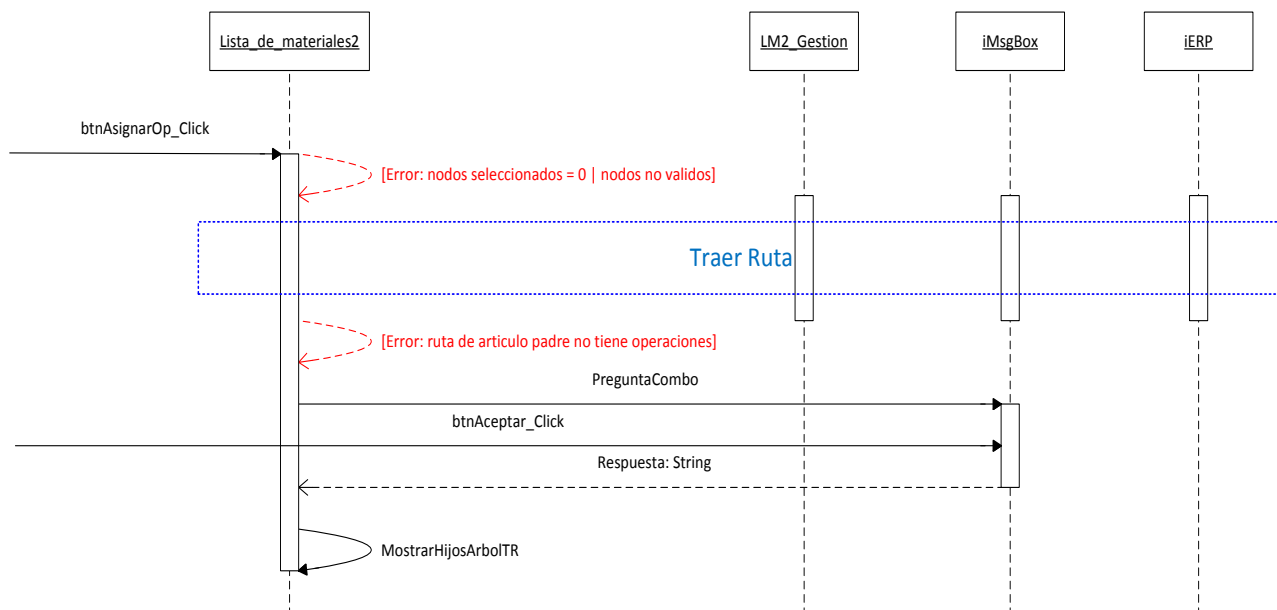
- Precondiciones:
- Entrada:
 - Datos (Collection) --> colección de datos del artículo cuya ruta se consulta
 - RutaPlantillaAlternativa (Boolean) --> indica si la ruta debe consultarse en la plantilla por defecto o en la plantilla de artículo incluida en la colección de datos
- Salida:
 - Tabla de datos con las operaciones que forman parte de la ruta consultada (DataTable)
- Postcondiciones:

EliminarRuta: Elimina las operaciones asignadas a los artículos representados por las filas de la tabla de datos pasada como argumento

- Precondiciones:
- Entrada:
 - CodigoArticulo (String) --> código del artículo cuyas operaciones se van a eliminar de la tabla de datos
 - TablaDestino (DataTable) --> tabla de datos con artículos y sus operaciones asignadas
- Salida:
 - TablaDestino (DataTable) --> tabla de datos con artículos sin operaciones asignadas
- Postcondiciones:

MostrarHijosArbolTR: ver caso de uso Importar material desde ERP

5.5.12 Caso de uso Asignar operación de ruta a material



Métodos

btnAsignarOp_Click: Controla el evento de clic sobre el botón asignar operación, y asigna la operación especificada al componente seleccionado.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

TraerRuta: ver caso de uso Importar ruta / Importar todas las rutas

PreguntaCombo: Muestra el formulario iMsgBox con el formato adecuado para que el usuario seleccione un valor de entre una lista de valores mediante un ComboBox

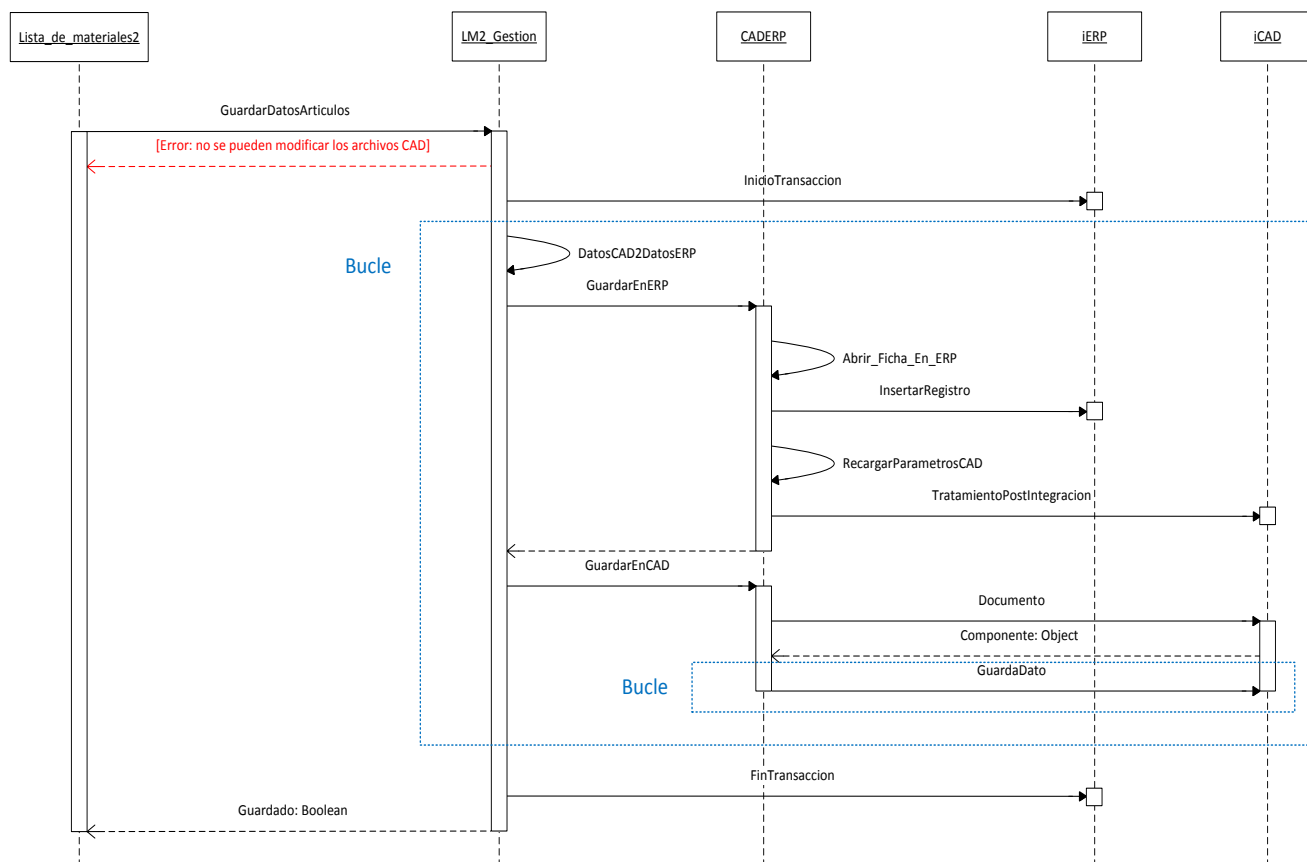
- Precondiciones:
- Entrada:
 - Mensaje (String) --> Texto a mostrar en el cuerpo del formulario iMsgBox
 - Titulo (String; Opcional) --> Texto a mostrar en la cabecera del formulario iMsgBox
 - DatosCombo (Object) --> Objeto que contiene los valores a mostrar en el ComboBox
 - Respuesta (String)
- Salida:
 - Respuesta (String; Retorno) --> valor seleccionado en el ComboBox que se devuelve al pulsar el botón 'Aceptar'

- Postcondiciones:

btnAceptar_Click: ver caso de uso Importar material desde ERP

MostrarHijosArbolTR: ver caso de uso Importar material desde ERP

5.5.13 Caso de uso Guardar cambios (ejecución en CAD)



Métodos

GuardarDatosArticulos: Método que controla el evento lanzado al pulsar el botón Activar Modo Edición (Guardar Cambios) durante el modo de edición.

- Precondiciones:
- Entrada:
 - Guardado (Boolean)
- Salida:
 - Guardado (Boolean; Retorno) --> Verdadero si los datos se han guardado correctamente, falso en caso contrario
- Postcondiciones:

InicioTransaccion: Función que inicia una 'transacción' para realizar varias acciones de forma transaccional en el ERP (o se realizan todas con éxito, o no se realiza ninguna). Guarda la información relevante del estado de la propuesta previa a las acciones a realizar, y la restaura en caso de que alguna acción falle o no se pueda realizar.

- Precondiciones:
- Entrada:

- IDTransaccion (Integer) --> número identificador de la transacción
- Comentario (String) --> breve descripción de las acciones a realizar en la transacción
- Salida:
- Postcondiciones:

DatosCAD2DatosERP: Vuelca los datos de un artículo obtenidos desde CAD sobre una colección de datos ERP según la configuración y relevancia de cada parámetro

- Precondiciones:
- Entrada:
 - DatosCAD (Collection) --> Colección de datos de artículo obtenidos desde el programa CAD
 - DatosERP (Collection)
 - CheckSoloLectura (Boolean) --> Indica si se debe tener en cuenta la relevancia del parámetro en el ámbito ERP_ART a la hora de volcar la información correspondiente de CAD (si relevancia = W, volcar información)
- Salida:
- Postcondiciones:

GuardarEnERP: Guarda los datos contenidos en la colección pasada como argumento sobre el artículo correspondiente en el ERP

- Precondiciones:
- Entrada:
 - DatosERP (Collection) --> Datos del artículo a guardar en el ERP
- Salida:
 - DatosERP (Collection; Retorno) --> Datos del artículo actualizados desde el ERP según la relevancia de cada parámetro
- Postcondiciones:

Abrir_Ficha_en_ERP: Obtiene los parámetros de un artículo en el ERP y guarda los valores en el programa CAD de aquellos cuya relevancia lo permita (relevancia = W)

- Precondiciones:
- Entrada:
 - Datos (Collection) --> colección de datos del artículo a guardar
 - ActualizarCAD (Boolean) --> indica si se deben actualizar los datos en el programa CAD tras guardar los datos en el ERP
- Salida:
 - Verdadero si se han podido guardar los datos del parámetro en el ERP, y falso en caso contrario
- Postcondiciones:

InsertarRegistro: Genera un nuevo artículo en el ERP con los datos pasados como argumento, o actualiza los datos de un artículo existente

- Precondiciones:

- Entrada:
 - DatosArticulo (Collection) --> Datos del artículo a generar o actualizar
 - IDTransaccion (Integer) --> identificador de la transacción a la que pertenece la operación
- Salida:
 - Verdadero si se ha creado un artículo nuevo o guardado los datos de uno existente, y falso en caso contrario.
- Postcondiciones:

RecargarParametrosCAD: Limpia los valores de una colección de parámetros de artículo obtenidos desde CAD

- Precondiciones:
- Entrada:
 - colParamCAD (Collection) --> colección de parámetros CAD cuyo valor queremos borrar
- Salida:
 - colParamCAD (Collection; Retorno) --> colección de parámetros CAD con valor borrado
- Postcondiciones:

TratamientoPostIntegracion: Ejecuta la macro del programa CAD configurada según el caso de uso *Configurar macro antes de*

- Precondiciones:
- Entrada:
- Salida:
 - Verdadero si la macro se ha ejecutado con éxito (o si no hay ninguna macro configurada), y falso en caso contrario (Boolean)
- Postcondiciones:

GuardarEnCAD: Guarda los datos del artículo sobre su archivo CAD correspondiente

- Precondiciones:
- Entrada:
 - Datos (Collection) --> colección de datos de artículo a guardar sobre el archivo CAD correspondiente
 - Archivo (String) --> ruta del archivo sobre el que guardar los datos
- Salida:
- Postcondiciones:

Documento: Obtiene un puntero a un objeto COM que representa un archivo CAD determinado

- Precondiciones:
- Entrada:
 - NombreDoc(String; Opcional) --> nombre del archivo CAD (sin la ruta de archivo)
 - CodigoDocumento (String; Opcional) --> código del artículo ERP representado por el archivo CAD

- Salida:
 - Puntero a un objeto COM que representa un archivo CAD abierto por el programa CAD
- Postcondiciones:

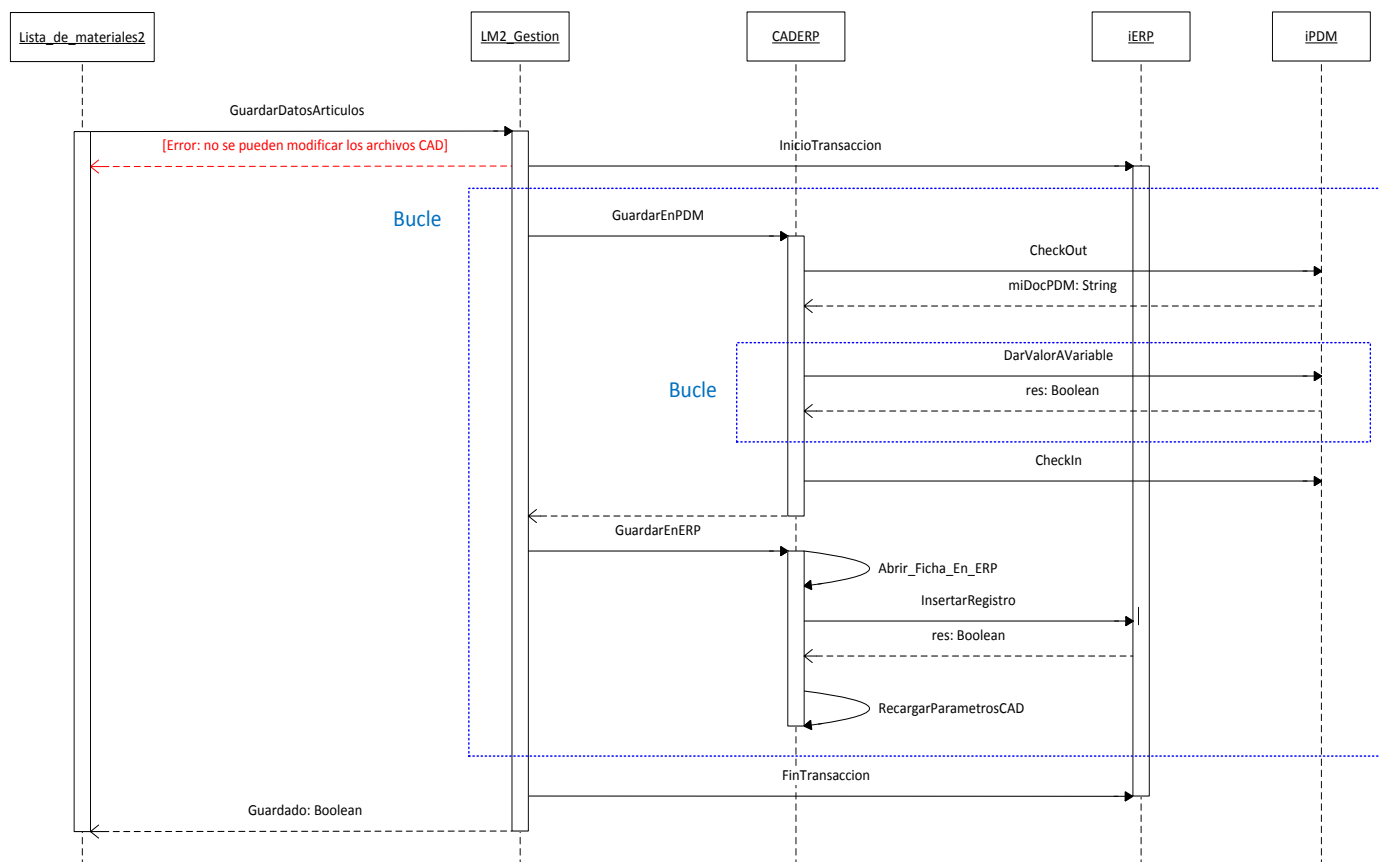
GuardaDato: Guarda datos de artículo sobre el archivo CAD abierto correspondiente, o sobre el mismo archivo CAD gestionado por la herramienta PDM en el caso de ejecución sin CAD

- Precondiciones:
- Entrada:
 - Dato (DefParametroAplicacion) --> parámetro de la integración que representa un dato de artículo asociado al archivo CAD
 - Componente (Object; Opcional) --> Puntero al objeto COM que representa el archivo CAD sobre el que guardar los datos; si se omite, los datos se guardan sobre el archivo actualmente abierto en el programa CAD.
- Salida:
 - Verdadero si se ha guardado el dato correctamente, y falso en caso contrario (Boolean)
- Postcondiciones:

FinTransaccion: Función que finaliza una transacción determinada, validando o anulando todas las operaciones realizadas en función del resultado de dichas operaciones.

- Precondiciones:
- Entrada:
 - IDTransaccion (Integer) --> número identificador de la transacción a finalizar
 - Transaccion_OK (Boolean) --> indica si se deben validar (TRUE) o anular (FALSE) las operaciones de la transacción.
- Salida:
 - Verdadero si se ha finalizado la transacción correctamente, y falso en caso contrario (Boolean)
- Postcondiciones:

5.5.14 Caso de uso Guardar cambios (ejecución sin CAD)



Métodos

GuardarDatosArticulos: ver caso de uso Guardar cambios (ejecución en CAD)

InicioTransaccion: ver caso de uso Guardar cambios (ejecución en CAD)

GuardarEnPDM: Controla el evento lanzado desde LM2_Gestion y procede a guardar los datos del artículo por medio de la herramienta PDM

- Precondiciones:
- Entrada:
 - Datos (Collection) --> colección de datos del artículo
 - Archivo (String) --> ruta o nombre del archivo
- Salida:
- Postcondiciones:

CheckOut: "Trae" el archivo (deshace el bloqueo que tuviera en el almacén de datos PDM), dejándolo en la posesión del usuario del sistema PDM hasta que éste lo "registre" (restablecer el bloqueo en el almacén)

- Precondiciones:

- Entrada:
 - Documento (String) --> ruta o nombre del archivo
 - EstabaTraido (Boolean)
- Salida:
 - EstabaTraido (Boolean; Retorno) --> Verdadero si el documento ya estaba traído por el usuario, y falso en caso contrario.
- Postcondiciones:

DarValorAVariable: Modifica el valor de una propiedad del archivo CAD, o la crea si no existe.

- Precondiciones:
- Entrada:
 - Documento (String) --> ruta o nombre del archivo CAD
 - VariableCAD (String) --> nombre de la propiedad a modificar o crear
 - Valor (String) --> valor de la propiedad a modificar o crear
- Salida:
 - Verdadero si se ha modificado o creado la variable, y falso en caso contrario
- Postcondiciones:

CheckIn: "Registra" el archivo, validando los cambios que el usuario en posesión del archivo haya realizado.

- Precondiciones: El archivo estaba "traído" por el usuario.
- Entrada:
 - Archivo (String) --> nombre o ruta del archivo
- Salida:
- Postcondiciones:

GuardarEnERP: ver caso de uso Guardar cambios (ejecución en CAD)

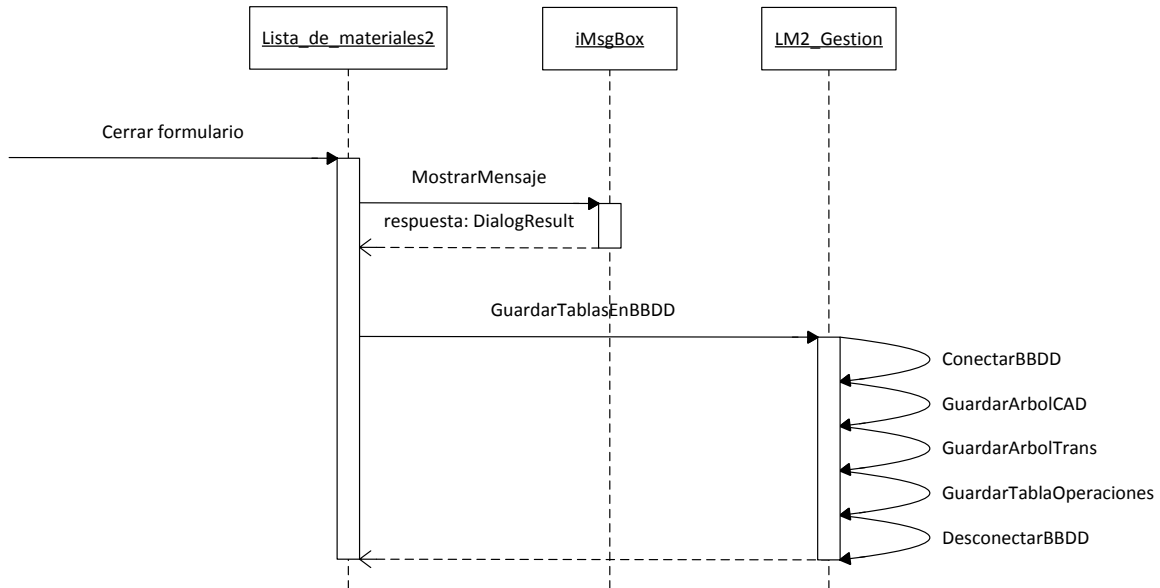
Abrir_Ficha_en_ERP: ver caso de uso Guardar cambios (ejecución en CAD)

InsertarRegistro: ver caso de uso Guardar cambios (ejecución en CAD)

RecargarParametrosCAD: ver caso de uso Guardar cambios (ejecución en CAD)

FinTransaccion: ver caso de uso Guardar cambios (ejecución en CAD)

5.5.15 Caso de uso Guardar tablas en BBDD



Métodos

MostrarMensaje: ver caso de uso Importar material desde ERP

GuardarTablasEnBBDD: Guarda la propuesta actual de lista de materiales en la base de datos de la integración.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

ConectarBBDD: Abre una conexión con la base de datos de la integración empleando las credenciales especificadas en la configuración (caso de uso Configurar BBDD LdM)

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

GuardarArbolCAD: Guarda la tabla dtTablaArbolCAD en la tabla de la base de datos dbo.ArbolCAD

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

GuardarArbolTrans: Guarda la tabla dtTablaArbolTR en la tabla de la base de datos dbo.ArbolTR

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

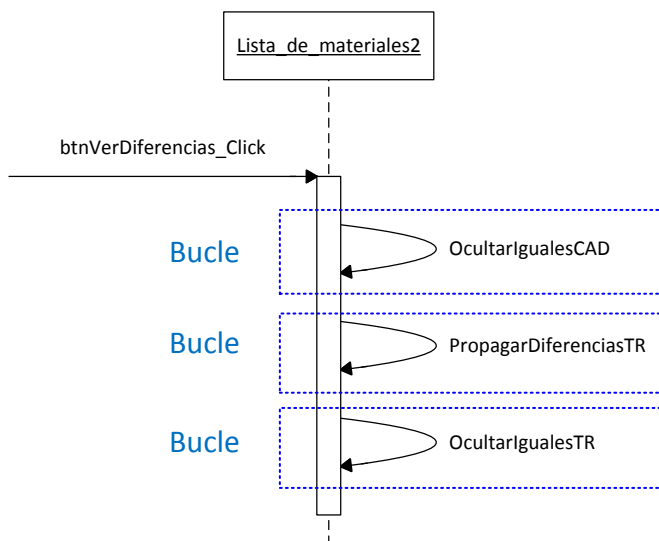
GuardarTablaOperaciones: Guarda la tabla dtTablaOperaciones en la tabla de la base de datos dbo.Operaciones

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

DesconectarBBDD: Cierra la conexión con la base de datos de la integración

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

5.5.16 Caso de uso Comparación CAD-TR



Métodos

btnVerDiferencias_Click: Controla el evento de clic sobre el botón Ver Diferencias

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

OcultarIgualesCAD: Método recursivo para ocultar los nodos del árbol CAD que tengan equivalencia en el árbol de transformación bajo la estructura del componente indicado

- Precondiciones:
- Entrada:
 - NodoPadre (TreeNode) --> nodo del árbol CAD que representa el componente cuyos hijos con equivalencia en TR se ocultan.
- Salida:
- Postcondiciones:

PropagarDiferenciasTR: Método recursivo para buscar componentes que presenten diferencias respecto del árbol CAD en el árbol de transformación.

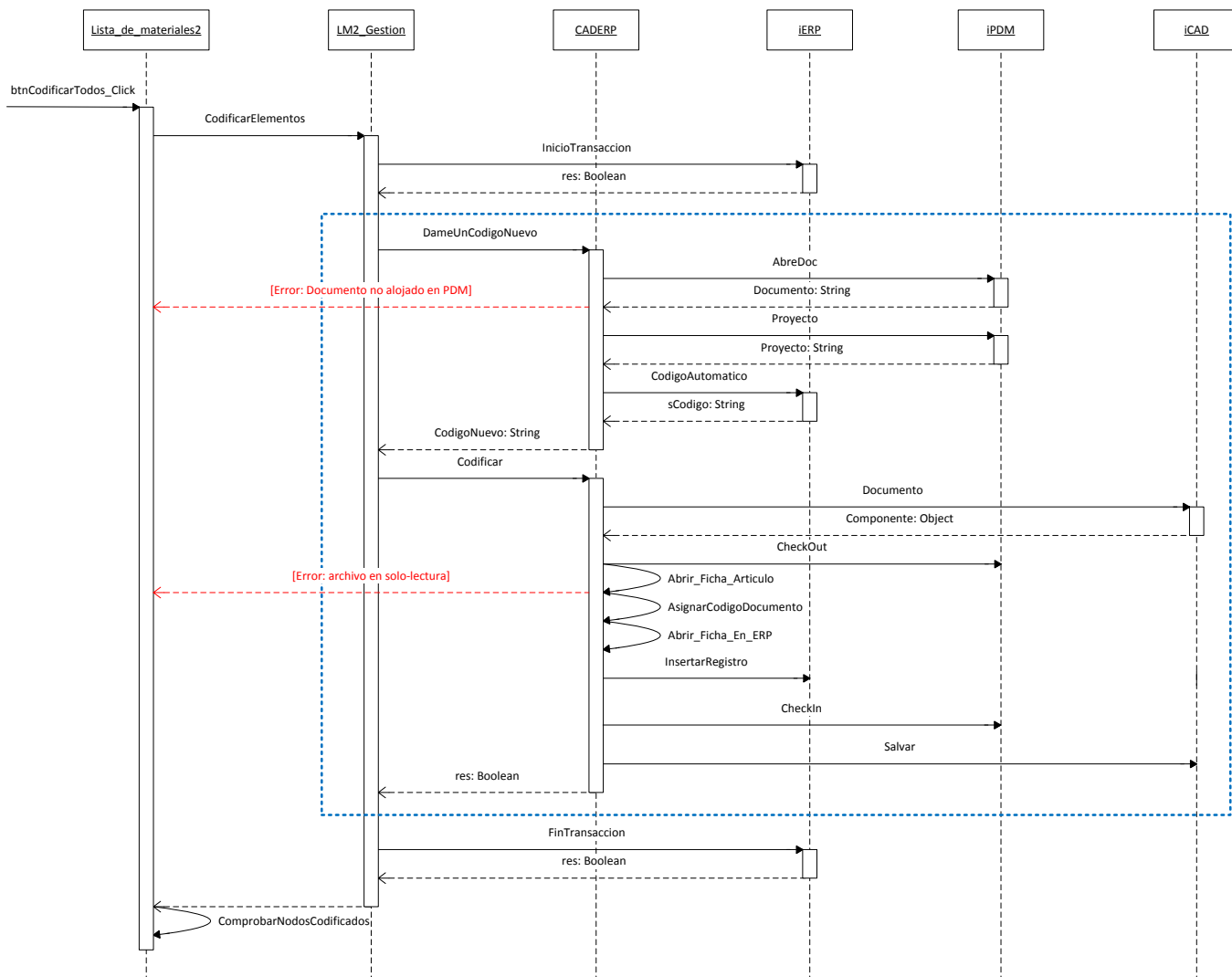
- Precondiciones:
- Entrada:
 - NodoPadre (TreeGridNode) --> nodo del árbol de transformación cuyos hijos heredarán el estado del padre.
- Salida:
 - Verdadero si alguno de los componentes bajo NodoPadre es diferente respecto del árbol CAD, y falso en caso contrario.

- Postcondiciones:

OcultarIgualesTR: Método recursivo para ocultar los nodos del árbol de transformación que tengan equivalencia en el árbol CAD, bajo la estructura

- Precondiciones:
- Entrada:
 - NodoPadre (TreeGridNode) --> nodo del árbol de transformación cuyos hijos con equivalencia en CAD se ocultan
- Salida:
- Postcondiciones:

5.5.17 Caso de uso Codificar Todos (ejecución en CAD)



Métodos

btnCodificarTodos_Click: Controla el evento de clic sobre el botón Codificar Todos

- Precondiciones: Formulario activado; hay elementos sin codificar en el árbol TR
- Entrada:
- Salida:
- Postcondiciones:

CodificarElementos: Da de alta en el ERP los componentes sin codificar en la propuesta de envío de lista de materiales

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

InicioTransaccion: ver caso de uso Guardar cambios (ejecución en CAD)

DameUnCodigoNuevo: Lanza una consulta al ERP para obtener un código que asignar a un artículo

- Precondiciones:
- Entrada:
 - NombreArchivo (String) → ruta de archivo del artículo a codificar
 - Datos (Collection) → colección de datos del artículo a codificar
- Salida:
 - Cadena de caracteres con el código con el código para dar de alta el artículo
- Postcondiciones:

AbreDoc: Devuelve la ruta local de un archivo gestionado por la herramienta PDM. Si no existe una copia local del archivo, la obtiene desde el almacén.

- Precondiciones:
- Entrada:
 - Documento (String) → nombre del archivo
 - Proyecto (String; Opcional) → nombre de la carpeta del almacén PDM bajo la que se halla el archivo. Por defecto, cadena vacía
- Salida:
 - RutaDeGuardado (String; Retorno) → ruta local del archivo
- Postcondiciones:

CodigoAutomatico: Consulta en el ERP por el siguiente código de artículo aún sin asignar

- Precondiciones:
- Entrada:
- Salida:
 - Cadena de caracteres con el siguiente código de artículo disponible según el ERP
- Postcondiciones:

Codificar: Da de alta un artículo en el ERP y asigna el código con el que se registra al archivo 3D correspondiente.

- Precondiciones:
- Entrada:
 - Fichero (String) → ruta de fichero del archivo 3D
 - Datos (Collection) → colección de datos del artículo a dar de alta en el ERP
 - CodigoOriginal (String) → código propuesto desde la integración para registrar el artículo
- Salida:
 - Datos (Collection; Retorno) → colección de datos del artículo tras codificarlo en el ERP
- Postcondiciones:

Documento: ver caso de uso Guardar cambios (ejecución en CAD)

Abrir_Ficha_Articulo: genera una colección de parámetros de relevancia CAD con los datos que corresponden al archivo indicado, y los asigna al artículo en ERP correspondiente según la configuración de relevancia de parámetros establecida.

- Precondiciones:
- Entrada:
 - Fichero (String) → ruta de fichero del archivo CAD
 - Datos (Collection) → colección de parámetros de relevancia CAD del artículo, inicialmente vacía.
- Salida:
 - Datos (Collection; Retorno) → colección de parámetros de relevancia CAD del artículo, con valores asignados.
 - Verdadero si la operación se ha realizado con éxito, falso en caso contrario.
- Postcondiciones:

AsignarCodigoDocumento: Asigna el código de ERP al archivo CAD correspondiente por medio del programa CAD

- Precondiciones:
- Entrada:
 - ModeloCAD (Object) → puntero al objeto COM que representa el archivo CAD
 - Código (String) → código del artículo ERP correspondiente al archivo CAD
- Salida:
- Postcondiciones:

Abrir_Ficha_En_ERP: ver caso de uso Guardar cambios (ejecución en CAD).

Salvar: Guarda los datos modificados sobre el archivo a través del programa CAD

- Precondiciones:
- Entrada:
 - ComponenteCAD (Object; Opcional) → puntero al objeto COM que representa el archivo a guardar. Si se omite, se guarda el modelo activo en el programa CAD.
- Salida:
 - Verdadero si la operación se ha realizado con éxito, falso en caso contrario.
- Postcondiciones:

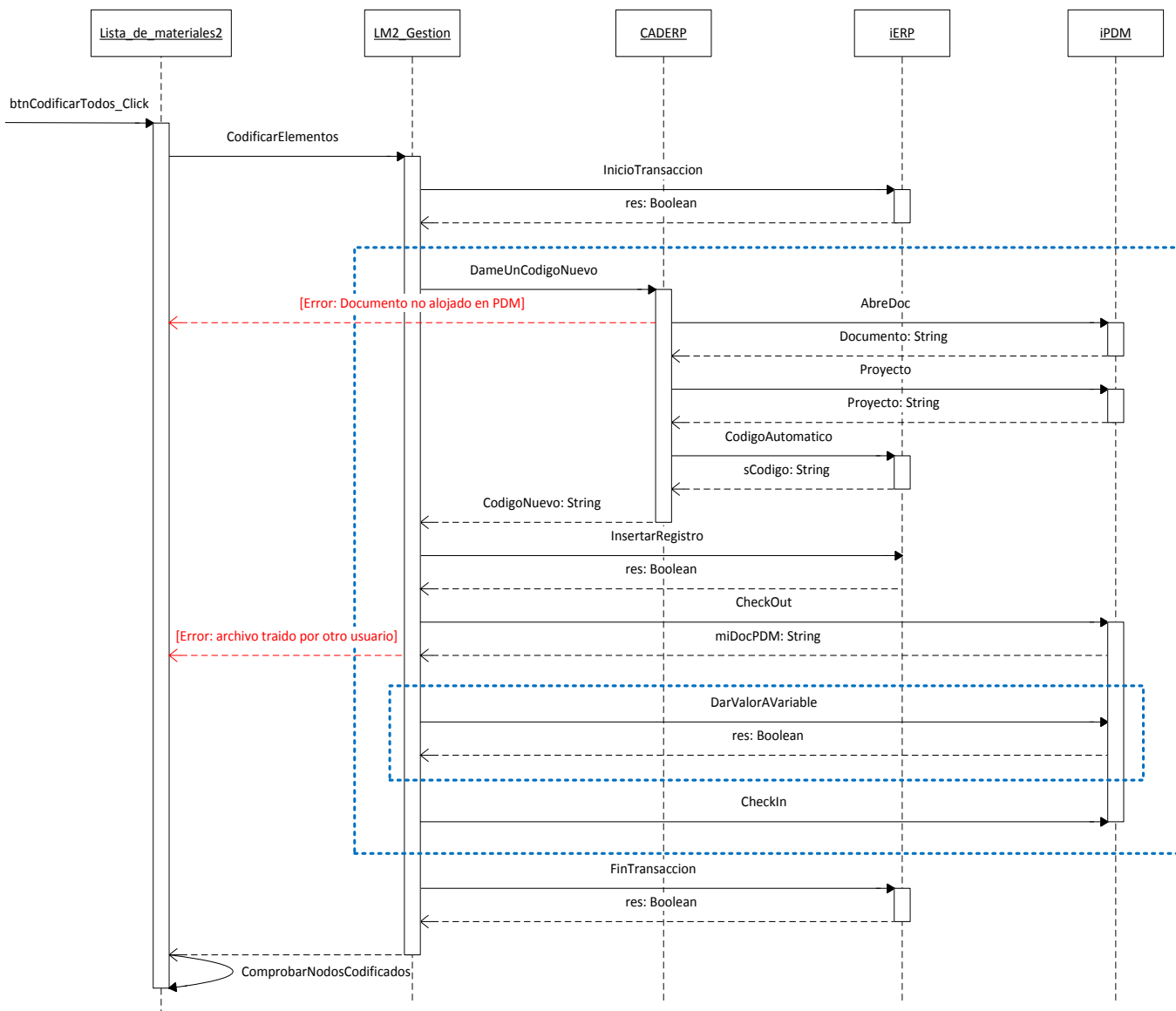
FinTransaccion: ver caso de uso Guardar cambios (ejecución en CAD)

ComprobarNodosCodificados: Método recursivo para asignar el icono correspondiente a los componentes del árbol de transformación bajo el nodo indicado, en función de si están codificados o no.

- Precondiciones:
- Entrada:
 - NodoPadre (TreeGridNode) → nodo del árbol de transformación para cuyos componentes se comprueba el estado de alta en el ERP.

- Salida:
 - Verdadero si existen elementos sin codificar en el árbol de transformación, y falso en caso contrario.
- Postcondiciones:

5.5.18 Caso de uso Codificar Todos (ejecución sin CAD)



Métodos

btnCodificarTodos_Click: ver caso de uso Codificar Todos (ejecución en CAD)

CodificarElementos: ver caso de uso Codificar Todos (ejecución en CAD)

InicioTransaccion: ver caso de uso Guardar cambios (ejecución en CAD)

DameUnCodigoNuevo: ver caso de uso Codificar Todos (ejecución en CAD)

AbreDoc: ver caso de uso Codificar Todos (ejecución en CAD)

CodigoAutomatico: ver caso de uso Codificar Todos (ejecución en CAD)

InsertarRegistro: ver caso de uso Guardar cambios (ejecución en CAD)

CheckOut: ver caso de uso Guardar cambios (ejecución sin CAD)

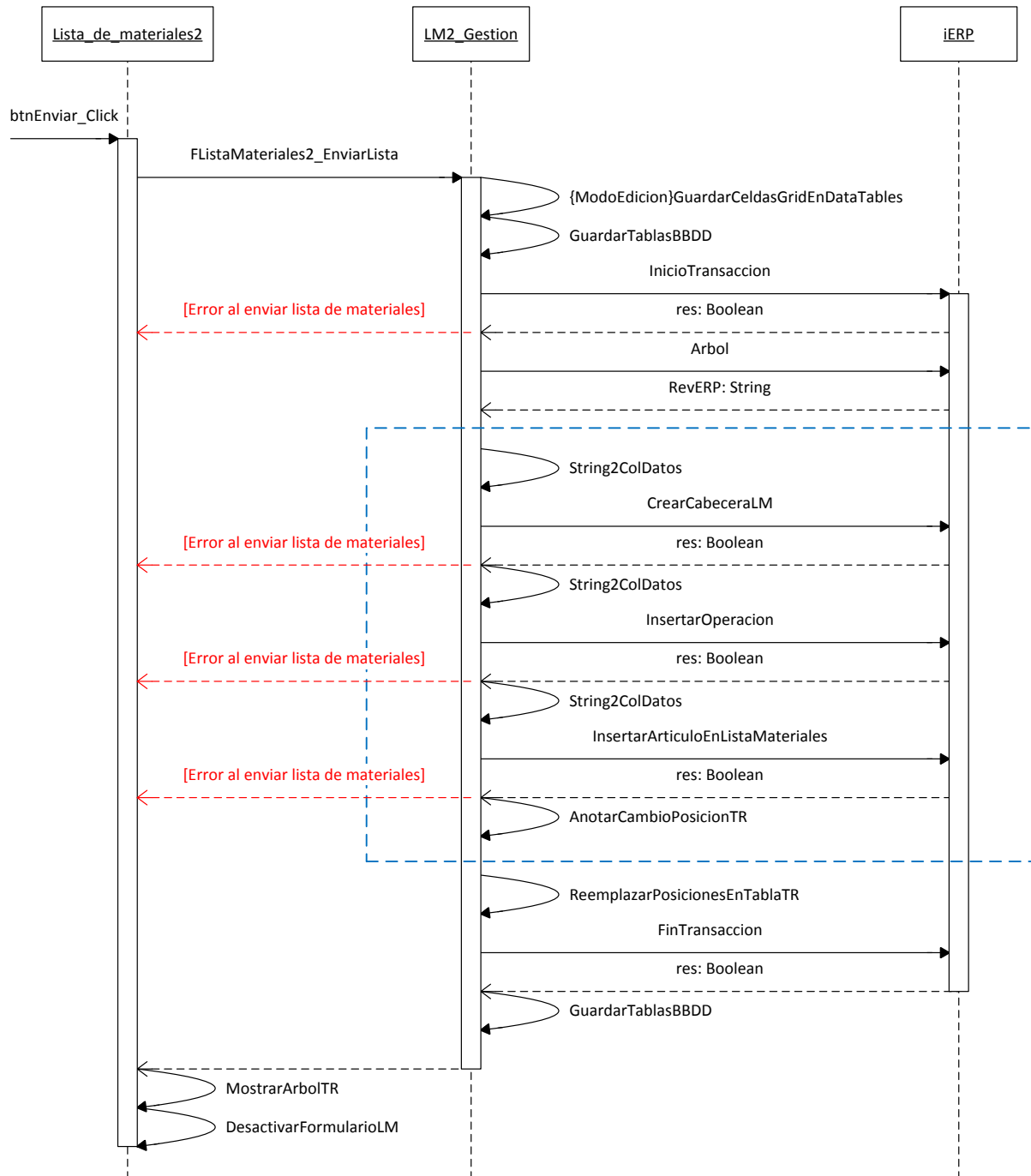
DarValorAVariable: ver caso de uso Guardar Cambios (ejecución sin CAD).

CheckIn: ver caso de uso Guardar Cambios (ejecución sin CAD).

FinTransaccion: ver caso de uso Guardar cambios (ejecución en CAD)

ComprobarNodosCodificados: ver caso de uso Eliminar material

5.5.19 Caso de uso Enviar lista de materiales a estructura



Métodos

btnEnviar_Click: Controla el evento de clic sobre el botón Enviar Lista

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

FListaMateriales2_EnviarLista: Captura el evento lanzado desde el formulario para enviar la lista de materiales, y abre una conexión con el sistema ERP

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

GuardarCeldasDataGridEnDataTables: Guarda los datos de artículos que hayan sido modificados en el modo de edición sobre las tablas de datos dtTablaArbolTR y dtTablaOperaciones

- Precondiciones: Formulario activado en modo de edición
- Entrada:
- Salida:
- Postcondiciones:

GuardarTablasBBDD: Guarda las tablas dtTablaArbolCAD, dtTablaArbolTR y dtTablaOperaciones en sus correspondientes tablas de la base de datos de la integración.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

InicioTransaccion: ver caso de uso Guardar Cambios (ejecución en CAD)

Arbol: Obtiene la estructura de la lista de materiales correspondiente desde el ERP, junto con la revisión del conjunto principal, y vuelca la estructura sobre la tabla dtTablaArbolERP

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

String2ColDatos: Deserializa una cadena de caracteres, devolviendo la colección de datos que contiene.

- Precondiciones:
- Entrada:

- Salida:
- Postcondiciones:

CrearCabeceraLM: Genera y da de alta la raíz de una lista de materiales de estructura de fabricación en el sistema ERP (vacía)

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

InsertarOperacion: Inserta las operaciones de un artículo con estructura bajo el correspondiente artículo en la lista de materiales correspondiente del ERP

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

InsertarArticuloEnListaMateriales: Inserta un artículo en la lista de materiales correspondiente del ERP, bajo la operación indicada de la estructura del componente padre indicado.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

AnotarCambioPosicionTR: Anota en una tabla de datos la posición que el sistema ERP ha asignado al artículo en la lista de materiales

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

ReemplazarPosicionesEnTablaTR: Reasigna las posiciones de los componentes en el árbol de transformación en base a las posiciones que el ERP les ha asignado.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

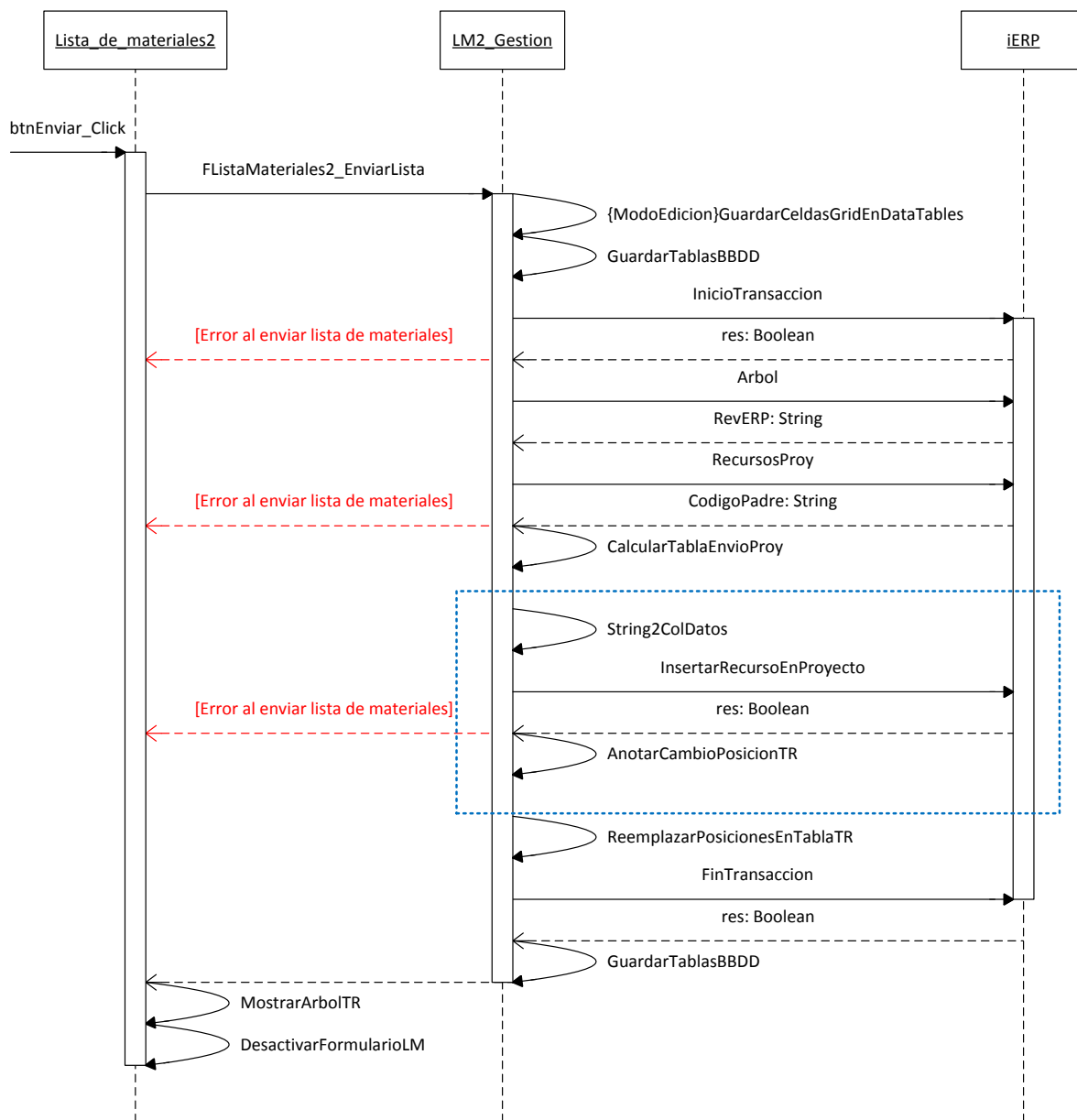
FinTransaccion: ver caso de Guardar Cambios (ejecución en CAD)

MostrarArbolTR: ver caso de uso Reagrupar desde CAD

DesactivarFormularioLM: Desactiva los botones del formulario (FormularioActivado = False)

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

5.5.20 Caso de uso Enviar lista de materiales a proyecto



Métodos

`btnEnviar_Click`: ver caso de uso Enviar lista de materiales a estructura

`FListaMateriales2_EnviarLista`: ver caso de uso Enviar lista de materiales a estructura

`GuardarCeldasDataGridEnDataTables`: ver caso de uso Enviar lista de materiales a estructura

`GuardarTablasBBDD`: ver caso de Guardar Cambios (ejecución sin CAD)

InicioTransaccion: ver caso de Guardar Cambios (ejecución en CAD)

Arbol: ver caso de uso Enviar lista de materiales a estructura

String2ColDatos: ver caso de uso Enviar lista de materiales a estructura

RecursosProy: Obtiene la lista de artículos registrados como recursos del proyecto desde el ERP

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

CalcularTablaEnvioProy: Genera la nueva tabla de recursos del proyecto con los antiguos y los nuevos recursos, actualizando cantidades y posiciones.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

InsertarRecursoEnProyecto: Registra un artículo del ERP como recurso de un proyecto (componente de lista de materiales de proyecto) con la cantidad y posición indicadas

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

AnotarCambioPosicionTR: Genera una tabla con las nuevas posiciones de los recursos insertados en el proyecto, según se han registrado en el ERP

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

ReemplazarPosicionesEnTablaTR: Actualiza la tabla dtTablaArbolTR con las nuevas posiciones de los recursos registradas en el ERP.

- Precondiciones:
- Entrada:
- Salida:
- Postcondiciones:

FinTransaccion: ver caso de Guardar Cambios (ejecución en CAD)

MostrarArbolTR: ver caso de uso Reagrupar desde CAD

DesactivarFormularioLM: ver caso de uso Enviar lista de materiales a estructura

6. Pruebas

6.1 Pruebas de robustez

De forma previa a la entrega del producto al personal de preventa, se realiza una serie de pruebas por cada caso de uso analizado, tratando de detectar el mayor número de errores. Para ello, se estudia el comportamiento de la aplicación cuando se hace un uso deliberadamente incorrecto de los casos de uso.

6.1.1 Agrupar artículos

- Seleccionar componentes de distinto nivel para agrupar
- Seleccionar componentes de distinto código para agrupar
- Seleccionar componentes de distinta unidad de medida para agrupar
- Seleccionar componentes que pertenecen a distintas estructuras para agrupar
- No seleccionar ningún componente para agrupar

6.1.2 Desagrupar artículos

- Seleccionar más de un componente para desagrupar
- Seleccionar conjunto principal para desagrupar
- No seleccionar ningún componente para desagrupar

6.1.3 Importar componente CAD

- Seleccionar componente ficticio para transferir
- Seleccionar componente ya existente en la propuesta para transferir
- Seleccionar componente raíz (conjunto principal) para transferir
- Seleccionar varios componentes para transferir

6.1.4 Importar artículo ERP

- Seleccionar más de un componente bajo el que insertar el artículo
- No seleccionar ningún componente bajo el que insertar el artículo
- Introducir caracteres no válidos (no numéricos) en el paso de introducir la cantidad
- Introducir cantidad no válida para la unidad de medida del artículo (números negativos, números no enteros para unidad de medida = UNIDAD)

6.1.5 Eliminar nodo

- No seleccionar ningún componente para eliminar
- Seleccionar conjunto principal para eliminar

6.1.6 Subir posición

- No seleccionar ningún componente para subir su posición
- Seleccionar más de un componente para subir su posición
- Seleccionar el conjunto principal para subir su posición
- Seleccionar el primer o único componente de una estructura para subir su posición
- Seleccionar un componente cuya posición está bloqueada en el ERP para subir su posición

6.1.7 Bajar posición

- No seleccionar ningún componente para bajar su posición

- Seleccionar más de un componente para bajar su posición
- Seleccionar el conjunto principal para bajar su posición
- Seleccionar el último o único componente de una estructura para bajar su posición
- Seleccionar un componente cuya posición está bloqueada en el ERP para bajar su posición

6.1.8 Asignar posición manual

- No seleccionar ningún componente para cambiar su posición
- Seleccionar más de un componente para cambiar su posición
- Seleccionar el conjunto principal para cambiar su posición
- Seleccionar el único componente de una estructura para bajar su posición
- Seleccionar un componente cuya posición está bloqueada en el ERP para bajar su posición

6.1.9 Importar ruta

- No seleccionar ningún componente para importar su ruta
- Asignar operación de ruta
- No seleccionar ningún componente para importar su ruta

6.2 Errores corregidos

6.2.1 Formulario bloqueado

- Identificado por: Orbinox
- Descripción: El formulario Lista_de_materiales2 se bloquea de forma modal impidiendo utilizar o cerrar el formulario, y evitando el acceso al programa CAD subyacente.
- Causa: La ubicación por defecto del archivo UserSettings.xml (carpeta de instalación de la integración bajo 'Archivos de programa') requiere ejecución con permisos administrativos del programa. Al ejecutar un usuario sin permisos la integración, el formulario se bloquea al tratar de leer del archivo las columnas a mostrar en el árbol de transformación.
- Solución: Ejecutar el programa con permisos de administrador.
 - Línea de mejora: Cambiar la ubicación del archivo UserSettings a una carpeta que no requiera permisos de administrador para la lectura.

6.2.2 No se importan operaciones (RPS2012)

- Identificado por: I+D
- Descripción: Al realizar una primera instalación de la integración en un sistema, no se importan las operaciones por defecto definidas para la estructura en un primer envío de lista de materiales con destino a estructura de fabricación.
- Causa: No existe definida un código de artículo plantilla por defecto en caso de que al conjunto principal de la lista de materiales no le haya sido asignado alguno.
- Solución: Incluir un campo en la configuración para definir un código de artículo plantilla del que importar las operaciones al enviar listas de materiales.

6.3 Análisis de rendimiento

Además del correcto funcionamiento de los métodos, se miden los recursos y tiempo de ejecución de determinados casos de uso a fin de detectar posibles cuellos de botella en la aplicación, y optimizar el código en consecuencia.

7. Seguimiento

7.1 Desviación respecto a lo planificado

Debido a requerimientos de la empresa, se dio comienzo al análisis y desarrollo de nuevas funcionalidades de forma previa al desarrollo de la funcionalidad principal:

- Se determinó que la aplicación debía contar con recursos multilinguaje; el desarrollo de la nueva funcionalidad de listas de materiales debía seguir esta metodología.
- La gestión de la calidad (cotas de inspección y pautas de recepción) debía ser implementada de forma previa a la nueva funcionalidad de listas de materiales.

Asimismo, durante el desarrollo de la nueva funcionalidad de listas de materiales, se identificaron ciertas potencialidades a incorporar en otros módulos de la integración; estas nuevas funcionalidades se han llevado a cabo solapándose en el tiempo con el desarrollo de la funcionalidad principal:

- Mantenimiento de posiciones de materiales y operaciones en reenvíos a estructuras
- Gestión de entidades extendidas en RPS2012

7.1.1 Desglose final de tareas

A continuación se presenta la tabla de tareas realizadas durante el ciclo de vida del proyecto; la duración de las tareas no refleja el total de horas dedicadas cada una. Debido a que varias tareas se solapan en el tiempo, se ha dedicado tiempo de desarrollo a distintas tareas en una misma jornada.

Nombre de tarea	Duration	Start	Finish
PLANIFICACION	5 days	Fri 12/10/12	Fry 19/11/12
DESARROLLO	144,88 days	Mon 15/10/12	Sun 05/05/13
Desarrollo de funcionalidades iniciales	30,63 days	Mon 22/10/12	Sun 02/12/12
Desarrollo de nuevas funcionalidades	144,88 days	Mon 15/10/12	Sun 05/05/13
Multilinguaje en interfaz de la aplicación	29,13 days	Mon 15/10/12	Thu 22/11/12
Generar tablas de traducción	2,13 days	Mon 15/10/12	Tue 16/10/12
Generar archivos de recursos	1,13 days	Tue 16/10/12	Thu 18/10/12
Parametrizar idiomas disponibles en configuración	1,13 days	Wed 17/10/12	Wed 17/10/12
Cargar textos de interfaz según cultura en uso	9 days	Wed 17/10/12	Wed 31/10/12
Traducción del manual de usuario	2,13 days	Mon 22/10/12	Tue 23/10/12
Traducción del manual de configuración	2,13 days	Tue 20/11/12	Thu 22/11/12
Pautas para control de calidad	64,13 days	Thu 25/10/12	Tue 22/01/13
Captura de cotas de inspección desde SW	7 days	Mon 29/10/12	Tue 06/11/12
Diseñar contenedor para visualizar cotas y pautas	13 days	Thu 25/10/12	Mon 12/11/12
Enlazar captura de cotas al control	4 days	Wed 07/11/12	Mon 12/11/12
Servicio conector: InfoPauta	2 days	Mon 14/01/13	Tue 15/01/13

Servicio conector: InsertarPauta	3 days	Wed 16/01/13	Fri 18/01/13
Enlazar captura e insercion de pautas al control	2 days	Mon 21/01/13	Tue 22/01/13
Capturar nombre de archivo relacionado en PDM	2,13 days	Wed 07/11/12	Thu 08/11/12
Definir propiedad en interfaz iCAD	1 day	Wed 07/11/12	Wed 07/11/12
Definir propiedad en interfaz iPDM	1 day	Wed 07/11/12	Wed 07/11/12
Implementar métodos auxiliares en proyecto iPDM	2 days	Wed 07/11/12	Thu 08/11/12
Controlar tamaño de archivo de traza	1,13 days	Tue 06/11/12	Wed 07/11/12
Mantenimiento de operaciones y posiciones en LdM	6,13 days	Tue 13/11/12	Tue 20/11/12
Mantener operaciones de estructura en reenvíos	4 days	Mon 12/11/12	Thu 15/11/12
Mantener posiciones de componentes en reenvíos	3 days	Fri 16/11/12	Tue 20/11/12
Entidades extendidas en RPS2012	7,13 days	Tue 27/11/12	Wed 05/12/12
Parametrización de entidades extendidas	0 days	Tue 27/11/12	Tue 27/11/12
Modificar métodos del conector	7,13 days	Tue 27/11/12	Wed 05/12/12
Metodos de lectura	2 days	Tue 27/11/12	Wed 28/11/12
Metodos de insercion	4 days	Fri 30/11/12	Wed 05/12/12
Capturar ID de archivo en PDM	1 day	Wed 19/12/12	Wed 19/12/12
Llamadas multiples a metodos del servicio RPS	7,88 days	Tue 11/12/12	Thu 20/12/12
Métodos de escritura	3 days	Tue 11/12/12	Thu 13/12/12
Métodos de lectura	4 days	Mon 17/12/12	Thu 20/12/12
Capturar ID de archivo en PDM	1 day	Wed 19/12/12	Wed 19/12/12
Siguiente Revision PDM	3 days	Fri 25/01/13	Wed 30/01/13
Codificacion manual RPS2012	2 days	Tue 29/01/13	Wed 30/01/13
Filtrar parametros en formulario articulo	1 day	Thu 31/01/13	Thu 31/01/13
Verificación de número de serie / licencia para servicio RPS2012	5,13 days	Wed 30/01/13	Tue 05/02/13
Verificar número de serie de CAD (SW)	4 days	Wed 30/01/13	Mon 04/02/13
Autorizar servidores de acceso libre mediante dirección MAC	2 days	Mon 04/02/13	Tue 05/02/13
Multiempresa en conector RPS2012	2,88 days	Tue 05/02/13	Thu 07/02/13
Modificar funciones de consulta	2 days	Tue 05/02/13	Wed 06/02/13
Nueva funcion InsertarArticleDetail	2 days	Wed 06/02/13	Thu 07/02/13
Modificar funciones de insercion	1 day	Thu 07/02/13	Thu 07/02/13
Obtener ID de entidad referenciada a partir de código	1 day	Mon 11/03/13	Mon 11/03/13
Módulo iSE	14 days	Tue 16/04/13	Fri 03/05/13
Métodos	14 days	Tue 16/04/13	Fri 03/05/13
Propiedades	13 days	Tue 16/04/13	Thu 02/05/13
Módulo iWIN	23 days	Tue 02/04/13	Thu 02/05/13

Métodos	23 days	Tue 02/04/13	Thu 02/05/13
Propiedades	23 days	Tue 02/04/13	Thu 02/05/13
SEGUIMIENTO	166,88 days	Fri 12/10/12	Fri 31/05/13
Seguimiento del alcance	144,13 days	Fri 12/10/12	Tue 30/04/13
Histórico de funcionalidades implementadas	144,13 days	Fri 12/10/12	Tue 30/04/13
Análisis de nuevas funcionalidades	118,88 days	Fri 12/10/12	Tue 26/03/13
Multilinguaje en interfaz de la aplicación	1,88 days	Fri 12/10/12	Sun 14/10/12
Controlar tamaño de archivo de traza	1,13 days	Thu 18/10/12	Thu 18/10/12
Pautas para control de calidad	67,13 days	Mon 22/10/12	Tue 22/01/13
Cotas de inspección CAD	2 days	Mon 22/10/12	Tue 23/10/12
Pautas de control ERP	63 days	Tue 23/10/12	Thu 17/01/13
Capturar nombre de archivo relacionado en PDM	1,13 days	Wed 07/11/12	Wed 07/11/12
Mantenimiento de operaciones y posiciones en LdM	1,13 days	Mon 12/11/12	Mon 12/11/12
Entidades extendidas en RPS2012	21,88 days	Tue 27/11/12	Wed 26/12/12
Parametrización de entidades extendidas	1 day	Wed 26/12/12	Wed 26/12/12
Modificar metodos del conector	1 day	Tue 27/11/12	Tue 27/11/12
Capturar ID de archivo en PDM	1 day	Tue 18/12/12	Tue 18/12/12
Llamadas multiples a metodos del servicio RPS	1 day	Mon 03/12/12	Mon 03/12/12
Servicio 'Nuevo proyecto'	6 days	Fri 21/12/12	Fri 28/12/12
Adapter Nuevo proyecto	2 days	Thu 03/01/13	Fri 04/01/13
Siguiente Revision PDM	3 days	Wed 23/01/13	Fri 25/01/13
Codificacion manual RPS2012	1 day	Mon 28/01/13	Mon 28/01/13
Filtrar parametros en formulario articulo	3 days	Mon 28/01/13	Wed 30/01/13
Verificación de número de serie / licencia para servicio RPS2012	1 day	Tue 29/01/13	Tue 29/01/13
Trabajo con grupos empresariales RPS2012	1 day	Mon 04/02/13	Tue 05/02/13
Obtener ID de entidad referenciada a partir de código	1 day	Fri 08/03/13	Fri 08/03/13
Nuevo módulo de tipo il_PDM: iWIN	3,13 days	Tue 12/03/13	Thu 14/03/13
Investigación: almacenamiento de datos en flujo alternativo de datos			
Nuevo módulo de tipo il_CAD: iSE	7,13 days	Mon 18/03/13	Tue 26/03/13
Seguimiento del tiempo	143,88 days	Fri 09/11/12	Wed 29/05/13
Actualizar cronograma	1 day	Fri 09/11/12	Fri 09/11/12
Actualizar cronograma	1 day	Fri 16/11/12	Fri 16/11/12
Actualizar cronograma	1 day	Wed 05/12/12	Wed 05/12/12
Actualizar cronograma	1,13 days	Thu 20/12/12	Thu 20/12/12
Actualizar cronograma	1,13 days	Tue 12/03/13	Wed 13/03/13
Actualizar cronograma	1,13 days	Wed 29/05/13	Wed 29/05/13

Seguimiento de comunicaciones	140 days	Tue 13/11/12	Tue 28/05/13
Reunion con tutor	1 day	Tue 13/11/12	Wed 14/11/12
Reunion con tutor	1 day	Sun 10/03/13	Mon 11/03/13
Reunion con tutor	1 day	Mon 27/05/13	Tue 28/05/13
Control de riesgos	123,13 days	Mon 05/11/12	Thu 25/04/13
Enfermedad MV	1 day	Mon 05/11/12	Mon 05/11/12
Fallo del servidor de control de código	1 day	Tue 13/11/12	Tue 13/11/12
Fallo del equipo remoto de desarrollo	1 day	Thu 15/11/12	Thu 15/11/12
Fallo del servidor de control de código	1,13 days	Tue 29/01/13	Wed 30/01/13
Fallo del equipo remoto de desarrollo	3,13 days	Wed 20/03/13	Fri 22/03/13
Enfermedad MV	1,13 days	Wed 24/04/13	Thu 25/04/13
Seguimiento de calidad	465,38 days	Mon 05/11/12	Fri 26/04/13
SINTESIS	24,13 days	Sun 26/05/13	Wed 26/06/13

7.1.2 Gantt planificado vs Gantt real

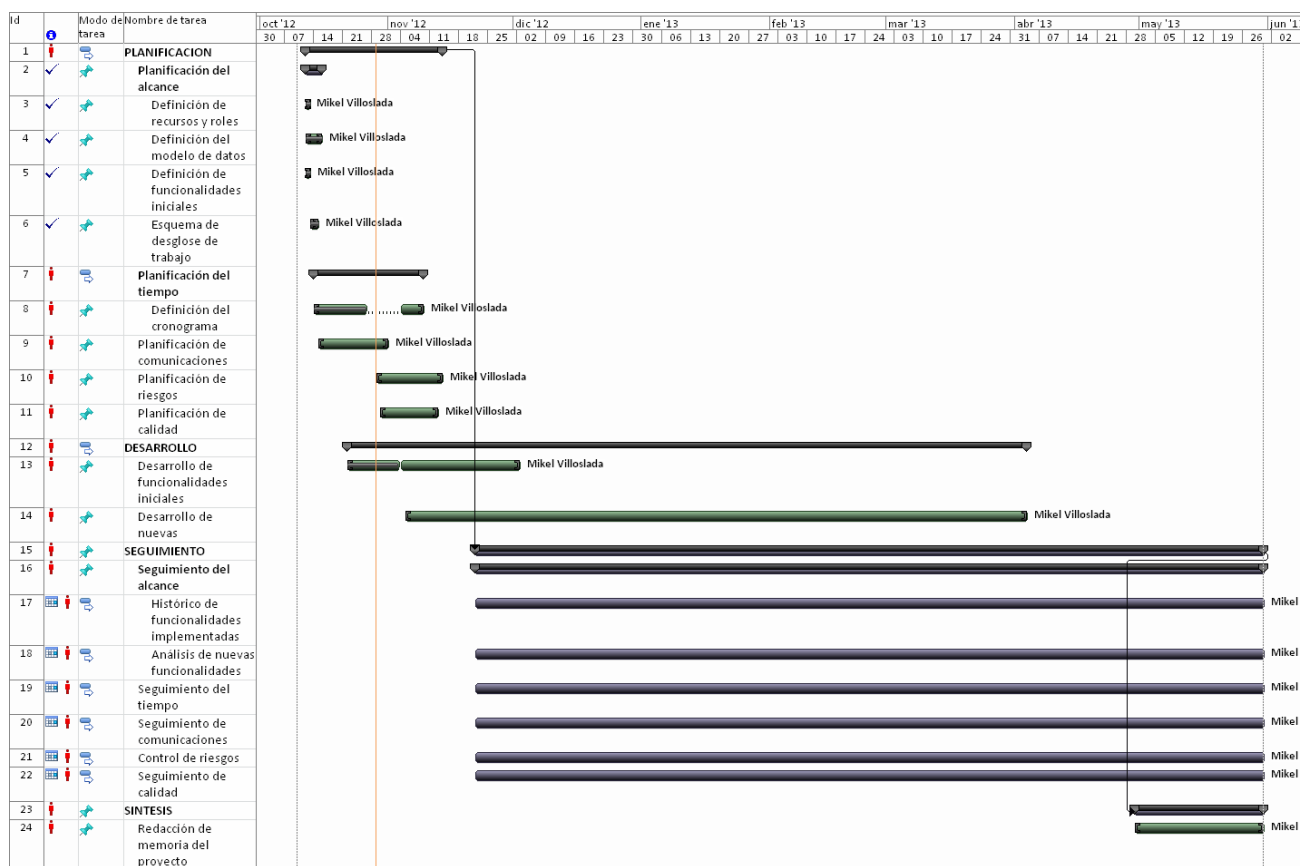


Ilustración 29: Gantt planificado

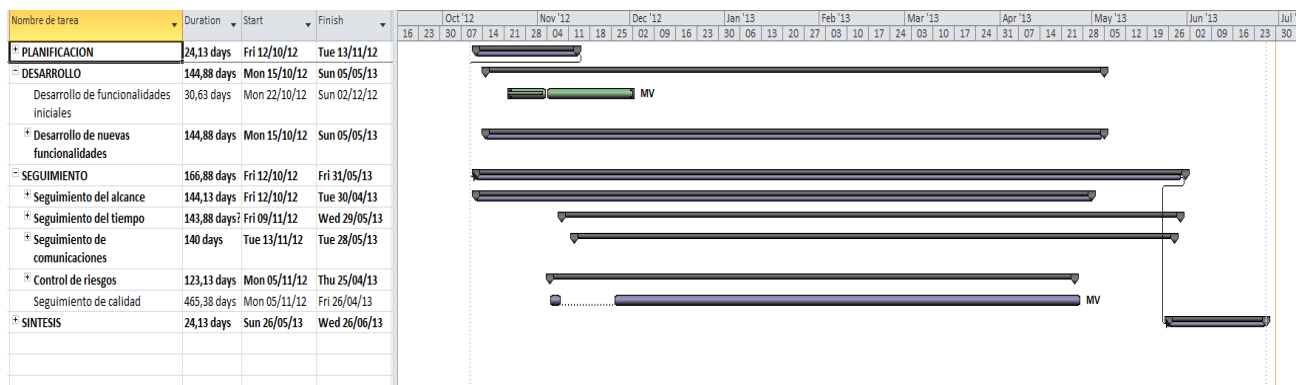


Ilustración 30: Gantt realizado (plegado)

7.2 Riesgos activados

Riesgo	Impacto	Contingencia
Enfermedad (14-XI-2012): Indisposición a causa de fiebre elevada del recurso MV	Muy leve; se retrasó por una jornada una de las tareas de investigación.	Pospuesta tarea de investigación al día siguiente.
Fallo técnico (15-XI-2012): Fallo del servidor de control de código fuente.	Muy leve; los cambios en el código fuente se llevaron a cabo de forma local en el equipo de cada desarrollador.	Pospuestas las actualizaciones de código hasta que el servidor volviese a estar disponible.
Fallo técnico (15-XI-2012 → 16-XI-2012): Fallo del servidor de equipos remotos de desarrollo.	Medio; los desarrollos se llevaron a cabo en el equipo de cada desarrollador, que no emulan fielmente las condiciones del sistema del cliente.	Realizadas pruebas de los desarrollos locales en los equipos remotos una vez que el servidor estuvo disponible.
Fallo técnico (29-I-2013): Fallo del servidor de control de código fuente.	Muy leve; los cambios en el código fuente se llevaron a cabo de forma local en el equipo de cada desarrollador.	Pospuestas las actualizaciones de código hasta que el servidor volviese a estar disponible.
Fallo técnico (20-III-2013): Fallo del servidor de equipos remotos de desarrollo.	Leve; los desarrollos se llevaron a cabo en el equipo de cada desarrollador, que no emulan fielmente las condiciones del sistema del cliente.	Realizadas pruebas de los desarrollos locales en los equipos remotos una vez que el servidor estuvo disponible.
Enfermedad (24-IV-2013): Indisposición a causa de indigestión del recurso MV	Leve; se retrasó por media jornada una de las tareas de desarrollo.	Pospuesta tarea de desarrollo al día siguiente.

7.3 Resumen de reuniones

Fecha	Concepto
19-XI-2012	Revisar documento de objetivos del proyecto y planificación.
11-III-2013	Revisar alcance del proyecto
27-V-2013	Revisar esbozo de memoria del proyecto.

7.4 Control de calidad

La calidad mínima del producto se ha asegurado mediante la estabilidad y robustez del programa, en base a las pruebas realizadas y recogidas en el aparatado de pruebas. El producto final ha sido dotado de valor añadido por medio de optimizaciones de código que han incrementado el rendimiento original del programa (ver apartado 6: Pruebas)

8. Resultados y conclusiones

8.1 Resultados

El producto ha sido implantado en diversos clientes, la mayoría de los cuales han confirmado que su productividad se ha visto aumentada en gran medida gracias a la herramienta. Gran parte de estos clientes han hecho hincapié en la ganancia en tiempo que han notado por medio de la nueva funcionalidad de listas de materiales, la cual les ha permitido integrar conjuntos de hasta 2000 piezas en sus respectivos sistemas ERP con rapidez.

Adicionalmente, se han esbozado e implementado diversas nuevas funcionalidades del producto, de las cuales se destacan las siguientes por su relevancia:

- **Multilinguaje:** Al dotar a la aplicación de nuevos idiomas de presentación, mas la posibilidad de agregar idiomas adicionales en el futuro, el resultado es un producto software globalizado, capaz de ser implantado en un ámbito internacional.
- **Módulo iWIN:** De forma previa al desarrollo de este módulo, la integración era capaz únicamente de interactuar con sistemas dotados de la tecnología PDM de SolidWorks. Mediante este nuevo módulo, la integración puede desplegarse en un amplio rango de sistemas Windows sin necesidad de la herramienta Enterprise PDM.
- **Módulo iSE:** De forma previa al desarrollo de este módulo, la integración era capaz únicamente de interactuar con el programa CAD SolidWorks, de Dassault Systemes. Con el nuevo módulo iSE, se posibilita la interacción con un programa CAD adicional, el popular Solid Edge de Siemens, con el cual cuentan gran cantidad de empresas de ingeniería.

8.2 Conclusiones

De entre las conclusiones personales más relevantes extraídas por el alumno durante el proyecto, se encuentra la importancia de un buen funcionamiento del equipo de desarrollo. Es importante contar con un modelo eficiente de desarrollo en función de la naturaleza del producto a realizar y de las necesidades del cliente, y trasladar este modelo al nuevo personal que se incorpora al equipo.

No obstante, tal y como el alumno ha podido observar, si los preceptos de la metodología entran en conflicto con necesidades apremiantes de un cliente, no se deben desatender dichas necesidades. Si la metodología carece de margen para las maniobras requeridas por estas necesidades, se debe replantear la estrategia de desarrollo (p.e, priorizar tareas de implantación sobre tareas de investigación).

8.2.1 Lecciones aprendidas

A lo largo del ciclo de vida del proyecto, el alumno ha podido obtener conocimiento de primera mano relativo al desarrollo de soluciones software para el ámbito empresarial:

- Contar con la metodología de trabajo adecuada
- Dar pronta respuesta a incidencias y preguntas de clientes
- Anotar potencialidades detectadas durante el desarrollo
- Congeniar con el personal

9. Bibliografía

Sitios web oficiales:

- Microsoft Developer Network:
<http://msdn.microsoft.com/es-es/>
- Soporte de API de SolidWorks:
http://www.solidworks.es/sw/support/1835_ESN_HTML.htm
- Soporte de API de Solid Edge:
http://dl2.plm.automation.siemens.com/solidedge/api/sesdk_web/webframe.html

Foros de desarrolladores:

- Code Project (documentación sobre desarrollos en .NET):
<http://www.codeproject.com/>
- Stack Overflow (documentación sobre desarrollos en .NET):
<http://stackoverflow.com>
- Foro de Jason Newell (para documentación sobre add-ins de Solid Edge):
<http://www.jasonnewell.net/>

Ibermática S.A: Varios manuales y recursos sobre Integración ERP y RPS2012

Anexos

A1. Análisis de funcionalidades adicionales

A1.1 Multilinguaje

Para potenciar la comercialización de la aplicación en el ámbito internacional, se ha decidido que los textos mostrados en la interfaz gráfica de la aplicación deben contar con traducciones correspondientes al idioma corporativo de cada cliente.

Para ello, se han definido los siguientes idiomas a utilizar en la aplicación:

- Español
- Inglés
- Alemán

El programa de configuración deberá contar con un desplegable para que el usuario pueda escoger el idioma en uso entre los definidos, o utilizar el idioma instalado en el ordenador, si este se cuenta entre los definidos. Por defecto, la integración mostrará los textos en Inglés.

Aquellos textos definidos por el usuario (nombres de propiedades, etiquetas personalizadas, etc.) quedan exentos de traducción, dado que no se ubican en el código fuente de la aplicación, sino en bases de datos y archivos xml.

Se contempla la posibilidad de añadir más idiomas en el futuro.

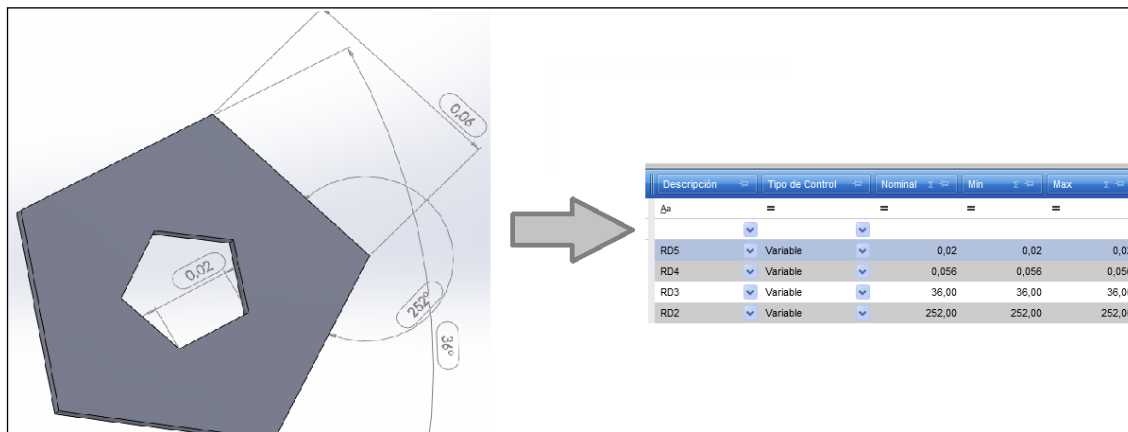
Ámbito: Toda la Integración

A1.2 Gestión de calidad

Para gestionar la calidad de ciertos artículos diseñados en programas CAD, se ha planificado que la integración cuente con la capacidad de obtener información relativa a cotas de dichos artículos. La magnitud de estas cotas se cotejará con pautas de control definidas en RPS2012.

Para la gestión de calidad en la integración, se trabajará con pautas de recepción, trabajando con los valores nominales, máximos y mínimos tanto de las cotas como de las características de las pautas, así como un valor textual identificativo.

La aplicación deberá ser capaz de extraer las cotas definidas en el modelo CAD (método en interfaz iCAD), así como de adecuarlas a un formato capaz de ser insertado como características de pauta en RPS2012. Las cotas del modelo y las pautas definidas para el artículo correspondiente podrán ser visualizadas en el formulario de artículo de la integración.



31. Cotas de inspección (izda) y pautas de recepción (dcha)

Ámbito:

- il_CAD : extracción de cotas
- Declaraciones : nuevo control ucTabCalidad, conversión de cotas a pautas
- il_ERP (iRPS2012): inserción de pautas

A1.3 Archivo asociado en PDM

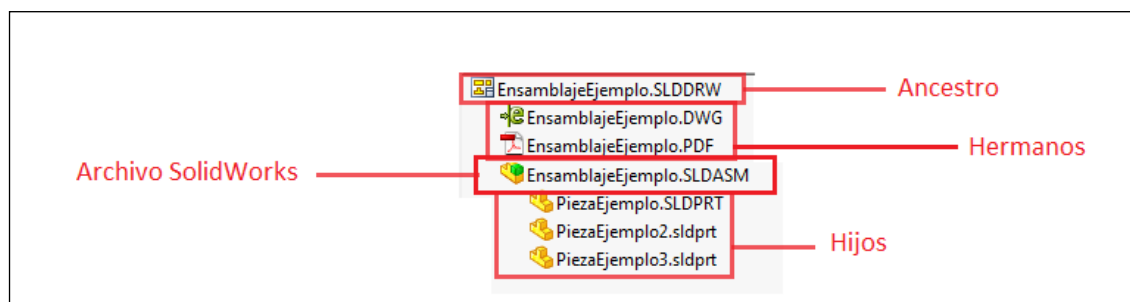
PDM gestiona, por cada archivo de SolidWorks, una estructura arborescente de referencias a archivos adicionales (planos, PDF, etc.) con los que guarda relación. El objetivo de esta nueva funcionalidad es, dado un archivo SolidWorks, obtener la ruta de acceso a un archivo asociado determinado, según extensión de archivo y una prioridad predefinida.

En la estructura de referencias, el plano correspondiente al archivo se ubica en el nivel inmediatamente superior al archivo (es decir, se considera un 'padre' del archivo). Los archivos PDF generados a partir de dicho plano se ubican en el nivel inmediatamente inferior al plano (es decir, se consideran 'hermanos' del archivo SolidWorks). Por último, cabe la posibilidad de que el usuario introduzca a mano referencias a otros archivos; estas referencias se ubican en el nivel inmediatamente inferior al archivo SolidWorks (es decir, se consideran 'hijos' del archivo SolidWorks).

La prioridad según la cual se debe efectuar la búsqueda es la siguiente:

1. Documentos de nivel inferior, o hijos (habitualmente referenciados por el usuario).
2. Documentos de nivel superior o padres (comúnmente planos generados desde SW).
3. Documentos del mismo nivel, o hermanos (habitualmente PDF generados desde el plano).

Si se omite la extensión a la hora de realizar la búsqueda, se tomará por defecto la extensión del plano del archivo SolidWorks (.SLDDRW)



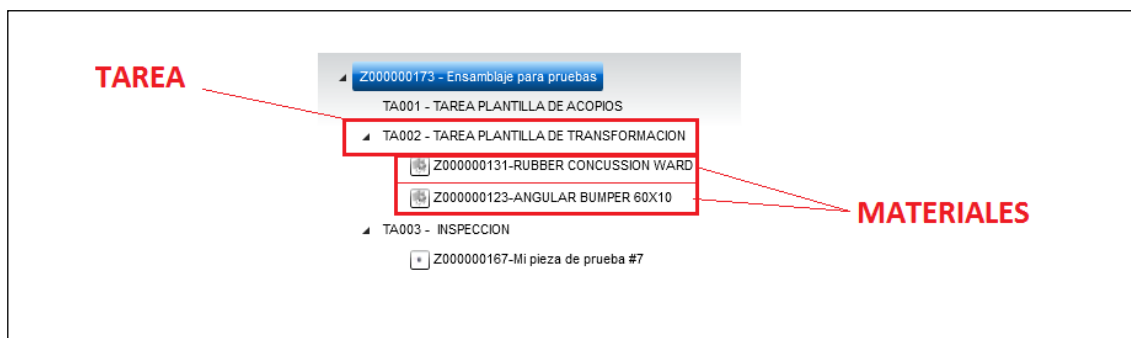
32: Estructura de referencias de un documento PDM

Ámbito:

- il_CAD (iSW): Captura del nombre de documento;
- il_PDM (iPDME): Captura de los nombres de archivo asociados

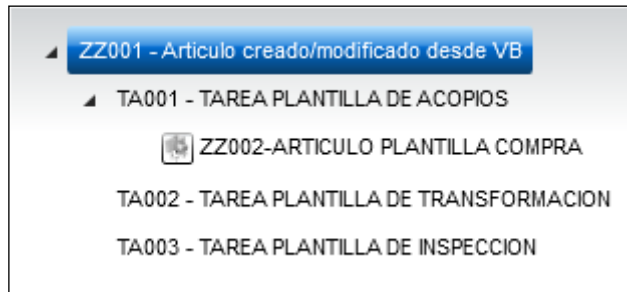
A1.4 Mantenimiento de operaciones y posiciones en listas de materiales (estructuras)

En RPS2012, las listas de materiales de estructuras de fabricación están organizadas de tal manera que los nodos de componentes están siempre asignados a nodos de tarea / operación.



33: Tareas y materiales de estructura

La forma de trabajar de la integración es a través de plantillas, donde para la creación de una estructura se toman como modelo las operaciones de un artículo plantilla. Los datos de estas operaciones pueden ser después modificados por el usuario, así como asignar distintas operaciones a los componentes de la estructura.



34: Artículo plantilla, vista de estructura

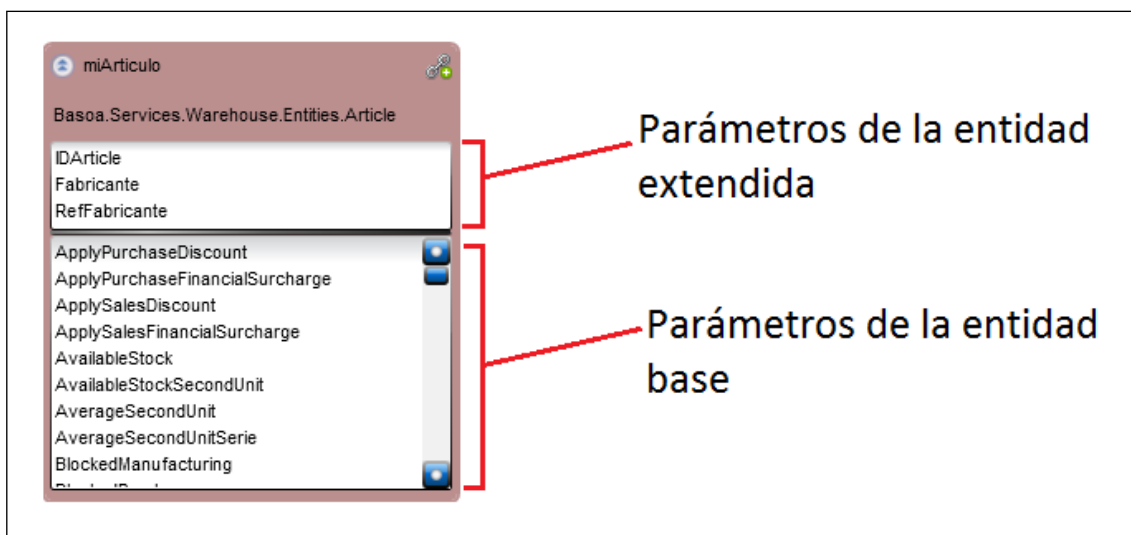
Por otra parte, los componentes cuentan con un campo denominado 'Numero de secuencia', que indica la posición que estos ocupan en la lista de materiales. La integración asigna estas posiciones en base la posición que los componentes ocupan en la estructura extraída desde SolidWorks.

El objetivo de esta funcionalidad es tomar como modelo la revisión de la estructura activa en RPS2012 para asignar a los componentes de la propuesta de envío actual (si se corresponden con los de la estructura activa) la misma operación y posición que ocupan en la revisión activa en RPS2012. Si no existe una estructura activa (no se ha realizado ningún envío previo) o los componentes han cambiado, se tomarán las operaciones de la plantilla y se mantendrán las posiciones por defecto.

Ámbito: il_ERP (iRPS2012)

A1.5 Entidades extendidas en RPS2012

La información de RPS2012 se almacena en una base de datos estructurada según un modelo de entidades. El cliente de RPS2012 permite al usuario definir entidades personalizadas partiendo de una entidad ya existente. A estas entidades definidas por el usuario se las denomina Entidades Extendidas; a las entidades predefinidas se las denomina en adelante como Entidades Base.



35: Ejemplo de entidad extendida a partir de STKArticle

El objeto conector que la integración utiliza para la comunicación con RPS2012 se vale de un servicio web de intercambio de mensajes. El objetivo de la nueva funcionalidad es adaptar la integración y el conector para acomodar los cambios realizados en el servicio para la gestión de entidades extendidas.

Planteamiento:

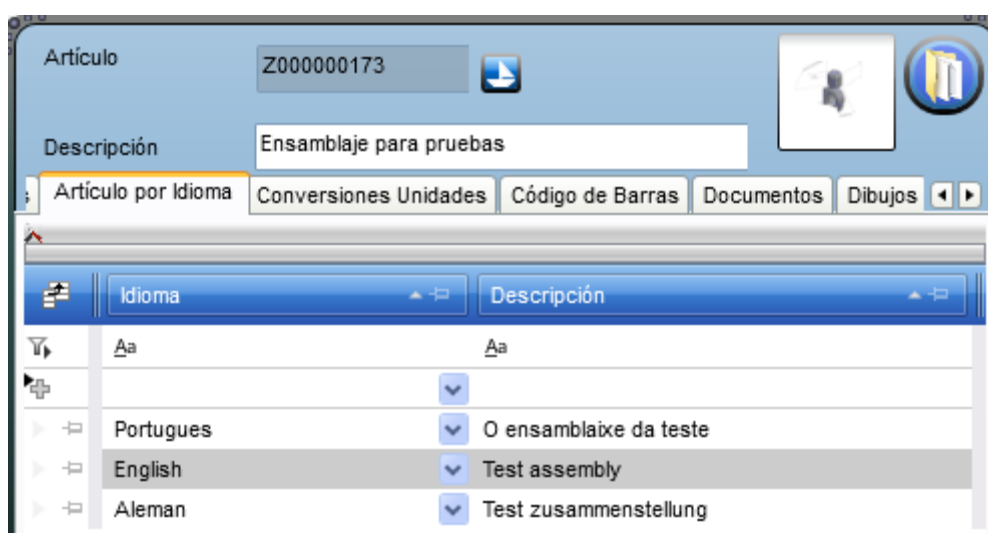
En la configuración, en la sección ERP de RPS2012, se definen los parámetros de validez para artículos, componentes de estructura y materiales de proyectos. Dado que las funciones del servicio han sido ampliadas para buscar propiedades de entidades extendidas, se debe incluir una nueva columna en las definiciones para indicar la entidad base que ha sido extendida.

Asimismo, las llamadas a los métodos del conector que se realicen desde la integración deben ser modificadas para incluir la entidad base cuando las propiedades a tratar pertenezcan a entidades extendidas.

Ámbito: il_ERP (iRPS2012)

A1.6 Llamadas múltiples a métodos del conector RPS2012

La información correspondiente a artículos, componentes y materiales con que trabaja la integración se almacena en una base de datos organizada en base a un modelo de entidades. Un mismo artículo, al tratarse de una entidad (así como componente de estructura, o material de proyecto) puede tener definidos datos relevantes en distintos valores de una misma propiedad de entidad.



36: Descripciones de artículo según idioma

Dado que los métodos de consulta deben devolver valores en una única fila, y los métodos de inserción actualizan o insertan filas de manera unitaria (una por llamada), se requiere modificar el tratamiento de los parámetros al construir la llamada a los métodos del servicio para contemplar esta casuística.

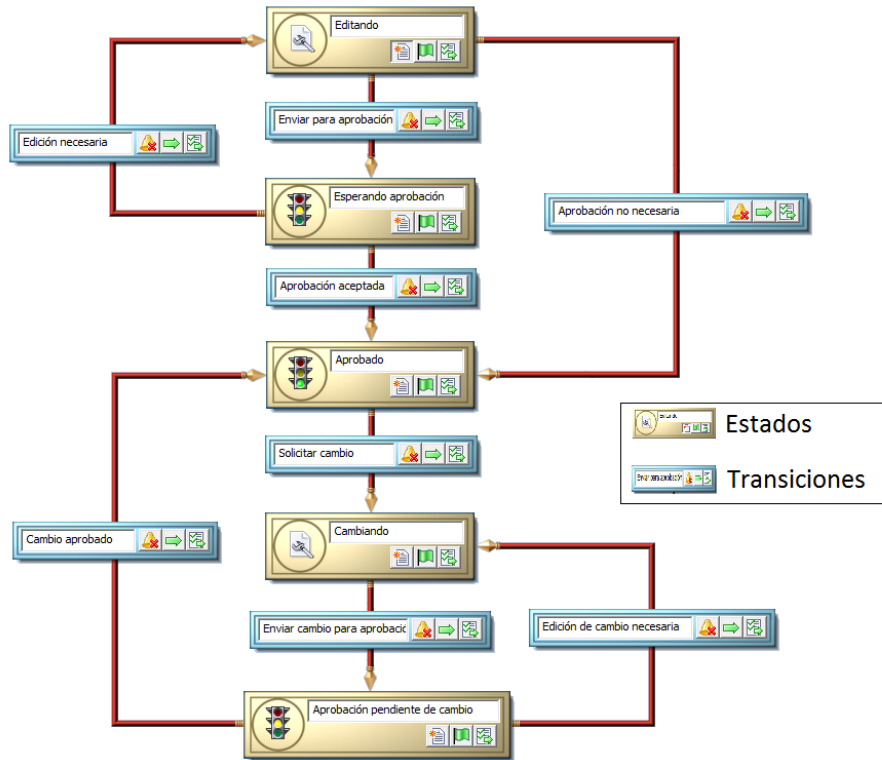
Para ello, se propone añadir filtros en los parámetros ERP de la integración, definidos por el usuario, a fin de desglosar el paquete de datos y construir varias llamadas que contengan únicamente el tratamiento de uno de los valores a leer y/o modificar. De este modo, en lugar de realizar una única a llamada a cada método, se realizará una llamada por cada parámetro de filtro que apunte a una propiedad duplicada en la construcción de la llamada.

Ámbito:

- Integración: Detección y extracción de parámetros duplicados
- iI_ERP (iRPS2012): Construcción de las llamadas múltiples

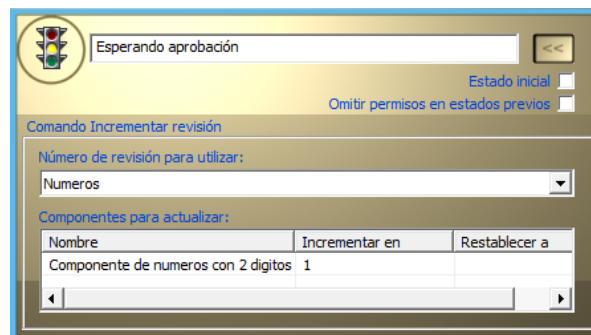
A1.7 Siguiente Revisión PDM

El sistema de gestión documental PDM, utilizado por SolidWorks para la gestión en empresa de los modelos 3D, cuenta con un sistema de asignación de versiones, estados y revisiones para los archivos que almacena, a fin de introducir estos archivos en un flujo de trabajo definido por la empresa.



37: Esquema de flujo de trabajo PDM

Por cada estado del flujo de trabajo puede definirse un número de revisión para asignar o incrementar las revisiones de los documentos que se hallen en dicho estado.



Estado de flujo de trabajo: Esperando aprobación

Estado inicial

Omitir permisos en estados previos

Comando Incrementar revisión

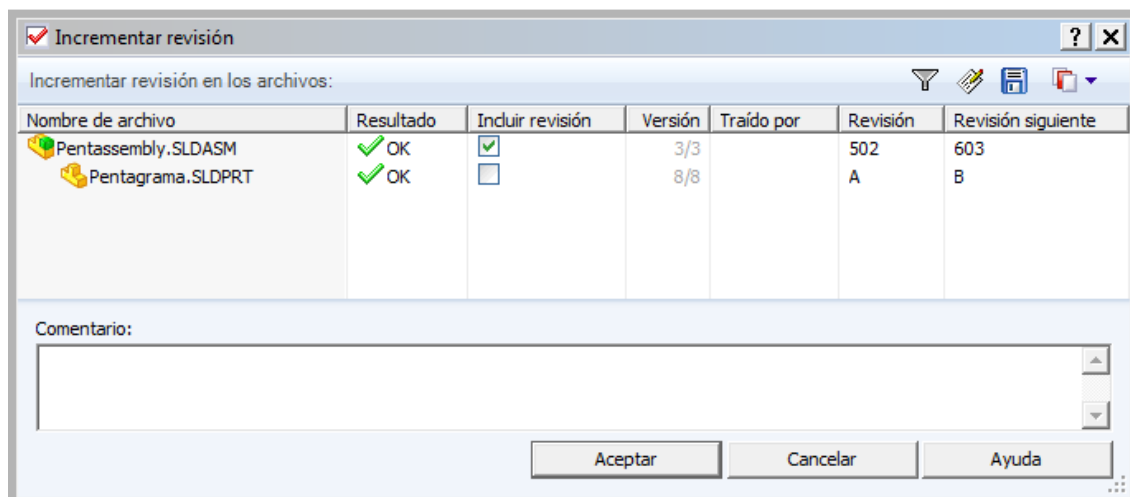
Número de revisión para utilizar:
Numeros

Componentes para actualizar:

Nombre	Incrementar en	Restablecer a
Componente de numeros con 2 digitos	1	

38: Estado de flujo de trabajo, con número de revisión asignado

Dado que la revisión, como elemento indicativo del estado del archivo en el flujo de trabajo, puede ser relevante en el sistema ERP, se considera una funcionalidad importante la de contar con el siguiente valor que la revisión de un archivo pueda tomar.



39: Dialogo de PDM para incrementar revisión manualmente

Entre las funcionalidades de la API de PDM no se cuenta la de obtener la siguiente revisión de un archivo, de modo que se plantea implementar una serie de funciones destinadas a calcular la siguiente revisión del archivo en base a su revisión actual.

Ámbito:

iI_PDM (iPDME)

A1.8 Codificación manual de artículos en RPS2012

RPS2012 cuenta con un sistema de series de numeración, particularizado para cada empresa, para la codificación de artículos. Cuando, para una empresa, está definido un sistema de codificación semejante, a los nuevos artículos les es asignado automáticamente un código distinto al que pueda haber facilitado el usuario que registraba el artículo.

Codificación Automática Artículo

Serie Codificación automática de artículo

Orden

Linea Producto

Familia Producto

Subfam. Producto

40: Serie de numeración en RPS2012

Para acomodar necesidades específicas de clientes, según las cuales se requiere registrar artículos en ERP con el código especificado en la integración, se plantea modificar la función de inserción de artículos del servicio conector RPS2012, InsertarArticle.

Se incluirá un parámetro adicional en la cabecera de la función: CodificacionAutomatica (booleano). Cuando el valor de CodificacionAutomatica es verdadero, el método tendrá el comportamiento actual, según el cual se utiliza la serie de numeración definida en el ERP para la codificación del artículo, independientemente del código suministrado. Cuando el valor de CodificacionAutomatica es falso, de forma previa al registro del artículo, la función anulará de forma temporal la serie de numeración del ERP para dicho artículo, de modo que el código con el que se registre será exactamente el definido en la integración.

Para prevenir colisiones por acceso simultaneo por parte de varios usuarios, toda la operación se realizará de forma transaccional.

Ámbito: Servicio RPS2012 (RPS2012_SW)

A1.9 Filtrar parámetros en formulario de artículo

Dado que el número de parámetros de relevancia en ERP_ART (artículos) que se pueden definir en la configuración de la integración es elevado, en ocasiones puede resultar interesante limitar o restringir el número de parámetros que se visualizan en el formulario de artículo, a fin de contar con una visualización más cómoda y menos congestionada.

La propuesta es como sigue: por cada parámetro a mostrar en el formulario de artículo (definidos en el apartado 'Presentación componente' de la configuración), se incluirán dos campos adicionales: ParFiltro y ValorVisible

ParFiltro representará el parámetro del artículo cuyo valor se evalúa para aplicar el filtro. Los posibles parámetros de filtro son aquellos definidos con relevancia en ERP_ART.

ValorVisible indica el valor que debe tomar ParFiltro para incluir o excluir el parámetro en el formulario. El formato para definir los valores debe ser el siguiente:

Para incluir en el formulario parámetros cuyo valor de ParFiltro toma un único valor: <valor>

Para incluir en el formulario parámetros cuyo valor de ParFiltro toma un valor de una lista de valores:
<valor 1> | <valor 2> | ... | <valor n>

Para excluir del formulario parámetros cuyo valor de ParFiltro toma un único valor: - <valor>

Para excluir del formulario parámetros cuyo valor de ParFiltro toma un valor de una lista de valores: -
<valor 1> | ... | <valor n>

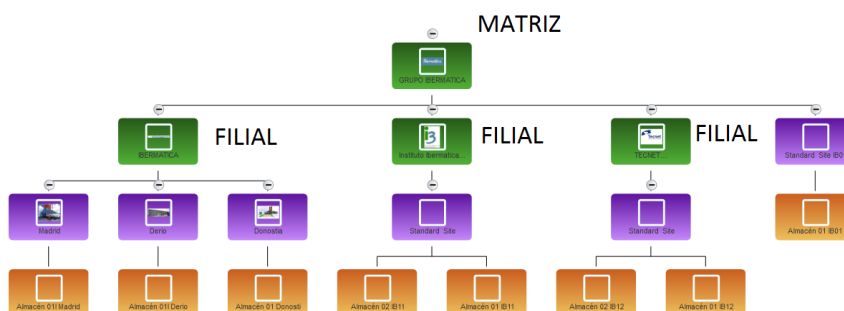
Los parámetros excluidos no se mostrarán en el formulario de artículo, y tampoco se registrarán en el ERP al dar de alta el artículo. Por extensión, tampoco se registrarán parámetros auxiliares (tales como los parámetros de filtro de la integración empleados en las llamadas múltiples*) referenciados por parámetros excluidos.

Ámbito:

- Información de artículo
- Lista de materiales

A1.10 Trabajo con grupos empresariales en RPS2012

La manera de trabajar de RPS se basa en la definición de entidades ‘Empresa’, bajo las cuales se catalogan el resto de datos (artículos, proyectos, parámetros, etc.). No obstante, en ocasiones puede trabajarse con ‘grupos empresariales’, los cuales se componen de una empresa matriz, y de uno o varios niveles inferiores de empresas ‘filiales’ ubicados más abajo en la jerarquía.



41: Esquema de grupo empresarial en RPS2012

Desde el punto de vista de los datos maestros que se manejan desde la integración, la mayor parte de ellos están vinculados a la empresa matriz. No obstante, hay tablas / entidades* que contienen datos específicos de la empresa de trabajo (filial o matriz).

Cuando se crea un grupo empresarial, los datos asociados a artículos, estructuras y proyectos se dividen en:

- Aquellos que son compartidos por todas las filiales (asociados internamente a la empresa matriz)
- Aquellos que son propios de cada empresa filial.

Se plantea que los datos de la empresa matriz sean visibles y modificables, según su nivel de permiso dentro del entorno multiempresa, para las empresas filiales.

Ámbito: Servicio RPS2012 (RPS2012_SW)

A1.11 Comprobación de licencia para utilizar el servicio conector RPS2012

Para evitar el acceso al servicio RPS2012 por parte de terceros no autorizados, se plantea el implementar un sistema de verificación de permisos en base a números de serie, de manera similar al chequeo de seguridad empleado en la integración.

El método empleado en la integración para autorizar el uso de la aplicación a los usuarios se basa en cotejar un determinado número de serie con claves encriptadas almacenadas en un archivo de texto. El proceso de encriptar y desencriptar los números de serie se realiza por medio de una serie de funciones en el módulo iPDME. Estas funciones pueden ser agregadas al servicio RPS2012 a modo de funciones auxiliares.

El número de serie se pasa cómo parámetro adicional, junto con el resto de credenciales de conexión, al método Conectar del servicio RPS2012, y se coteja con las claves encriptadas de un archivo de texto almacenado en el servidor donde está alojado el servicio.

Adicionalmente, se plantea el "autorizar" determinados servidores (permitir el acceso libre al servicio hospedado) por medio de la dirección física (MAC) de sus adaptadores de red, de forma análoga al chequeo empleado con los números de serie. De esta manera, aunque el número de serie suministrado no sea válido, podrá utilizarse el servicio siempre que la dirección MAC de alguno de sus adaptadores de red se encuentre encriptada en el archivo de claves.

Ámbito:

- Servicio RPS2012 (RPS2012_SW): Cotejar número de serie
- il_ERP (iRPS2012): Incluir número de serie en parámetros de conexión al servicio.

A1.12 Módulo iWIN

La interfaz il_PDM de la integración contempla las funciones y métodos dedicados a la gestión documental, y control de versiones de los archivos CAD. Los dos módulos existentes que implementan la interfaz son iPDME e iPDME, que utilizan funciones de las APIs de SolidWorks PDM Workgroup y SolidWorks Enterprise PDM respectivamente.

Tanto Workgroup como Enterprise son herramientas de gestión documental desarrolladas como parte de SolidWorks, por lo que son válidos únicamente si el programa CAD empleado es SolidWorks; las interfaces iPDME e iPDME son, por lo tanto, únicamente válidas en conjunción con la interfaz iSW.

Para sortear esta limitación, se propone desarrollar un nuevo módulo que implemente la interfaz il_PDM, empleando exclusivamente las capacidades de gestión documental de Windows.

Métodos de la interfaz il_PDM:

Propiedades

Property Usuario() *As String*

Resumen: Devuelve o establece el nombre de usuario

- iPDME: Devuelve o establece el nombre de usuario para la conexión con PDM
- iWIN: Devuelve o establece el nombre de usuario de Windows

Property Password() *As String*

Resumen: Devuelve o establece la contraseña

- iPDME: Devuelve o establece la contraseña para la conexión con PDM
- iWIN: [No implementar]

Property Servidor() *As String*

Resumen: Devuelve o establece el nombre del almacén de documentos

- iPDME: Devuelve o establece el nombre del almacén para la conexión PDM
- iWIN: Devuelve o establece la ruta local bajo la que se almacenan los documentos

Property ResponsablesDeProyectos() *As Collection*

Resumen: Almacena los nombres de usuario responsables de proyectos.

- iPDME:
- iWIN: [No implementar]

[ReadOnly Property](#) Propietario([ByVal Documento As String](#)) [As String](#)

Resumen: Devuelve el nombre de usuario en posesión del documento

- iPDME: Devuelve el nombre de usuario PDM en posesión del documento
- iWIN: se genera un archivo *Documento.owner* al hacer checkout con el usuario que se trae el documento

[ReadOnly Property](#) Proyecto([ByVal Documento As String](#)) [As String](#)

Resumen: Devuelve la carpeta bajo la que está ubicado el documento.

- iPDME: Devuelve la carpeta del almacén PDM bajo la que está ubicado el documento
- iWIN: Devuelve la ruta local de la carpeta bajo la que está ubicado el documento

[ReadOnly Property](#) Version([ByVal Documento As String](#)) [As String](#)

Resumen: Devuelve la versión actual del documento

- iPDME: Devuelve la versión actual del documento en PDM
- iWIN: Devuelve cadena vacía ("")

[ReadOnly Property](#) Revision([ByVal Documento As String](#)) [As String](#)

Resumen: Devuelve la revisión actual del documento

- iPDME: Devuelve la revisión actual en PDM del documento.
- iWIN: devolver ""

[ReadOnly Property](#) ArchivoAsociado([ByVal Documento3D As String](#),
[ByVal Extension As String](#)) [As String](#)

Resumen: Devuelve la ruta local de un archivo asociado al documento, según extensión.

- iPDME: Devuelve la ruta local de un archivo asociado al documento, según extensión. Si no se indica extensión, se utiliza por defecto la de un plano de SolidWorks (.slddrw)
- iWIN: Devuelve la ruta local de un archivo con el mismo nombre que el documento, según extensión. Si no se indica extensión, se utiliza por defecto la de un plano de Solid Edge (.dft)

[ReadOnly Property](#) PermiteModifVariables() [As Boolean](#)

Resumen: Informa sobre si se permite la modificación de variables en el almacén.

- iPDME: Informa sobre si se permite la modificación de variables en el almacén.
- iWIN: Devolver TRUE

`ReadOnly Property` `SiguienteRevision(ByVal Documento As String) As String`

Resumen: Calcula la siguiente revisión que le corresponde a un documento en su estado actual

- iPDME: Calcula la siguiente revisión que le corresponde a un documento en PDM, según su revisión local y su estado actual en el flujo de trabajo.
- iWIN: **[No implementar]**.

Funciones:

`Sub` `Conectar(ByVal Usuario As String, _`
`ByVal Password As String, _`
`ByVal Servidor As String)`

Resumen: Establece una conexión con el servidor donde se alojan los documentos

- iPDME: Establece una conexión con el almacén PDM indicado, con las credenciales indicadas.
- iWIN: Verifica que la ruta indicada en *Servidor* existe en el equipo, y que el usuario tiene acceso a dicha ruta. Si el resultado es falso, no se permite acceder al resto de métodos.

`Sub` `Desconectar()`

Resumen: Cierra la conexión con el servidor

- iPDME: Cierra la conexión con el almacén PDM
- iWIN: **[No implementar]**

`Function` `ResponsablesProyecto(ByVal Documento As String) As Collection`

Resumen:

- iPDME:
- iWIN: **[No implementar]**

`Function` `AbreDoc(ByVal Documento As String, _`
`ByRef RutaDeGuardado As String, _`
`Optional ByVal Proyecto As String = "") As String`

Resumen: Busca un documento en el servidor, y devuelve su ruta local de guardado.

- iPDME: Busca un documento en el almacén PDM y devuelve su ruta local de guardado. Puede indicarse bajo que carpeta de PDM se encuentra el archivo, para el caso de que haya varios con el mismo nombre.
- iWIN: Busca un documento en el 'servidor' y devuelve su ruta de guardado.

```
Function AbreDocs(ByVal Documento As String, _  
    ByVal Revision As String, _  
    ByRef RutaDeGuardado As String, _  
    ByVal SoloPlanos As Boolean, _  
    Optional ByVal IncluirComponentes As Boolean = False) As Object
```

Resumen:

- iPDME: Lista los nombres de los archivos asociados al documento pasado como parámetro en PDM.
- iWIN: Ignorar revisión y listar archivos con el mismo nombre que el pasado como parámetro

```
Function CheckOut(ByVal Documento As String, _  
    ByRef EstabaTraido As Boolean) As String
```

Resumen: Busca el documento indicado en el almacén, y lo 'trae' para el usuario actual, bloqueando el acceso de edición a otros usuarios. Si el usuario ya tenía el documento bajo su posesión, se informa mediante la variable *EstabaTraido*. Devuelve la ruta local de guardado del documento.

- iPDME: Busca el documento indicado en el almacén, y lo 'trae' para el usuario actual. Otros usuarios pueden visualizar el documento y editarlo, pero los cambios que puedan realizar no quedarán registrados en PDM. Si el usuario ya tenía el documento bajo su posesión, se informa mediante la variable *EstabaTraido*.
- iWIN: Busca el documento indicado en el almacén y elimina la marca de solo-lectura. Si el documento estaba habilitado para edición, se informa en la variable *EstabaTraido*.

```
Function CheckOut(ByVal Documento As String, _  
    ByRef RutaDeGuardado As String, _  
    ByRef EstabaTraido As Boolean) As String
```

Resumen: Sobrecarga del método *CheckOut* anterior.

- iPDME: Sobrecarga del método *CheckOut* anterior.
- iWIN: Sobrecarga del método *CheckOut* anterior.

```
Sub CheckIn(ByVal Documento As String)
```

Resumen: Valida el documento bajo la posesión del usuario, registrando los cambios realizados en el almacén.

- iPDME: Registra el documento, creando una nueva versión y permitiendo a otros usuarios 'traerlo' y modificarlo.
- iWIN: Registra el documento como de solo-lectura en el almacén.

Sub CheckIn(**ByVal** Documento **As String**, **ByVal** Comentario **As String**)

Resumen: Sobrecarga del método anterior, apostillando un breve comentario acerca de la nueva versión del documento registrada.

- iPDME: Sobrecarga del método anterior, apostillando un breve comentario acerca de la nueva versión del documento registrada.
- iWIN: Sobrecarga del método anterior, sin comentario.

Sub SoltarPosesion(**ByVal** Archivo **As String**)

Resumen: Anula la posesión del usuario actual sobre el documento, descartando los cambios que este haya podido realizar.

- iPDME: Anula la posesión del usuario actual sobre el documento, descartando los cambios que este haya podido realizar
- iWIN: Registra el documento como de solo-lectura en el almacén.

Function ExisteCodigo(**ByVal** Codigo **As String**) **As Boolean**

Resumen: Busca un documento de nombre *Codigo* en el almacén. Devuelve TRUE si existe, y FALSE en caso contrario.

- iPDME: Busca un documento de nombre *Codigo* en el almacén. Devuelve TRUE si existe, y FALSE en caso contrario.
- iWIN: Busca un documento de nombre *Codigo* en el almacén. Devuelve TRUE si existe, y FALSE en caso contrario.

Function CrearProyectoPDM(**ByVal** NombreProy **As String**, _
ByVal DescProy **As String**, _
ByVal ProyPadre **As String**) **As Boolean**

Resumen: Crea un proyecto en PDM, generando toda la estructura de carpetas bajo la carpeta indicada.

- iPDME: Crea un proyecto en PDM, generando toda la estructura de carpetas bajo la carpeta indicada.
- iWIN: Crea un proyecto en PDM, generando toda la estructura de carpetas bajo la carpeta *ProyPadre*


```
Function RenombrarDocumento(ByVal Documento As String, _  
    ByVal NuevoNombre As String) As Boolean
```

Resumen: Renombra un documento

- iPDME: Modifica el nombre de un documento de forma local y en el almacén PDM
- iWIN: Modifica el nombre de un documento de forma local y en el almacén PDM

```
Function DarValorAVariable(ByVal Documento As String, _  
    ByVal ConfiguracionRelevante As String, _  
    ByVal VariablePDM As String, _  
    ByVal Valor As String) As Boolean
```

Resumen: Modifica el valor de una variable en PDM, o la añade si no existe.

- iPDME: : Modifica el valor de una variable en PDM, o la añade si no existe
- iWIN: [No implementar]

Ámbito:

il_PDM: Nuevo módulo para implementar la interfaz (iWIN)

A1.13 Módulo iSE

Bajo demanda de cliente, se propone la implementación de un módulo que permita a la integración trabajar con el programa CAD Solid Edge, de Siemens. Deberá estudiarse cuales de los métodos de la API de Solid Edge son apropiados para las funcionalidades de la integración.

Métodos de la interfaz il_CAD:

Propiedades

`ReadOnly Property` Conectado() `As Boolean`

Resumen: Informa sobre si el módulo está conectado a una instancia en ejecución del programa CAD.

- iSW: : Informa sobre si el módulo está conectado a una instancia en ejecución de SolidWorks
- iSE: : Informa sobre si el módulo está conectado a una instancia en ejecución de SolidEdge

`ReadOnly Property` DocumentoActivo() `As String`

Resumen: Devuelve el nombre del documento activo

- iSW: Devuelve el nombre del documento activo
- iSE: Devuelve el nombre del documento activo

`ReadOnly Property` RutaFichero(`Optional ByVal` ComponenteSW `As Object = Nothing`) `As String`

Resumen: Devuelve la ruta de fichero del documento activo

- iSW: Devuelve la ruta de fichero del documento activo
- iSE:

`ReadOnly Property` NombreFichero(`Optional ByVal` ComponenteSW `As Object = Nothing`) `As String`

Resumen: Devuelve el nombre de fichero del componente CAD pasado por parámetro (por omisión del documento activo)

- iSW: Devuelve el nombre de fichero del componente SW pasado por parámetro (por omisión del documento activo)
- iSE:

Property Configuracion() As String

Resumen: Devuelve o establece la configuración del documento actual en el programa CAD

- iSW: Devuelve o establece la configuración del documento actual en el programa CAD
- iSE: [No Implementar]

PropertyCodigoDocActivo() As String

Resumen: Devuelve o establece el código del documento activo

- iSW: Devuelve o establece el código del documento activo
- iSE: Devuelve o establece el código del documento activo

PropertyCodigoDoc(Optional ByVal ComponenteSW As Object = Nothing) As String

Resumen: Devuelve o establece el código del componente pasado como parámetro. Si se omite, devuelve o establece el código del documento activo en el programa CAD.

- iSW: Devuelve o establece el código del componente pasado como parámetro. Si se omite, devuelve o establece el código del documento activo en SolidWorks
- iSE: Devuelve o establece el código del componente pasado como parámetro. Si se omite, devuelve o establece el código del documento activo en SolidEdge

PropertyDescripcion(ByVal Configuracion As String) As String

Resumen: Devuelve o establece la descripción del documento, según la configuración indicada.

- iSW: Devuelve o establece la descripción del documento, según la configuración indicada
- iSE:

ReadOnly Property ExtensionDoc(Optional ByVal ComponenteSW As Object = Nothing) As String

Resumen: Devuelve la extensión del componente pasado como parámetro. Si se omite, se devuelve la extensión del documento activo.

- iSW: Devuelve la extensión del componente pasado como parámetro. Si se omite, se devuelve la extensión del documento activo
- iSE:

ReadOnly Property Proyecto(ByVal Documento As String) As String

Resumen: Devuelve la carpeta contenedora del documento indicado.

- iSW: Devuelve la carpeta contenedora del documento indicado
- iSE:

ReadOnly Property Propietario(ByVal Documento As String) As String

Resumen: Devuelve el propietario del documento.

- iSW: Devuelve el nombre de usuario propietario del documento en PDM
- iSE:

Property FechaActual() As String

Resumen: Devuelve la fecha actual.

- iSW: Devuelve la fecha actual.
- iSE: Devuelve la fecha actual

Property FechaYHoraActual() As String

Resumen: Devuelve la fecha y hora actuales.

- iSW: Devuelve la fecha y hora actuales.
- iSE: Devuelve la fecha y hora actuales

ReadOnly Property Usuario() As String

Resumen: Devuelve el nombre de usuario actualmente conectado.

- iSW: Devuelve el nombre de usuario PDM actualmente conectado.
- iSE: Devuelve el nombre de usuario actualmente conectado en Windows

ReadOnly Property Version(Optional ByVal Documento As String = "") As String

Resumen: Devuelve la versión actual del documento indicado; si este se omite, se devuelve la versión actual del documento activo.

- iSW: Devuelve la versión actual del documento en PDM
- iSE:

`ReadOnly Property` Plano(`Optional ByVal` ComponenteSW `As Object = Nothing`) `As String`

Resumen: Devuelve el nombre del componente sin la extensión.

- iSW: Devuelve el nombre del componente SolidWorks sin la extensión
- iSE: Devuelve el nombre del componente SolidEdge sin la extensión

`Property` Propiedad(`ByVal` Configuracion `As String`, `_`
`ByVal` NombrePropiedad `As String`, `_`
`ByVal` TipoDato `As String`, `_`
`Optional ByVal` ModeloSW `As Object = Nothing`) `As String`

Resumen: Devuelve o establece el valor de la propiedad indicada, formateado según el tipo de dato de la propiedad. Si no se indica ningún documento, se trabaja con las propiedades del documento activo por omisión.

- iSW: Devuelve o establece el valor de la propiedad indicada, formateado según el tipo de dato de la propiedad. Si no se indica ningún documento, se trabaja con las propiedades del documento activo por omisión.
- iSE: Devuelve o establece el valor de la propiedad indicada, formateado según el tipo de dato de la propiedad. Si no se indica ningún documento, se trabaja con las propiedades del documento activo por omisión.

`Property` DatosDeConfiguracion() `As Object` 'Declaraciones.ClasesBase.DatosConf

Resumen: Devuelve o establece una instancia del objeto DatosConf

- iSW: Devuelve o establece una instancia del objeto DatosConf
- iSE: Devuelve o establece una instancia del objeto DatosConf

`ReadOnly Property` Se_Puede_Renombrar(`Optional ByVal` configuracion `As Object = Nothing`)
`As Boolean`

Resumen: Devuelve TRUE si se permite modificar el nombre del documento, y FALSE en caso contrario.

- iSW: Devuelve TRUE si se permite modificar el nombre del documento SolidWorks, y FALSE en caso contrario.
- iSE: Devuelve TRUE si se permite modificar el nombre del documento SolidEdge, y FALSE en caso contrario

`ReadOnly Property ConfiguracionActiva() As String`

Resumen: Devuelve el nombre de la configuración activa en el programa CAD

- iSW: Devuelve el nombre de la configuración activa en SolidWorks
- iSE: Devuelve "Custom" (la única configuración editable por el usuario en SolidEdge)

`ReadOnly Property ContornoXYZ(Optional ByVal ComponenteSW As Object = Nothing) As String`

Resumen: Devuelve la longitud del contorno X . Y . Z del componente indicado (por omisión, del documento activo)

- iSW:
- iSE: [No Implementar]

`ReadOnly Property ContornoX(Optional ByVal ComponenteSW As Object = Nothing) As String`

Resumen: Devuelve la longitud del contorno X del componente indicado (por omisión, del documento activo)

- iSW:
- iSE: [No Implementar]

`ReadOnly Property ContornoY(Optional ByVal ComponenteSW As Object = Nothing) As String`

Resumen: Devuelve la longitud del contorno Y del componente indicado (por omisión, del documento activo)

- iSW:
- iSE: [No Implementar]

`ReadOnly Property ContornoZ(Optional ByVal ComponenteSW As Object = Nothing) As String`

Resumen: Devuelve la longitud del contorno Z del componente indicado (por omisión, del documento activo)

- iSW:
- iSE: [No Implementar]

ReadOnly Property MPEnPropiedades() **As Boolean**

Resumen: Devuelve TRUE si la integración está configurada para extraer los datos de Materia Prima desde las propiedades del documento, y FALSE en caso contrario.

- iSW: Devuelve TRUE si la integración está configurada para extraer los datos de Materia Prima desde las propiedades del documento SolidWorks, y FALSE en caso contrario
- iSE: Devuelve TRUE

ReadOnly Property MPPropCodigo() **As String**

Resumen: Devuelve el nombre de la propiedad configurada para almacenar el código de la Materia Prima.

- iSW: Devuelve el nombre de la propiedad configurada para almacenar el código de la Materia Prima.
- iSE: Devuelve el nombre de la propiedad configurada para almacenar el código de la Materia Prima.

ReadOnly Property MPPropCantidad() **As String**

Resumen: Devuelve el nombre de la propiedad configurada para almacenar la cantidad de Materia Prima

- iSW: Devuelve el nombre de la propiedad configurada para almacenar la cantidad de Materia Prima.
- iSE: Devuelve el nombre de la propiedad configurada para almacenar la cantidad de Materia Prima.

ReadOnly Property MPPropUnidades() **As String**

Resumen: Devuelve el nombre de la propiedad configurada para almacenar las unidades de la Materia Prima

- iSW: Devuelve el nombre de la propiedad configurada para almacenar las unidades de la Materia Prima
- iSE: Devuelve el nombre de la propiedad configurada para almacenar las unidades de la Materia Prima

ReadOnly Property MPPPropDescripcion() As String

Resumen: Devuelve el nombre de la propiedad configurada para almacenar la descripción de la Materia Prima

- iSW: Devuelve el nombre de la propiedad configurada para almacenar la descripción de la Materia Prima
- iSE: Devuelve el nombre de la propiedad configurada para almacenar la descripción de la Materia Prima

ReadOnly Property EsVirtual(Optional ByVal ComponenteSW As Object = Nothing) As Boolean

Resumen: Devuelve TRUE si se identifica el componente indicado como virtual en el programa CAD (por omisión, el documento activo), y FALSE en caso contrario.

- iSW: Devuelve TRUE si se identifica el componente indicado como virtual en SolidWorks (por omisión, el documento activo), y FALSE en caso contrario
- iSE: Devuelve TRUE si se identifica el componente indicado como virtual en SolidEdge (por omisión, el documento activo), y FALSE en caso contrario

ReadOnly Property ConfiguracionRelevante(Optional ByVal DocSW As Object = Nothing) As String

Resumen: Devuelve la configuración del programa CAD empleada a la hora de trabajar con el documento indicado (por omisión con el documento activo)

- iSW: Devuelve la configuración de SolidWorks empleada a la hora de trabajar con el documento indicado (por omisión con el documento activo)
- iSE: Devuelve "Custom" (la única configuración editable por el usuario en SolidEdge)

ReadOnly Property NombrePDM(Optional ByVal ComponenteSW As Object = Nothing) As String

Resumen: Devuelve el nombre con el que se identifica el componente indicado en PDM (por omisión, el documento activo)

- iSW: Devuelve el nombre con el que se identifica el componente indicado en PDM (por omisión, el documento activo)
- iSE: Devolver nombre de archivo con la extensión configurada (si no está ,devolver el nombre con la extension original)

ReadOnly Property SoloLectura(Optional ByVal Componente As Object = Nothing) As Boolean

Resumen: Devuelve TRUE si el componente indicado está en estado de sólo-lectura (por omisión, el documento activo)

- iSW: Devuelve TRUE si el componente indicado está en estado de sólo-lectura (por omisión, el documento activo)
- iSE:

ReadOnly Property formConfiguracion() **As** Form

Resumen: Devuelve un puntero al objeto formulario de configuración CAD.

- iSW: Devuelve un puntero al objeto formulario de configuración iSW
- iSE: Devuelve un puntero al objeto formulario de configuración iSE

Property xmlConfiguracion() **As** Xml.XmlNode

Resumen: Devuelve o establece el puntero al objeto XML de configuración CAD.

- iSW: Devuelve o establece el puntero al objeto XML de configuración CAD
- iSE: Devuelve o establece el puntero al objeto XML de configuración CAD

Property ExcluirDeLM(Optional ByVal ComponenteSW **As** Object = Nothing) **As** String

Resumen: Informa o establece, según el formato Si / No de la integración, si el componente indicado está excluido o no de las listas de materiales.

- iSW:
- iSE: [No Implementado ¿?]

ReadOnly Property Kit(Optional ByVal ComponenteSW **As** Object = Nothing) **As** String

Resumen: Informa o establece, según el formato Si / No de la integración, si el componente indicado es un Kit.

- iSW:
- iSE: [Implementado ¿?]

ReadOnly Property Ficticio(Optional ByVal ComponenteSW **As** Object = Nothing) **As** String

Resumen: Informa o establece, según el formato Si / No de la integración, si el componente indicado si el componente indicado es ficticio.

- iSW: Informa o establece, según el formato Si / No de la integración, si el componente indicado si el componente indicado es ficticio
- iSE: Informa o establece, según el formato Si / No de la integración, si el componente indicado si el componente indicado es ficticio

Property TipoComponente(Optional ByVal ComponenteSW As Object = Nothing) As String

Resumen: Devuelve el tipo de componente del componente indicado (kit, estándar o ficticio; por omisión, tipo del documento activo)

- iSW: Devuelve el tipo de componente del componente indicado (kit, estándar o ficticio; por omisión, tipo del documento activo)
- iSE: Devuelve el tipo de componente del componente indicado (kit, estándar o ficticio; por omisión, tipo del documento activo)

ReadOnly Property CantidadEnLM(Optional ByVal ComponenteSW As Object = Nothing) As String

Resumen: Devuelve la cantidad con la que el componente interviene en la lista de materiales (por omisión, cantidad del documento activo)

- iSW: Devuelve la cantidad con la que el componente interviene en la lista de materiales (por omisión, cantidad del documento activo)
- iSE: Devuelve la cantidad con la que el componente interviene en la lista de materiales (por omisión, cantidad del documento activo). Si no existe el campo 'CantidadLM' en SE, devolver 1

ReadOnly Property Imagen(Optional ByVal ComponenteSW As Object = Nothing) As Object

Resumen: Devuelve una imagen en miniatura del componente indicado (por omisión, del documento activo)

- iSW: Devuelve una imagen en miniatura del componente indicado (por omisión, del documento activo)
- iSE: Devuelve una imagen en miniatura del documento activo.

ReadOnly Property ArchivoAsociadoPDM(ByVal ComponenteSW As Object, _
ByVal Extension As String) As String

Resumen: Devuelve la ruta local de archivo asociado al componente indicado, según extensión (por omisión, del documento activo).

- iSW: Devuelve la ruta local de archivo asociado al componente indicado, según extensión (por omisión, del documento activo). Puente con la propiedad ArchivoAsociado de il_PDM.
- iSE: [Enlazar a il_PDM](#)

ReadOnly Property `SiguienteRevisionPDM(ByVal ComponenteSW As Object) As String`

Resumen: Devuelve la siguiente revisión en PDM del componente indicado, en su estado actual (por omisión, del documento activo)

- iSW: Devuelve la siguiente revisión en PDM del componente indicado, en su estado actual (por omisión, del documento activo). Puente con la propiedad `SiguienteRevision` de `il_PDM`.
- iSE: [Enlazar a il_PDM](#)

ReadOnly Property `RevisionPDM(ByVal ComponenteSW As Object) As String`

Resumen: Devuelve la revisión actual en PDM del componente indicado (por omisión, del documento activo)

- iSW: Devuelve la revisión actual en PDM del componente indicado (por omisión, del documento activo)
- iSE: [Enlazar a il_PDM](#)

Property `AdapterPDM() As String`

Resumen: Devuelve el nombre del adapter PDM actual (el módulo de tipo `il_PDM` en uso)

- iSW: Devuelve el nombre del adapter PDM actual (el módulo de tipo `il_PDM` en uso)
- iSE: Devuelve el nombre del adapter PDM actual (el módulo de tipo `il_PDM` en uso)

Funciones

Function `Documento(Optional ByVal NombreDoc As String = "", _
Optional ByVal CodigoDocumento As String = "") As Object`

Resumen: Devuelve un puntero al documento CAD indicado, identificado por nombre de documento o por código (por omisión se devuelve un puntero al documento activo)

- iSW: Devuelve un puntero al documento CAD indicado, identificado por nombre de documento o por código (por omisión se devuelve un puntero al documento activo)
- iSE: Devuelve un puntero al documento CAD indicado, identificado por nombre de documento o por código (por omisión se devuelve un puntero al documento activo)

Function `Arbol(ByRef ElArbol As System.Data.DataTable) As String`

Resumen: Extrae en una tabla de datos la estructura de componentes del ensamblaje activo en el programa CAD, y devuelve el código de artículo que corresponde a dicho ensamblaje.

- iSW: Extrae en una tabla de datos la estructura de componentes del ensamblaje activo en el programa CAD, y devuelve el código de artículo que corresponde a dicho ensamblaje
- iSE: Extrae en una tabla de datos la estructura de componentes del ensamblaje activo en el programa CAD, y devuelve el código de artículo que corresponde a dicho ensamblaje

```
Function InfoArticulo(ByRef Datos As Collection, _  
    Optional ByRef TipoComponente As Integer = 0, _  
    Optional ByVal DocSW As Object = Nothing, _  
    Optional ByVal ConfActiva As String = "") As Boolean
```

Resumen: Extrae en una colección los valores de los parámetros de la integración con relevancia en CAD y que corresponden al documento CAD indicado, según la configuración indicada, e indica el tipo de componente del documento. Devuelve TRUE si el documento existe y la operación se ha realizado con éxito, y FALSE en caso contrario.

- iSW: Extrae en una colección los valores de los parámetros de la integración con relevancia en CAD y que corresponden al documento CAD indicado, según la configuración indicada, e indica el tipo de componente del documento. Devuelve TRUE si el documento existe y la operación se ha realizado con éxito, y FALSE en caso contrario.
- iSE: Extrae en una colección los valores de los parámetros de la integración con relevancia en CAD y que corresponden al documento CAD indicado, según la configuración indicada, e indica el tipo de componente del documento. Devuelve TRUE si el documento existe y la operación se ha realizado con éxito, y FALSE en caso contrario.

```
Function ItemSeleccionado(ByRef DatosCAD As Collection, _  
    ByRef DatosOper As Collection, _  
    ByRef Componente As Object, _  
    ByRef TipoComponente As Integer, _  
    ByRef HijoSeleccionado As Boolean) As String
```

Resumen: Extrae en una colección los valores de los parámetros de la integración con relevancia en CAD, y que corresponden a elemento actualmente seleccionado en el programa CAD, junto con su tipo de componente. Si el documento activo es un ensamblaje y se ha seleccionado alguno de sus componentes, se devuelve TRUE en la variable de retorno *HijoSeleccionado* (FALSE en caso contrario), además de un puntero al elemento en la variable de retorno *Componente* (NOTHING en caso contrario).

Si el documento activo no es un ensamblaje, no tiene componentes o no se ha seleccionado ningún componente en el programa CAD, se realiza la operación sobre el documento activo.

- iSW:
- iSE:

```
Function GuardaDato(ByVal Dato As DefParametroAplicacion, _  
    ByVal Configuracion As String, _  
    Optional ByVal Componente As Object = Nothing) As Boolean
```

Resumen: Guarda el valor de un parámetro de la integración con relevancia CAD según su método de escritura establecido y la configuración indicada, sobre el componente indicado (por omisión, sobre el documento activo)

- iSW: Guarda el valor de un parámetro de la integración con relevancia CAD según su método de escritura establecido y la configuración indicada, sobre el componente indicado (por omisión, sobre el documento activo)
- iSE: Guarda el valor de un parámetro de la integración con relevancia CAD según su método de escritura establecido y la configuración indicada, sobre el componente indicado (por omisión, sobre el documento activo)

```
Function ObtenDato(ByVal Dato As DefParametroAplicacion, _  
    ByVal Configuracion As String, _  
    Optional ByVal Modelo As Object = Nothing) As Boolean
```

Resumen: Extrae el valor de un parámetro de la integración con relevancia CAD según su método de lectura establecido y la configuración indicada, del componente indicado (por omisión, del documento activo)

- iSW: Extrae el valor de un parámetro de la integración con relevancia CAD según su método de lectura establecido y la configuración indicada, del componente indicado (por omisión, del documento activo)
- iSE: Extrae el valor de un parámetro de la integración con relevancia CAD según su método de lectura establecido y la configuración indicada, del componente indicado (por omisión, del documento activo)

```
Function Salvar(Optional ByVal ComponenteSW As Object = Nothing) As Boolean
```

Resumen: Realiza una acción de guardado en el programa CAD sobre el componente indicado (por omisión, sobre el documento activo)

- iSW:
- iSE:

Function CotasDeInspeccion(**ByVal** Componente **As Object**) **As DataTable**

Resumen: Devuelve en una tabla de datos los nombres, valores y tolerancias (valores máximos y mínimos) de las cotas de inspección encontradas en el componente indicado; por omisión se buscan las cotas del documento activo.

- iSW:
- iSE:

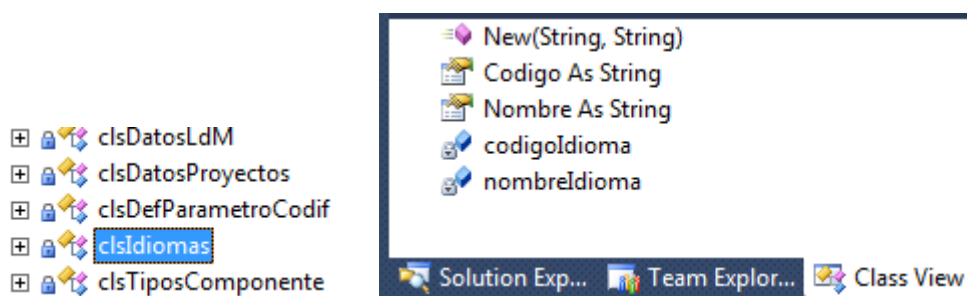
A2. Resumen de implementación de funcionalidades adicionales

A2.1 Multilinguaje

La manera de trabajar con recursos de varios idiomas en .NET es mediante la generación de ensamblados* (DLL) satélite. Estos ensamblados son adicionales al ensamblado principal de la aplicación, y se cargan en función de la cultura* en uso de la aplicación. Los ensamblados satélite se emplean habitualmente para cargar recursos particulares de una cultura específica que sean distintos a la cultura neutral de la aplicación, como pueden ser formatos de fecha/hora, separadores de decimales, y textos en otros idiomas. Para generar estos ensamblados, .NET emplea archivos de recursos en formato XML, específicos de cada cultura, donde se alberga contenido textual particular o 'localizado'. Los ensamblados se cargan en función de la cultura definida para el entorno de la aplicación, la cual, por defecto es heredada de la cultura definida para el equipo.

La gestión de los idiomas se ha implementado según lo siguiente:

Se ha agregado a las clases base de la integración la clase idiomas.



- El campo *Codigo* indica el código del idioma según el estándar de dos caracteres ISO (es: Español, en: Ingles, etc.).
- El campo *Nombre* contiene el nombre del idioma según se escribe en el propio idioma en cuestión.

Los objetos instanciados a partir de esta clase se utilizan para generar una colección de idiomas, agregando el paso de lectura y escritura de los parámetros correspondientes en el XML de configuración.

```
<IDIOMA ACTUAL="en" />
```

```
<IDIOMAS>
```

```
<IDIOMA NOMBRE="Español" CODIGO="es" />
```

```
<IDIOMA NOMBRE="English" CODIGO="en" />
```

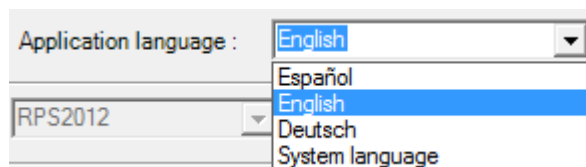
```
<IDIOMA NOMBRE="Deustch" CODIGO="de" />
```

```
<IDIOMA NOMBRE="System Language" CODIGO="" />
```

```
</IDIOMAS>
```

- El campo ACTUAL del elemento IDIOMA indica el código ISO del idioma a emplear en la ejecución de la aplicación.
- Los elementos IDIOMA bajo el subcampo IDIOMAS representan los idiomas para los que actualmente se cuenta con recursos, y que por tanto pueden emplearse en la aplicación. El elemento con el campo CODIGO vacío indica que el idioma a emplear es el utilizado por el sistema operativo. Si la aplicación no cuenta con recursos para el idioma en uso en el equipo, se utilizará por defecto el inglés (en).

El idioma a utilizar puede ser modificado desde el formulario de configuración general.



El control permite al usuario seleccionar el idioma en uso de la aplicación, el cual es leído por la aplicación a través de la propiedad *Idioma* del objeto de configuración. Los valores del *ComboBox* se cargan mediante una colección de objetos de la clase *clsIdioma*, cuyos valores se leen desde el XML.

El contenido textual de controles de formularios .NET, tales como botones, etiquetas y menús contextuales puede asignarse de forma dinámica mediante la clase *ComponentResourceManager* de .NET. El objeto instanciado a partir de esta clase cuenta con métodos para acceder a recursos multilingüaje.

`Dim resources As New`

`System.ComponentModel.ComponentResourceManager(GetType(Lista_de_materiales2))`

El contenido textual localizado se almacena en los archivos de recursos .resx empleados por Visual Studio para la generación de los ensamblados satélite.

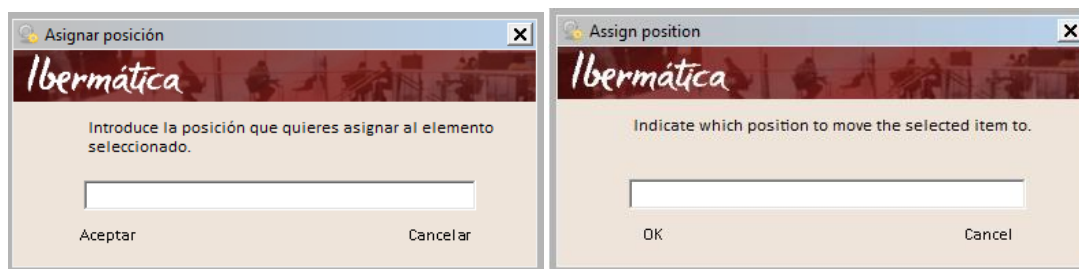
Name	Value	Comment
\$this.Text	Upload Bill of Materials	Formulario
btnCancelar.Text	Cancel	
btnCodificarTodos.Text	Encode all items	
btnModoEdicion.Text	Enable edition mode	
btnTraerTodasLasRutas.Text	Import all routes	
CambiarUsuarioPDMToolStripMenuItem.Text	Change PDM user	
Cantidad.HeaderText	Quantity	Columna de cantidad en arbol TR
Codigo.HeaderText	Code	Columna de código del arbol TR
imbAsignarOp	Bill of materials: assign operation	Cabecera de iMsgBox para asignar operacion
imbEdicionDatos	Data edition.	Cabecera de iMsgBox para edicion de datos
imbIntroducePos	Assign position	Cabecera de iMsgBox para asignar posicion

En estos archivos, la información de la columna de valor (*Value*) se utiliza como el valor del recurso localizado, identificado por la clave en la columna *Name*. La columna de comentario no se lee durante

el proceso de compilación del ensamblado satélite y se utiliza únicamente de modo orientativo para clasificar los recursos. La clave permite acceder al valor del recurso a través del método *GetString("clave")* del objeto gestor de recursos.

```

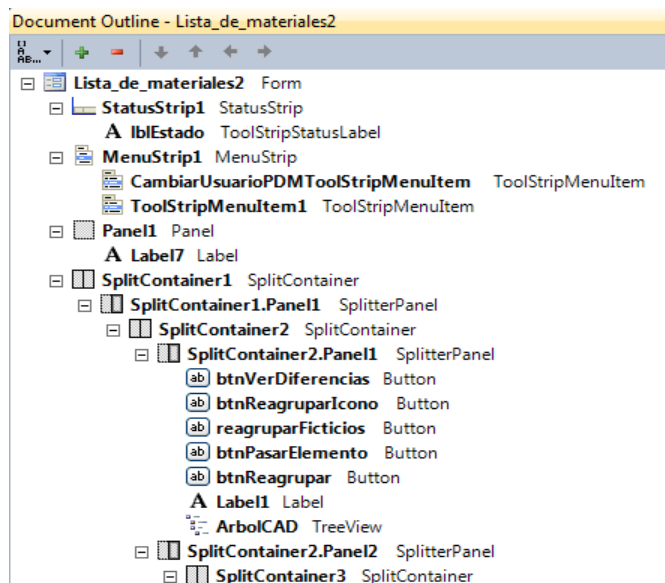
Dim imbIntroducPosic As String = resources.GetString("imbIntroducPosic")
Dim sAceptar As String = resources.GetString("sAceptar")
Dim sCancelar As String = resources.GetString("sCancelar")
  
```



Si en el valor de la clave se proporciona la ruta de acceso a la propiedad textual de un control (formada por el nombre del control, seguido de un punto y el nombre de la propiedad), el gestor de recursos podrá asignar el valor textual que corresponda a la propiedad de forma dinámica mediante el método *ApplyResources(nombreControl, "nombreControl.Propiedad")*



Inicialmente, la metodología para aplicar los recursos era de forma individual, con una llamada al método correspondiente (*GetString* en el caso de cadenas de caracteres, *ApplyResources* en el caso de controles). Tras un breve análisis se decidió implementar un método recursivo para la asignación dinámica de recursos, aprovechando la naturaleza jerárquica de la organización de los controles de formulario .NET, donde los controles están a su vez contenidos en otros controles.



La propiedad *HasChildren* de la clase *System.Windows.Forms.Control* indica si el control en cuestión es un contenedor de otros controles, los cuales, en el caso de que *HasChildren* sea verdadero, pueden ser accedidos mediante la propiedad *Controls*, que retorna una colección punteros a cada uno de los controles hijos.

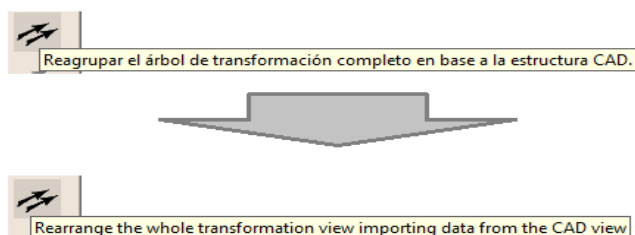
```

Private Sub AplicarRecursosAControles(ByRef ctrl As Control)
  For Each miControl As Control In ctrl.Controls
    resources.ApplyResources(miControl, miControl.Name)
    'Aplicar Tooltips multilinguaje (si hay)
    ToolTip1.SetToolTip(miControl, resources.GetString("tt" & miControl.Name))
    If miControl.HasChildren Then
      AplicarRecursosAControles(miControl)
    End If
  Next
End Sub
  
```

Iterando por cada control, y accediendo a sus hijos de forma recursiva, se consigue aplicar recursos multilinguaje a todos los controles visibles del formulario mediante una única llamada al método auxiliar *AplicarRecursosAControles*, pasando como parámetro el contenedor principal.

En caso de haber controles fuera de la jerarquía del contenedor principal, como en el caso de los controles *MenuStrip* y *ContextMenuStrip*, que representan menús y menús contextuales, se requieren llamadas adicionales al método tomando como parámetro dichos controles.

En el caso especial de los controles *ToolTip*, o pistas de utilidad, se aprovecha cada iteración del bucle para comprobar si el control a traducir cuenta con una clave de pista en el control *ToolTip*, y asignar el valor localizado correspondiente cargado desde el archivo de recursos. La clave de pista se identifica por el prefijo "tt" seguido del nombre del control.



A2.2 Gestión de calidad

La captura de las cotas de SolidWorks se ha realizado en el módulo iSW, mediante la nueva función CotasDeInspeccion:

```
Function CotasDeInspeccion(ByVal Componente As Object) As DataTable
```

El parámetro Componente es un puntero a un objeto que representa un componente de SolidWorks. La función retorna los valores de las cotas definidas como de 'inspección' definidas para dicho componente en una tabla de datos con las siguientes columnas:

- Nombre: nombre de la cota dentro de la configuración del componente SW (string).
- Nominal: valor nominal de la cota (decimal).
- Max: valor de la cota aplicando la tolerancia superior (decimal).
- Min: valor de la cota aplicando la tolerancia inferior (decimal).

Por otra parte, la captura e inserción de pautas de recepción, y de características de dichas pautas se ha realizado en el servicio conector de RPS2012, por medio de un par de funciones consulta-inserción: InfoPauta e InsertarPauta

```
// Consulta la pauta de recepción mas reciente asociada a un articulo.  
// Devuelve además de las características de la pauta,  
// su versión y estado de aprobación
```

```
Boolean InfoPauta(string usuario,  
                 string password,  
                 string idioma,  
                 string CodCompany,  
                 string CodPlanta,  
                 string CodArticulo,  
                 ref string Version,  
                 ref bool Aprobada,  
                 ref DataTable TablaCaracteristicas);
```

Los parámetros *usuario*, *password*, *idioma*, *CodCompany*, *CodPlanta* son comunes a todos los métodos y se utilizan como credenciales para crear una sesión en RPS2012 y poder realizar la lectura e inserción de entidades de forma segura. El resto de parámetros:

- CodArticulo: el código que identifica al artículo cuya pauta queremos consultar.
- Version: parámetro de salida con la versión de la pauta que queremos consultar.
- Aprobada: parámetro de salida que indica si la versión de pauta ha sido aprobada.
- TablaCaracterísticas: la tabla de datos que contiene las características de la versión de pauta.

La función devuelve verdadero si la consulta se ha realizado correctamente (aunque no se haya encontrado ninguna pauta) y falso si ha habido algún error durante la consulta.

```
// Intenta crear una nueva versión de pauta de recepción para un artículo.
// Si la versión de pauta más reciente en el ERP no está aprobada, no se
// podrán crear más versiones.
// Si la versión de pauta a crear ya existe en el ERP, pueden agregarse
// nuevas características si su nombre no se encuentra entre las características
// existentes (esté la versión aprobada o no).
```

```
Boolean InsertarPauta(string usuario,
    string password,
    string idioma,
    string CodCompany,
    string CodPlanta,
    string CodArticulo,
    string Version,
    string CodArticuloPlantilla,
    bool Copiar,
    DataTable TablaCaracteristicas,
    ref string MensajeRetorno);
```

Los parámetros *usuario*, *password*, *idioma*, *CodCompany*, *CodPlanta* son comunes a todos los métodos y se utilizan como credenciales para crear una sesión en RPS2012 y poder realizar la lectura e inserción de entidades de forma segura. El resto de parámetros:

- *CodArticulo*: Código del artículo cuya versión de pauta queremos crear o actualizar.
- *Version*: la versión de pauta a crear o actualizar.
- *CodArticuloPlantilla*: el código del artículo utilizado a modo de plantilla para la creación de la versión de pauta; debe contar con al menos una versión de pauta de recepción, que a su vez cuente con al menos una característica.
- *Copiar*: indica si se copiarán las características del artículo plantilla además de las contenidas en *TablaCaracteristicas*.
- *TablaCaracteristicas*: la tabla con las características de la pauta.
- *MensajeRetorno*: parámetro de salida que contiene un mensaje con información de error si la inserción no se ha realizado por alguna razón.

La función devuelve verdadero si la inserción se ha realizado con éxito, y falso en caso contrario. Para la inserción de los registros correspondientes a la versión de pauta y sus características, se han agregado las siguientes funciones auxiliares, las cuales se llaman desde *InsertarPauta*:

```
//Intenta crear una nueva versión de pauta de recepción para un artículo, o actualizar
una de //las existentes. Debe haber al menos una pauta de recepción registrada para el
artículo //plantilla, con al menos una característica.
```

```
private bool _CreateCharacControlVersion(
    string usuario,
    string password,
    string idioma,
    string CodCompany,
    string CodPlanta,
    string CodArticulo,
    string CodArticuloPlantilla,
    string Version,
    ref DataTable misCaracteristicas,
    ref string IDPauta,
    ref string IDPautaPlantilla,
    bool Copiar,
    ref string MensajeRetorno)
```

Los parámetros *usuario*, *password*, *idioma*, *CodCompany*, *CodPlanta* son comunes a todos los métodos y se utilizan como credenciales para crear una sesión en RPS2012 y poder realizar la lectura e inserción de entidades de forma segura. El resto de parámetros:

- *CodArticulo*: Código del artículo cuya versión de pauta queremos crear o actualizar.
- *CodArticuloPlantilla*: el código del artículo utilizado a modo de plantilla para la creación de la versión de pauta; el artículo plantilla debe contar con al menos una versión de pauta de recepción que contenga al menos una característica.
- *Version*: la versión de pauta a crear o actualizar.
- *misCaracteristicas*: parámetro de salida; la tabla con las características de la pauta a crear.
- *IDPauta*: parámetro de salida; cadena de caracteres con el identificador en base de datos de la pauta creada, o de la pauta a actualizar en caso de que la versión exista.
- *IDPautaPlantilla*: parámetro de salida; cadena de caracteres con el identificador en base de datos de una de las pautas de recepción definidas para el artículo plantilla.
- *Copiar*: parámetro que determina si se deben copiar las características de la pauta del artículo plantilla al generar la tabla de características.
- *MensajeRetorno*: parámetro de salida; contiene un mensaje con información de error si la inserción no se ha realizado por alguna razón.

Devuelve verdadero si la inserción se ha realizado con éxito, falso en caso contrario.

```
//Inserta las características pasadas como parámetro en la pauta creada por la función  
//_CreateCharacControlVersion.
```

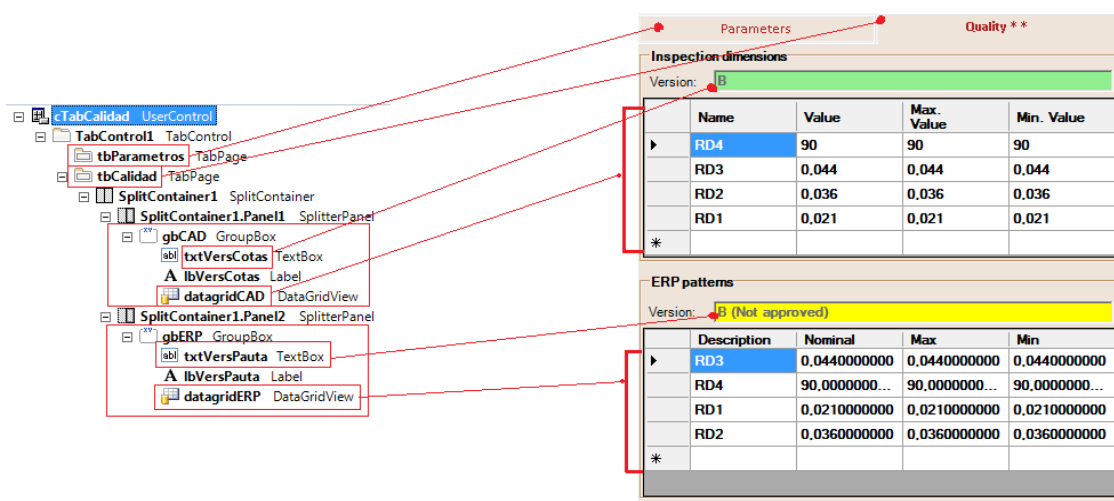
```
private bool _CreateCharacteristicsOfControl(  
    string usuario,  
    string password,  
    string idioma,  
    string CodCompany,  
    string CodPlanta,  
    string IDPauta,  
    string IDPautaPlantilla,  
    DataTable TablaCaracteristicas,  
    ref string MensajeRetorno)
```

Los parámetros *usuario*, *password*, *idioma*, *CodCompany*, *CodPlanta* son comunes a todos los métodos y se utilizan como credenciales para crear una sesión en RPS2012 y poder realizar la lectura e inserción de entidades de forma segura. El resto de parámetros:

- *IDPauta*: el identificador en base de datos de la pauta a la que añadir las características.
- *IDPautaPlantilla*: el identificador en base de datos de la pauta plantilla. Las características se generan clonando una de las características de esta plantilla y modificando sus propiedades en base a los datos de *TablaCaracteristicas*.
- *TablaCaracteristicas*: tabla de datos con los valores necesarios para generar nuevas características en la pauta a insertar o actualizar.
- *MensajeRetorno*: parámetro de salida; contiene un mensaje con información de error si la inserción no se ha realizado por alguna razón.

La función devuelve verdadero si la inserción de características se ha realizado de forma correcta, y falso en caso contrario.

Para gestionar la información de calidad relativa a las cotas y a las pautas desde el formulario de artículo, se ha diseñado un nuevo control a mostrar en el formulario de artículo: `cTabCalidad`, un control extendido a partir del control `.NET TabControl`:



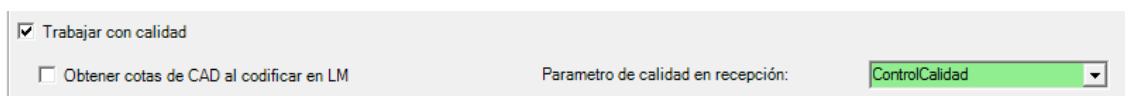
En el panel superior se visualizan las cotas de inspección capturadas del componente SW seleccionado, con sus valores nominales, máximos y mínimos. La versión de cota es igual a la revisión del artículo; si el artículo no tiene revisión, la versión de cota se mostrará en rojo, para indicar que no se pueden crear pautas a partir de las cotas. El `DataGridView` se rellena con la tabla de datos devuelta por la función `CotasDeInspeccion`, de `iCAD`.

En el panel inferior se visualizan las características de la última versión de pauta de recepción definida para el artículo. Si la versión actual no está aprobada, la versión se mostrará en amarillo junto con un mensaje que indica que la pauta no está aprobada. El `DataGridView` se rellena con la tabla de datos devuelta por la función `BuscarPautasRecepcion`, la cual a su vez llama al método `InfoPauta` del servicio conector `RPS2012`.

Al pulsar el botón Aceptar, la aplicación tratará de crear la versión de pauta indicada en el panel superior. Si coincide con la del panel inferior tratará de actualizar la pauta existente, agregando las nuevas características a la versión actual. El método que trata la tabla de cotas e invoca al método `InsertarPauta` del conector es la función `CrearPautaRecepcion` de `iRPS2012`.

Para registrar pautas de recepción asociadas a artículos, se requiere que el artículo asociado tenga activado el parámetro de RPS 'Calidad en Recepción'. Para ello, se incluye en el paquete de datos de artículo (la colección de parámetros de la integración con relevancia en `ERP_ART`) un parámetro, previamente definido para tal efecto, que modifique la columna `'QualityOnReception'` de la tabla `ArticleDetail` de la base de datos de `RPS2012`.

La gestión de la calidad se ha parametrizado en la configuración de la aplicación, y se puede acceder a ella a través del formulario de configuración general.



Si el `CheckBox` 'Trabajar con calidad' está desmarcado, no se capturarán cotas ni pautas relacionadas con el componente SW, y el formulario de artículo de la integración únicamente mostrará los

parámetros definidos en la sección de configuración general 'Presentación componente'. Para gestionar la calidad se requiere, además de habilitar el *CheckBox*, definir un parámetro que apunte a la propiedad *ArticleDetail.QualityOnReception* (booleano), el cual debe tener relevancia en ERP_ART.

(Articulo) Metodos del parametro : ControlCalidad

Read method: Write method:

ControlCalidad	False	STK	Article.ArticleDetail.QualityOnReception
----------------	-------	-----	--

A2.3 Archivo asociado en PDM

Para la captura de la ruta de archivos relacionados con un determinado componente SW, se ha implementado la siguiente función en el módulo iPDME (implementación de il_PDM):

```
Private Function BuscarReferenciaPorExtension(  
    ByVal Documento3D As String,  
    ByVal Extension As String) As String
```

- Documento3D: ruta local del acceso al documento PDM que contiene el componente SW cuyo archivo asociado queremos buscar.
- Extension: Extensión del archivo a buscar.

La función retorna la ruta local del primer documento PDM que cumpla con el criterio de búsqueda, según la prioridad establecida en el análisis. Si no se encuentra ningún archivo asociado al componente que cumpla con el criterio de búsqueda, se devuelve una cadena de caracteres vacía.

Para la búsqueda en los niveles superior e inferior de la estructura arborescente de referencias, se han implementado las funciones auxiliares:

```
Private Sub BuscarReferenciasAncestrosInmediatos(  
    ByVal Reference As IEdmReference5,  
    ByVal isTop As Boolean,  
    ByRef projName As String,  
    ByRef arrArchivos() As String)
```

- Reference: puntero para recorrer la lista de elementos referenciados de nivel inmediatamente superior al componente SW.
- isTop: booleano que indica si el componente SW es el nodo raíz de su estructura de referencias.
- projName: parámetro que contiene el nombre de la carpeta PDM bajo la que se ubica el documento al que apunta actualmente el puntero *Reference*.
- arrArchivos: parámetro de salida; array que contiene la lista de rutas de los documentos que forman parte de los elementos de nivel inmediatamente superior en la estructura de referencias del componente SW.

```
Private Sub BuscarReferenciasHijosInmediatos(  
    ByVal Reference As IEdmReference5,  
    ByVal isTop As Boolean,  
    ByRef projName As String,  
    ByRef arrArchivos() As String)
```

- Reference: puntero para recorrer la lista de elementos referenciados de nivel inmediatamente inferior al componente SW.
- isTop: parámetro local; booleano que indica si el componente SW es el nodo raíz de su estructura de referencias.
- projName: parámetro local que contiene el nombre de la carpeta PDM bajo la que se ubica el documento al que apunta actualmente el puntero *Reference*.
- arrArchivos: parámetro de salida; array que contiene la lista de rutas de los documentos que forman parte de los elementos de nivel inmediatamente inferior en la estructura de referencias del componente SW.

En una llamada a *BuscarReferenciaPorExtension*, se obtiene primero la lista de elementos de nivel inferior mediante *BuscarReferenciasHijosInmediatos* para cotejarlos con la extensión de archivo indicada. Si se encuentra un documento cuya extensión coincide, se devuelve su ruta local y termina la ejecución.

En caso de no hallar el documento deseado entre los hijos, se obtiene la lista de elementos de nivel superior mediante *BuscarReferenciasAncestrosInmediatos* y se cotejan las extensiones de las rutas de archivo devueltas. Por último, de no encontrarse el documento entre los padres, se hace una llamada a *BuscarReferenciasHijosInmediatos* por cada documento 'padre' obtenido, y se cotejan las extensiones en busca del documento PDM deseado.

Una nueva propiedad de la interfaz *il_PDM*, *ArchivoAsociado*, expone la funcionalidad para el resto de módulos de la aplicación:

```
ReadOnly Property ArchivoAsociado(  
    ByVal Documento3D As String, ByVal Extension As String) As String
```

El módulo *iPDME* implementa esta propiedad, llamando a la función *BuscarReferenciaPorExtension*:

```
Public ReadOnly Property ArchivoAsociado(  
    ByVal Documento3D As String, ByVal Extension As String) As String _  
    Implements iI_PDM.ArchivoAsociado  
    Get  
        Return BuscarReferenciaPorExtension(Documento3D, Extension)  
    End Get  
End Property
```

Por último, se añade una nueva propiedad a la interfaz *il_CAD*, *ArchivoAsociadoPDM*, la cual actúa como enlace con la propiedad *ArchivoAsociado* de *il_PDM*:

```
ReadOnly Property ArchivoAsociadoPDM(ByVal ComponenteSW As Object, _  
    ByVal Extension As String) As String
```

con su respectiva implementación en el módulo *iSW*:

```
Public ReadOnly Property ArchivoAsociadoPDM(  
    ByVal ComponenteSW As Object,  
    ByVal Extension As String) As String Implements iI_CAD.ArchivoAsociadoPDM  
    Get  
        Dim miPDM As iI_PDM = DatosConF.ConexionPDM  
        Dim swModelDoc As sldworks.ModelDoc2  
        Dim swConfiguracion As sldworks.Configuration  
        Dim sNombreDoc As String  
        If Componente2Modelo(ComponenteSW, swModelDoc, swConfiguracion) Then  
            sNombreDoc = swModelDoc.GetPathName  
            If Extension = "" Then Extension = ".SLDDRW"  
            Return miPDM.ArchivoAsociado(sNombreDoc, Extension)  
        Else  
            Return ""  
        End If  
    End Get  
End Property
```

Esto permite asignar la propiedad como método de lectura de parámetros desde el formulario de configuración CAD.

NombreDocu	C:\ITS_INTG\Proyectos\Pruebas MV\PiezaEjemplo.SLDPRT	System.String RutaFichero [System.O	T
NombreFicP	C:\ITS_INTG\Proyectos\Pruebas MV\PiezaEjemplo.PDF	System.String ArchivoAsociadoPDM [PDF	T
NombreFicPI	C:\ITS_INTG\Proyectos\Pruebas MV\PiezaEjemplo.SLDDRW	System.String ArchivoAsociadoPDM [SLDDRW	T
NombrePDM	PiezaEjemplo.PDM	System.String NombrePDM [System.O	T
NumeroSecu			T

Name : Default value :

Method :

Feature : Data type :

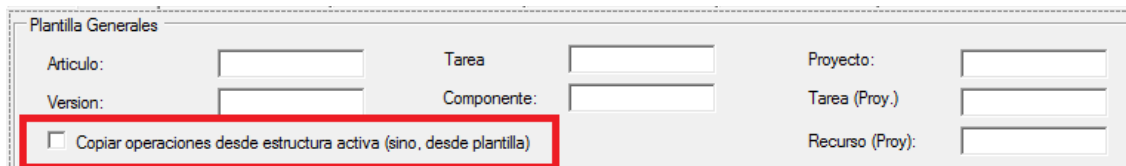
A2.4 Mantenimiento de operaciones y posiciones en reenvíos LdM

Se ha parametrizado en la configuración del módulo iRPS2012 la opción de copiar las operaciones de la estructura activa (iRPS2012::ConfiguracionERP):

```

Public Property CopiarOpsEstruActiva() As Boolean
  Get
    Return bCopiarOpsEstrucActiva
  End Get
  Set(ByVal value As Boolean)
    bCopiarOpsEstrucActiva = value
  End Set
End Property
  
```

La opción puede habilitarse desde un nuevo CheckBox introducido en la sección ERP del formulario de configuración correspondiente al módulo iRPS2012 (pestaña 'Plantillas'):



La opción es visible a través del objeto de configuración ERP:
miConfiguracion.CopiarOpsEstruActiva

La función InsertarOperacion del módulo iRPS2012 genera las operaciones de la estructura en base a las operaciones del artículo plantilla:

```

Public Function InsertarOperacion(
  ByVal Datos As Microsoft.VisualBasic.Collection, _
  ByVal colDatosPadre As Microsoft.VisualBasic.Collection, _
  ByVal IDTransaccion As Integer, _
  Mensajes As String) As Boolean
  Implements Declaraciones.ClasesBase.iI_ERP.InsertarOperacion
  ...

  TablaOperaciones = BuscarOperaciones(
    miConfiguracion.PlantillasERP.Articulo, _
    miConfiguracion.PlantillasERP.Version)
  ...

  For i = 0 To TablaOperaciones.Rows.Count - 1
    ...

    bResul = CrearOperacion(Datos, Mensajes)
    ...

  Next

  Return bResul
  
```

End Function

Para mantener las operaciones respecto a un envío anterior, se requiere modificar la cabecera de la función para incluir la revisión de la estructura activa en RPS0212 como parámetro opcional:

```
Public Function InsertarOperacion(
    ByVal Datos As Microsoft.VisualBasic.Collection, _
    ByVal colDatosPadre As Microsoft.VisualBasic.Collection, _
    ByVal IDTRansaccion As Integer, _
    Mensajes As String, _
    Optional ByVal RevERP As String = "") As Boolean
    Implements Declaraciones.ClasesBase.iI_ERP.InsertarOperacion
    ...
```

Este parámetro es necesario para identificar versión correcta de la estructura de la que extraer las operaciones a insertar, y se alimenta desde el formulario de lista de materiales.

La DataTable TablaOperaciones se carga con las operaciones asignadas a la estructura en RPS2012 en lugar de con las operaciones plantilla, y se genera además una nueva DataTable TablaComponentes, que contiene los componentes de la estructura activa:

```
If miConfiguracion.CopiarOpsEstruActiva Then
    TablaComponentes = BuscarComponentes(
        Datos(DatosConF.DatosLdM.LdMPadre).Valor, RevERP)
    TablaOperaciones = BuscarOperaciones(
        Datos(DatosConF.DatosLdM.LdMPadre).Valor, RevERP)
```

Una vez cargadas las tablas de operaciones y componentes, se itera por cada operación en TablaOperaciones y por cada componente en TablaComponentes en un bucle anidado, comprobando si los componentes de la propuesta de envío LdM actual concuerdan con los de la versión de estructura activa:

```
Dim dvTablaOps = New DataView(TablaOperaciones)
'iterar generando las operaciones en la nueva estructura
For Each RowViewOP As DataRowView In dvTablaOps
    ...

If TablaComponentes.Rows.Count > 0 Then
    For Each componente As DataRow In TablaComponentes.Rows
        ...

        'si el componente y la operación coinciden, incluir en la operación
        If componente.Item(DatosConF.DatosLdM.LdMHijo) =
            Datos(DatosConF.DatosLdM.LdMHijo).Valor _
            AndAlso componente.Item(DatosConF.DatosLdM.LdMOperacion) =
            operacion.Item(DatosConF.DatosLdM.LdMOperacion)
            Then
                For Each Columna As DataColumn In TablaOperaciones.Columns
                    ...
                    ClonDatos(Columna.ColumnName).Valor =
                    operacion.Item(Columna.ColumnName).ToString
                    Datos(Columna.ColumnName).Valor =
                    operacion.Item(Columna.ColumnName).ToString
                    ...
                Next
```

End If

```
'copiamos el resto de operaciones de la estructura activa
For Each Columna As DataColumn In TablaOperaciones.Columns

    If Columna.ColumnName <> DatosConF.DatosArticulo.REVARTICULO
    And Datos(Columna.ColumnName) IsNot Nothing
    Then
        Dim auxValor As Object = Datos(Columna.ColumnName).Valor
        If auxValor Is Nothing
        OrElse auxValor <> operacion.Item(Columna.ColumnName)
        OrElse TypeOf auxValor Is String AndAlso auxValor = ""
        Then
            ClonDatos(Columna.ColumnName).Valor =
                operacion.Item(Columna.ColumnName).ToString
        End If
    End If
Next

bResul = CrearOperacion(ClonDatos, Mensajes)

Next

...

Return bResul
''
End Function
```

A2.5 Siguiete Revisión PDM

PDM asigna revisiones a archivos en función de varios elementos:

- Componentes de revisión: se trata de estructuras de datos que constan de:
 - Cadena de formato: secuencia de caracteres que indica el formato del componente de revisión:
 - Alfabético
 - Numérico (número de dígitos variable)
 - Fecha / Hora
 - Hexadecimal
 - Componente de lista (lista de elementos definida por el usuario)
 - Incremento inicial: el valor con el que se inicializarán los primeros componentes de una revisión.
- Números de revisión: Consta de varios componentes de revisión, cuyo orden es definido por el usuario.
- Estado del flujo de trabajo: por cada estado del flujo de trabajo en PDM (y por cada transición entre estados) pueden definirse los formatos de número de revisión a emplear, así como la manera de incrementar dichos números de revisión.

El proyecto iPDME, que implementa la interfaz il_PDM, cuenta con un método de búsqueda de propiedades de documento alojados en PDM: BuscarDocumento

```
Private Function BuscarDocumento(
    ByVal Documento As String,
    ByVal funcion As Integer,
    ByVal Carpeta As String,
    Optional ByRef file As IEdmFile5 = Nothing) As String
```

En base al valor del parámetro *funcion*, la función BuscarDocumento retornará la propiedad deseada en formato textual:

```
'Si el parámetro funcion es 1 devuelve Propietario
'           Si es 2 devuelve Proyecto
'           Si es 3 devuelve Version
'           Si es 4 devuelve ruta completa del fichero
'           Si es 5 devuelve Revision
```

Se expande la funcionalidad de BuscarDocumento para devolver la siguiente revisión:

```
'           Si es 6 devuelve Next Revision
```

Para ello se agrega un *Case* adicional a la estructura *Select* que evalúa el valor de *funcion*:

```
Select Case funcion
...
Case 6 'devuelve siguiente revision
Return GetNextRevision(file, Conexion)
...
End Select
```

La nueva función *GetNextRevision* calcula la siguiente revisión que le corresponde al documento PDM en función de los números / componentes de revisión* que tenga asignados en el estado actual del flujo de trabajo* PDM.

GetNextRevision trabaja con los siguientes parámetros:

- File: puntero al objeto ledmFile5 (de la librería EdmLib de la API de Enterprise PDM) que representa el documento PDM cuya ruta local pasamos como parametro a BuscarDocumento
- Conexion: puntero al objeto EdmVault5 (de la librería EdmLib; API de Enterprise PDM) que representa el almacén PDM donde está alojado el documento.

Para calcular la siguiente revisión, se han identificado los objetos de la API de Enterprise PDM que intervienen en la asignación de revisiones:

- ledmRevisionMgr2: Objeto que gestiona números de revisión. Se utiliza para obtener los números y componentes de revisión.
- EdmRevNo: Objeto que representa un número de revisión. Un número de revisión se compone de una secuencia ordenada de identificadores en base de datos de los componentes de revisión que forman el número, un nombre descriptivo y un identificador del número de revisión en la base de datos.
- EdmRevComponent2: Objeto que representa un componente de revisión. Los componentes de revisión constan de:
 - Identificador en la base de datos.
 - Nombre del componente.
 - Una serie de flags (se trata como un número entero de 64 bits) que proporcionan información relativa al componente:
 - 1 --> Notificar a grupo en lugar de a usuario al agotar componente de lista
 - 2 --> Al agotar componente de lista, reiniciar lista
 - 4 --> Al agotar componente de lista, seguir utilizando el último elemento de la lista
 - 8 --> Al agotar componente de lista, abortar operación (no se puede incrementar la revisión)
 - 16 --> Al agotar componente de lista, notificar a usuario o grupo vía email
 - 32 --> El componente de revisión es de tipo cadena de formato
 - 64 --> El componente de revisión es de tipo lista
 - El identificador en la base de datos del usuario que recibirá un correo al llegar al final de un componente de revisión de tipo lista.
 - Valor inicial del componente de revisión.
 - Tipo de componente de revisión. Puede ser uno de dos tipos:
 - Lista de elementos: el valor del componente de revisión se toma de una lista de cadenas de caracteres definida por un usuario, separada por el carácter de nueva línea.
 - Cadena de formato: el valor del componente de revisión se determina en función de unos formatos predefinidos, identificados por una serie de caracteres:

%C	Caracteres alfabéticos en mayúsculas	A, B, C, ..., Z
%c	Caracteres alfabéticos en minúsculas	a, b, c, ..., z
%01d	Número entero ; al menos 1 dígito.	1,2,3, ...
%02d	Número entero; al menos 2 dígitos, rellenar con 0 (etc)	01, 02, 03, ...
%01x	Número hexadecimal; minúsculas; al menos 1 dígito	1,..., a, ..., f,

%02x	Número hexadecimal; minúsculas; al menos 2 dígitos (etc)	01, ..., 0a, ..., 0f, ...
%01X	Número hexadecimal; mayúsculas; al menos 1 dígito	1, ..., A, ..., F, ...
%02X	Número hexadecimal; mayúsculas; al menos 2 dígitos (etc)	01, ..., 0A, ..., 0F, ...
%4y	Año actual; 4 dígitos	2012, 2013, ...
%2y	Año actual; 2 dígitos	12, 13, ...
%01m	Número de mes actual; al menos 1 dígito	1, 2, ..., 12
%02m	Número de mes actual; al menos 2 dígitos	01, 02, ..., 12
%01D	Día del mes actual; al menos 1 dígito	1, 2, ..., 31
%02D	Día del mes actual; al menos 2 dígitos	01, 02, ..., 31
%01H	Hora actual; al menos 1 dígito	0, 1, ..., 23
%02H	Hora actual; al menos 2 dígitos	00, 01, ..., 23
%01M	Minuto actual; al menos 1 dígito	0, 1, ..., 59
%02M	Minuto actual; al menos 2 dígitos	00, 01, ..., 59
%01S	Segundo actual; al menos 1 dígito	0, 1, ..., 59
%02S	Segundo actual; al menos 2 dígitos	00, 01, ..., 59

Por otro lado, el objeto `ledmFile5` contiene una propiedad que devuelve la revisión completa actual del documento en una cadena de caracteres: `ledmFile5::CurrentRevision`

En base a lo expuesto, tras obtener el número de revisión asignado al documento PDM y sus componentes de revisión, se coteja la secuencia ordenada de componentes con la revisión actual por medio de una iteración que recorre la cadena de caracteres que contiene el valor actual de revisión.

Se obtiene el número de revisión actual por medio de los métodos `GetRevisionNumberIDFromState` y `GetRevisionNumbers` del objeto `ledmRevisionManager`:

```
Dim revMgr As IEdmRevisionMgr2 = vault.CreateUtility(EdmUtility.EdmUtil_RevisionMgr)
Dim IDRevNo As Integer = revMgr.GetRevisionNumberIDFromState(file.CurrentState.ID)
'revNo es un parámetro de salida: lista de punteros a números de revisión
revMgr.GetRevisionNumbers(IDRevNo, revNo)
```

se obtiene también el valor actual de la revisión para el documento PDM:

```
Dim actual As String = file.CurrentRevision
```

Los componentes de los números de revisión se extraen iterando por cada número de revisión obtenido, por medio del método `GetRevisionNumberComponents2` del objeto `ledmRevisionMgr2`:

```
'componentes es un parámetro de salida: lista de punteros a componentes de revisión
revMgr.GetRevisionNumberComponents2(IDRevNo, componentes)
```

Una vez obtenidos y ordenados los componentes de revisión a manejar, se cotejan con el valor actual de revisión para el documento PDM. Si la revisión actual está vacía, se recorre la lista de componentes de revisión, generando el primer valor correspondiente a cada uno por medio de la nueva función auxiliar `GetFirstRevNoComponentValue`:

```
Private Function GetFirstRevNoComponentValue(
    ByVal initialCounter As Integer,
    ByVal format As String) As String 'primer valor de un componente de revisión
```

- `initialCounter`: El valor a partir del cual se debe generar el primer componente; se obtiene de una propiedad del objeto `EdmRevComponent2`.

- **format:** la serie de caracteres que identifica el formato a utilizar en la generación del componente.

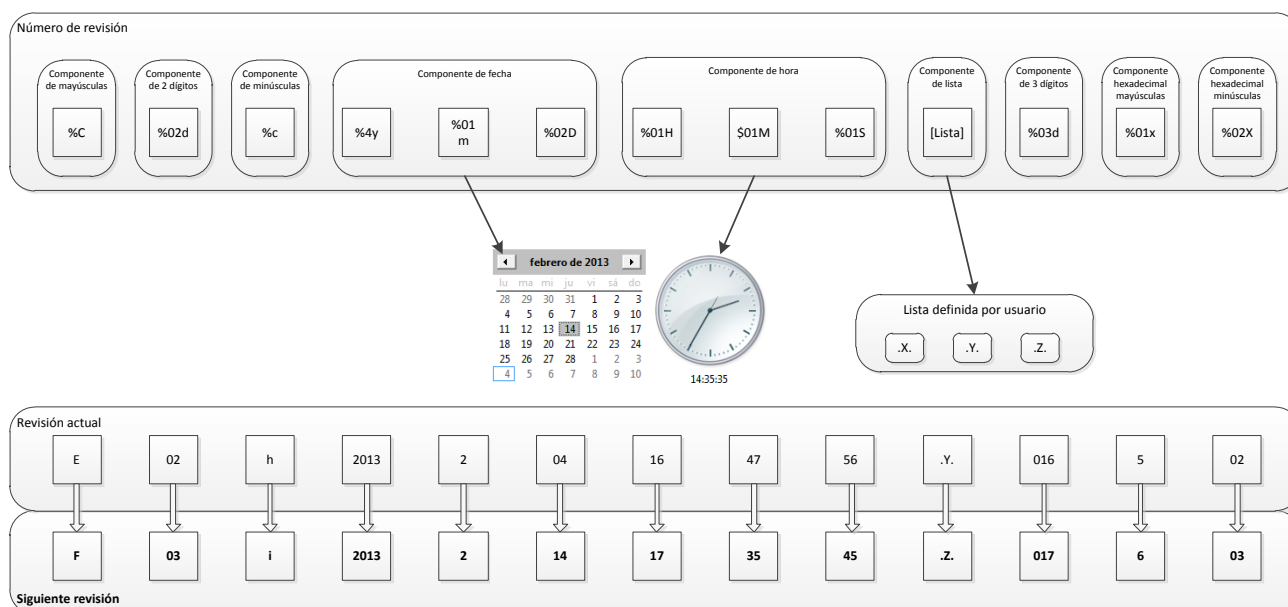
En caso de no estar vacío el campo de revisión del archivo, se recorre la cadena cotejando los caracteres generados por cada componente, y se calcula el valor de cada siguiente componente mediante la función *GetNextRevNoComponentValue*:

```

Private Function GetNextRevNoComponentValue(
  ByVal value As String,
  ByVal format As String,
  ByVal increment As Integer) As String 'siguiente valor de un componente de
revisión

```

- **value:** cadena de caracteres con el valor actual del componente de revisión a tratar.
- **format:** cadena de caracteres que identifica el formato del componente de revisión a tratar.
- **increment:** valor de paso para en base al cual se incrementará el valor actual del componente de revisión.



A2.6 Filtrar parámetros en formulario de artículo

Se han parametrizado en ClasesBase::DefParametroForm las nuevas propiedades de filtro correspondientes a la presentación de la ficha de artículo.

```
Public Class DefParametroForm
```

```
...
```

```
Dim sParFiltro As String
Dim sValFiltro As String
```

```
...
```

```
Public Property ParFiltro() As String
  Get
    Return sParFiltro
  End Get
  Set(ByVal value As String)
    If value Is Nothing Then
      sParFiltro = ""
    Else
      sParFiltro = value
    End If
  End Set
End Property

Public Property ValorVisible() As String
  Get
    Return sValFiltro
  End Get
  Set(ByVal value As String)
    If value Is Nothing Then
      sValFiltro = ""
    Else
      sValFiltro = value
    End If
  End Set
End Property
End Class
```

Nombre	ParamFiltro	ValorVisible
NombreDocumento		
SiguienteRevision	ControlCalidad	False
DescripcionIng		
TipoComponente		
TipoArticulo		
LineaDeProducto		
Familia		
Peso	Cantidad	- 0
Unidades	TipoArticulo	1 3
PlazoAprob	TipoArticulo	- 1 3
PrecioMedio		
Observaciones		
DesglosarCompo...		
Color		
RevisionPDM	ControlCalidad	True

Las celdas de la columna ParamFiltro son de tipo *ComboBox*, y sus posibles valores se obtienen de la lista de parámetros de la integración que tengan relevancia en ERP_ART, mediante una nueva función en Utilidades:

```
Public Function ColumnaParFiltroDatosForm(
  ByRef dgv As DataGridView,
  ByVal parametros As Collection) As DataGridViewComboBoxColumn
  Dim columnaParFiltros As New DataGridViewComboBoxColumn
  Dim paramsArticulo As New Collection
  paramsArticulo.Add("")
  For Each miParam As DefParametroIntegracion In parametros
    If miParam.ERP_ART <> "" OrElse miParam.ERP_ART <> "-" Then
      paramsArticulo.Add(miParam.Nombre)
    End If
  Next
  columnaParFiltros.DataSource = paramsArticulo
  dgv.Columns.Add(columnaParFiltros)
```

End Function

Los parámetros a mostrar se filtran en el control *ucFichaArticulo*, donde se evalúa el valor de cada parámetro y se compara con su valor de filtro:

'Genera los controles del formulario de artículo destinados a la visualización y edición de 'valores de los parámetros de artículo.

```
Private Sub CrearFormArticulo()
```

```

For Each miParamForm In misDatosConfiguracion.DatosFormulario.Parametros
  'Filtrar parametros a mostrar en funcion de parametrizacion
  If miParamForm.ParFiltro <> "" AndAlso miParamForm.ValorVisible <> "" Then
    'Si el parametro no se cuenta entre los parametros ERP, omitir
    If Not misDatosERP.Contains(miParamForm.ParFiltro) Then Continue For
    Dim miValFiltro As String = miParamForm.ValorVisible
    Dim bFiltrar As Boolean = True
    'Comprobar si el filtro es de exclusión o de inclusión
    Dim bExcluir As Boolean = miValFiltro.StartsWith("-")
    'Eliminar el carácter de guión si es filtro de exclusión
    If bExcluir Then miValFiltro = miValFiltro.TrimStart("-").Trim()
    Dim valoresFiltro() As String = Array.CreateInstance(GetType(String), 1)
    'Comprobar si el valor de filtro es único o múltiple
    If miValFiltro.Contains("|") Then
      'Valores múltiples de filtro
      valoresFiltro = miValFiltro.Split("|")
    Else
      'Valor único de filtro
      valoresFiltro(0) = miValFiltro
    End If
    'Si es filtro de exclusión:
    If bExcluir Then
      bFiltrar = False
      For Each valFiltro As String In valoresFiltro
        If misDatosERP(miParamForm.ParFiltro).Valor = valFiltro.Trim()
        Then
          'Excluir el parámetro
          bFiltrar = True
          Exit For
        End If
      Next
    'Si es filtro de inclusión
    Else
      bFiltrar = True
      For Each valFiltro As String In valoresFiltro
        If misDatosERP(miParamForm.ParFiltro).Valor = valFiltro.Trim()
        Then
          'Incluir el parámetro
          bFiltrar = False
          Exit For
        End If
      Next
    End If
    If bFiltrar Then
      'Agregar el parámetro a la colección de parámetros excluidos
      misDatosFiltrar.Add(misDatosERP(miParamForm.Nombre),
        miParamForm.Nombre)
      'Omitir el paso de agregar control al formulario
      GoTo siguiente
    End If
  End If
  'Fin filtrar
Next
...
End Sub

```

Además de omitir la creación y presentación de los controles en el formulario de artículo, también se omiten los parámetros excluidos en el paso de registrar el artículo en ERP. Adicionalmente, si un parámetro excluido tiene asociado otro parámetro (como es el caso de los parámetros de filtro utilizados en la creación de características mediante *InsertarArticleFeature*):

```

'Extrae los valores introducidos en el formulario de artículo y se los asigna a la
colección 'de parámetros utilizada para el registro del artículo en ERP
Private Sub LeerFormArticulo()
...

Dim miDatoERP As DefParametroAplicacion
For Each miDatoERP In misDatosERP 'Colección de datos para registrar artículo
  Select Case miDatoERP.Nombre
    ...

  Case Else
    Try
      'Comprobar si el parámetro está en la colección de excluidos
      If misDatosFiltrar.Contains(miDatoERP.Nombre) OrElse _
        'Comprobar si el parámetro tiene un parametro de filtro excluido
        ((miDatoERP.DefValor IsNot Nothing AndAlso miDatoERP.DefValor <>
""))
        AndAlso misDatosFiltrar.Contains(miDatoERP.DefValor)) Then
          'Eliminar el parámetro de la colección a enviar
          misDatosERP.Remove(miDatoERP.Nombre)

        Else
          miDatoERP.Valor = LineasForm.Item(miDatoERP.Nombre).valor
        End If
      Catch ex As Exception

    End Try
  End Select
Next miDatoERP

...

End Sub
  
```

A2.7 Codificación manual de artículos en RPS2012

Para registrar artículos con código especificado por el usuario, se requiere deshabilitar temporalmente el mecanismo de codificación definido para una empresa en RPS2012. Para ello, se ha identificado la entidad en RPS2012 responsable de asignar códigos en base a series de numeración:

STKParameterStock

Se ha modificado la función `InsertarArticle2` del servicio conector RPS2012 agregando un nuevo parámetro de entrada: `CodificacionAutomática`.

```
public Boolean InsertarArticle2(  
    string usuario,  
    string password,  
    string idioma,  
    string CodCompany,  
    string CodPlanta,  
    string EntidadExt,  
    string CodArticuloPlantilla,  
    ref string CodArticulo,  
    bool CodificacionAutomatica,  
    string[] ListadeDatos1,  
    string[] ListadeDatos2,  
    ref string MensajeRetorno,  
    byte[] Imagen)
```

Los parámetros *usuario*, *password*, *idioma*, *CodCompany*, *CodPlanta* son comunes a todos los métodos y se utilizan como credenciales para crear una sesión en RPS2012 y poder realizar la lectura e inserción de entidades de forma segura. El resto de parámetros:

- *EntidadExt*: nombre de la entidad extendida a modificar. Si está vacío, se trata de la entidad base *STKArticle*.
- *CodArticuloPlantilla*: código del artículo plantilla a partir del cual vamos a generar la nueva entrada.
- *CodArticulo*: parámetro de entrada/salida; contiene el código con el que se registra el artículo en el caso de inserción de nuevo artículo. Si el código corresponde a un artículo existente, se actualizará la información de dicho artículo.
- **CodificacionAutomatica**: Si éste parámetro es verdadero, el registro del artículo se realizará en base a la serie de numeración definida; si el parámetro es falso, el código con el que se registrará el artículo será el indicado en el parámetro *CodArticulo*.
- *ListadeDatos1*: lista de cadenas de caracteres que contienen las propiedades del artículo a generar o modificar.
- *ListadeDatos2*: lista de cadenas de caracteres que contienen los valores de las propiedades del artículo a generar o modificar.
- *MensajeRetorno*: parámetro de salida; contiene información sobre el error en caso de que la inserción / actualización haya fallado.
- *Imagen*: lista de bytes que representan la imagen en miniatura del artículo capturada desde SolidWorks.

La función devuelve verdadero si la inserción o actualización se han realizado con éxito, y falso en caso contrario.

La función `InsertarArticle2` llama al método auxiliar `_CreateArticle`, el cual genera una nueva entidad `STKArticle` en base a los datos proporcionados. Esta función también se ha modificado agregando el parámetro adicional `CodificacionAutomatica`:

```
private bool _CreateArticle(  
    string CodCompany,  
    ref string CodArticulo,  
    bool CodificacionAutomatica,  
    string CodArticuloPlantilla,  
    string EntidadExt,  
    string[] ListadeDatos1,  
    string[] ListadeDatos2,  
    byte[] Imagen,  
    ref string MensajeRetorno)
```

Parámetros:

- `CodCompany`: código de la empresa bajo la que registrar el nuevo artículo.
- `CodArticulo`: parámetro de entrada/salida; código con el que se ha registrado el nuevo artículo.
- **`CodificacionAutomatica`**: Si éste parámetro es verdadero, el registro del artículo se realizará en base a la serie de numeración definida; si el parámetro es falso, el código con el que se registrará el artículo será el indicado en el parámetro `CodArticulo`.
- `CodArticuloPlantilla`: código del artículo plantilla a partir del cual vamos a generar la nueva entrada.
- `EntidadExt`: nombre de la entidad extendida a modificar. Si está vacío, se trata de la entidad base `STKArticle`.
- `ListadeDatos1`: lista de cadenas de caracteres que contienen las propiedades del artículo a generar o modificar.
- `ListadeDatos2`: lista de cadenas de caracteres que contienen los valores de las propiedades del artículo a generar o modificar.
- `Imagen`: lista de bytes que representan la imagen en miniatura del artículo capturada desde `SolidWorks`.
- `MensajeRetorno`: parámetro de salida; contiene información sobre el error en caso de que la inserción / actualización haya fallado.

La función devuelve verdadero si la inserción se ha realizado con éxito, y falso en caso contrario. Será esta función la encargada de generar una transacción en la cual:

1. Se comprueba el valor de `CodificacionAutomatica`
2. Si `CodificacionAutomatica` es verdadero, se deshabilita la serie de numeración de la empresa identificada por `CodCompany`.
3. Se registra el artículo.
4. Si `CodificacionAutomatica` es verdadero, se restaura la serie de numeración de la empresa identificada por `CodCompany`

Si alguna de las operaciones falla, se anula la transacción y la base de datos vuelve al estado previo a la llamada a `_CreateArticle`; por el contrario, si todo va bien la transacción se completa y se confirman los cambios en la base de datos.

```

private bool _CreateArticle(string CodCompany, ref string CodArticulo,
bool CodificacionAutomatica, string CodArticuloPlantilla, string EntidadExt,
string[] ListadeDatos1, string[] ListadeDatos2, byte[] Imagen,
ref string MensajeRetorno)
{
    ...

    //Transaccion
    TransactionOptions tranOptions = new TransactionOptions();
    tranOptions.Timeout = TimeSpan.FromSeconds(60);
    tranOptions.IsolationLevel = System.Transactions.IsolationLevel.ReadCommitted;
    TransactionScope ts =
        new TransactionScope(TransactionScopeOption.Required, tranOptions)

    using (ts)
    {

        //Limpiar serie de numeración para codificación manual
        Basoa.Services.Warehouse.Entities.ParameterStock enParameterStock = null;
        var blParameterStock =
            BusinessLogicConfiguration.Create<Warehouse.Contracts.IParameterStock>();
        string IDSerieNumeracion = "";
        if (!CodificacionAutomatica)
        {
            var cr = new CriteriaSet
            {
                new Criteria(
                    Basoa.Services.Warehouse.Entities.ParameterStock.Properties.CodCompany,
                    ComparisonOperators.Equals,
                    CodCompany)
            };
            enParameterStock = blParameterStock.GetEntity(cr)[0];
            if (enParameterStock != null)
            {
                IDSerieNumeracion = enParameterStock.IDCodingSerieArticle;
                enParameterStock.IDCodingSerieArticle = null;
                if (!blParameterStock.UpdateWithCheck(
                    ref enParameterStock,
                    ref errDet))
                {
                    MensajeRetorno = string.Join(Environment.NewLine,
                        errDet.Select(err => err.Message));
                }
            }
        }
        //Fin limpiar serie numeración
        //Registrar el artículo
        result = blArticle.InsertWithCheck(ref newArticle, ref errDet);
        if (!result)
        {
            MensajeRetorno = string.Join(Environment.NewLine,
                errDet.Select(err => err.Message));
        }
        else
        {
            CodArticulo = newArticle.CodArticle;
            //Restaurar serie de numeración
            if (!CodificacionAutomatica)
            {
                enParameterStock.IDCodingSerieArticle = IDSerieNumeracion;
                if (!blParameterStock.UpdateWithCheck(ref enParameterStock, ref errDet))
                {
                    MensajeRetorno = string.Join(Environment.NewLine, errDet.Select(err =>
                        err.Message));
                }
            }
            //Fin restaurar; validar transacción si todo ha ido bien
            if (result) { ts.Complete(); }
        }
    }

    ...
}
  
```

A2.8 Trabajo con grupos empresariales en RPS2012

Para la gestión del trabajo con multiempresa, el servicio conector RPS2012 debe ser capaz de capturar la empresa matriz a la que pertenece una empresa filial determinada. Para ello se ha agregado la siguiente función:

```
// Devuelve el código de la empresa matriz de la empresa filial identificada por
CodCompany.
// Si la empresa no tiene matriz, devuelve una cadena de caracteres vacía.
private string EmpresaMatriz(string CodCompany, ref string CodPlantaMatriz)
```

Parámetros:

- CodCompany: cadena de caracteres con el código de la empresa filial cuyo código de empresa matriz queremos obtener.
- CodPlantaMatriz: parámetro de salida; contiene el código de la planta por defecto de la empresa matriz cuyo código devuelve la función.

Para realizar las consultas en un entorno multiempresa, se han modificado las funciones indicadas en el análisis para que obtengan los datos asignados a la empresa matriz de forma recursiva, en caso de no encontrar registros para la empresa filial:

```
public Boolean InfoEntidadE(string usuario, string password, string idioma,
    string CodCompany, string CodPlanta, string Modulo, string Entidad,
    string EntidadExt, string[] ListadeFiltros1, string[]
ListadeFiltros2,
    string Operador, ref DataTable Retorno)
{
    //componer consulta
    ...
    // ejecutar consulta y rellenar tabla Retorno
    ...
    if (Retorno.Rows.Count == 0)
    {
        string CodPlantaMatriz = "";
        string CodMatriz = EmpresaMatriz(CodCompany, ref CodPlantaMatriz);
        if (CodMatriz == "" || CodPlantaMatriz == "") { return true; }

        return InfoEntidadE(usuario, password, idioma, CodMatriz, CodPlantaMatriz,
            Modulo, Entidad, EntidadExt,
            ListadeFiltros1, ListadeFiltros2, Operador, ref
Retorno);
    }
    return true;
}
```

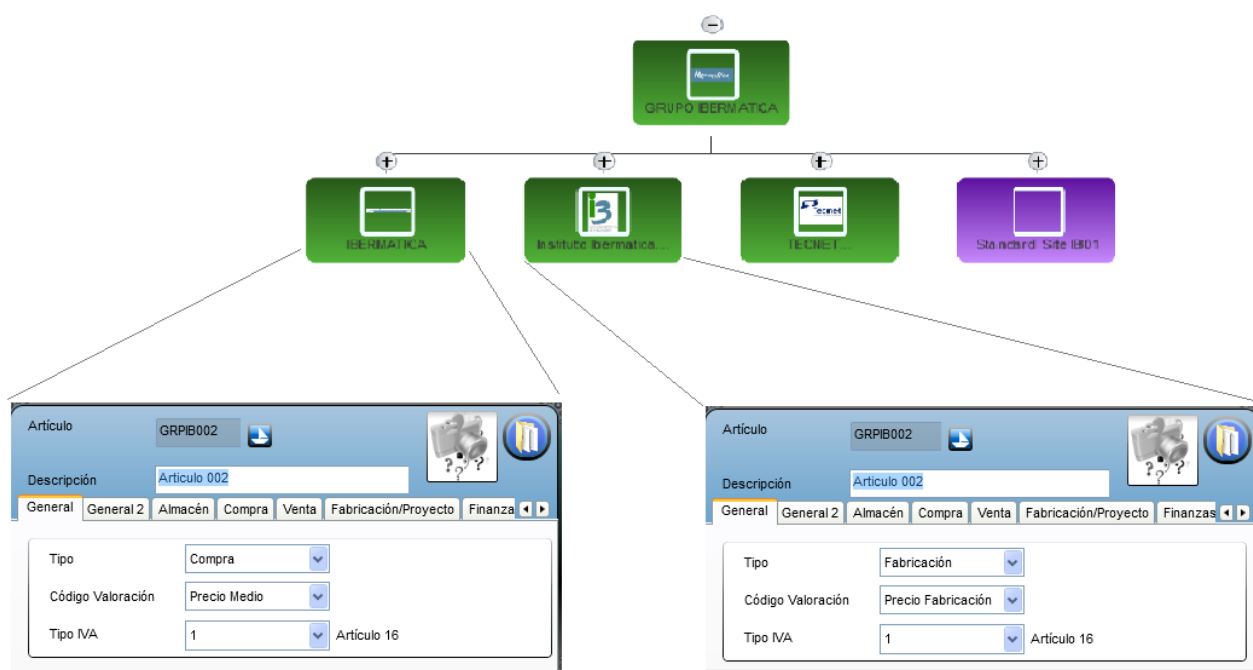
Esta modificación, con variaciones en función de la naturaleza de cada una, se ha aplicado sobre las funciones:

- InfoEntidadE
- InfoArticleByLang

- InfoArticlePicture
- InfoArticleFeature
- InsertarEstructura
- InsertarMaterialEstructura
- InsertarMaterialProy

La nueva función *InsertarArticleDetail* genera una nueva entrada en la tabla *STKArticleDetail*, donde se guarda información detallada de cada artículo, cuyo contenido es variable de cara a cada una de las empresas del grupo empresarial.

	CodArticle	Description	CodCompany	Type	ValuationCostType
1	GRPIB002	Artículo 002	IB10	1	2
2	GRPIB002	Artículo 002	IB11	2	3



// Intenta generar una entrada de detalles específicos de la empresa indicada, para el //artículo indicado.

```

public Boolean InsertarArticleDetail(
    string usuario,
    string password,
    string idioma,
    string CodCompany,
    string CodPlanta,
    string CodArticulo,
    string[] ListadeDatos1,
    string[] ListadeDatos2,
    ref string MensajeRetorno)
  
```

Los parámetros *usuario*, *password*, *idioma*, *CodCompany*, *CodPlanta* son comunes a todos los métodos y se utilizan como credenciales para crear una sesión en RPS2012 y poder realizar la lectura e inserción de entidades de forma segura. El resto de parámetros:

- CodArticulo: cadena de caracteres con el código del artículo cuya entrada de detalles queremos crear o actualizar
- ListadeDatos1: lista de cadenas de caracteres, cuyos componentes equivalen a las propiedades (columnas de base de datos) de la entidad que queremos generar / modificar
- ListadeDatos2: lista de cadenas de caracteres, cuyos componentes contienen los valores de las propiedades a generar / modificar.
- MensajeRetorno: parámetro de salida; cadena de caracteres con una descripción del error acaecido si la inserción falla

La función devuelve verdadero si la inserción / actualización se ha realizado con éxito, y falso en caso contrario.

Para el trabajo con multiempresa desde la integración, al trabajar desde una empresa filial, se requiere modificar las siguientes funciones de inserción para que generen entradas de detalles de artículo particulares a la empresa filial, siempre que el artículo en cuestión exista en la empresa matriz.

- InsertarEstructura2: función para generar la cabecera de una estructura a partir de un artículo para una empresa determinada.
- InsertarMaterialEstructura2: función para agregar entradas de material de estructura a partir de artículos, para una estructura determinada.
- InsertarMaterialProy2: función para agregar entradas de material de proyecto a partir de artículos, para una estructura determinada.

```

public Boolean InsertarEstructura2(string usuario, string password, string idioma,
string CodCompany, string CodPlanta, string EntidadExt, string CodArticulo,
string VersionEstruc, string[] ListadeDatos1, string[] ListadeDatos2,
ref string MensajeRetorno)
{
    ...
    //Si el articulo no existe para la empresa filial, comprobar si existe para la
    matriz
    if (GetArticleIDByCode(CodCompany, CodArticulo) == "")
    {
        string CodPlantaMatriz = "";
        string CodCompanyMatriz = EmpresaMatriz(CodCompany, ref CodPlantaMatriz);
        //Si el articulo no existe para la empresa filial, pero existe para la
        matriz, //crear entrada para empresa filial
        if (GetArticleIDByCode(CodCompanyMatriz, CodArticulo) == "")
        {
            if (!InsertarArticleDetail(usuario, password, idioma, CodCompany,
            CodPlanta,
            CodArticulo,
            ListadeDatos1,
            ListadeDatos2,
            ref MensajeRetorno))
            {
                MensajeRetorno +=
                "No se ha podido crear la entrada del articulo " +
                CodArticulo +
                " en la empresa " +
                CodCompany;
                return false;
            }
        }
    }
}
  
```

El resto de métodos se han modificado de forma análoga, con variaciones en función de la naturaleza de cada función.

A1.9 Comprobación de licencia para utilizar el servicio conector RPS2012

Las funciones de la integración encargadas del chequeo de seguridad son las siguientes (VB.NET):

```
Public Function DameNumeroSeriePDMW() As String
'Devuelve el número de serie de SolidWorks almacenado en el registro del sistema

Function ChequearSeguridadListaSNPDMW(ByVal RutaLista As String) As Integer
'Coteja el número de serie con las claves encriptadas en el archivo
'<RutaLista>\ListaSNPDMW.txt y devuelve un código de retorno en función del resultado

Private Function Encriptar(ByVal v_strOrigen As String) As String
'Encripta un string empleando el método de Cifrado Base64 y la clave P_CTE_CLAVE_BASE64.

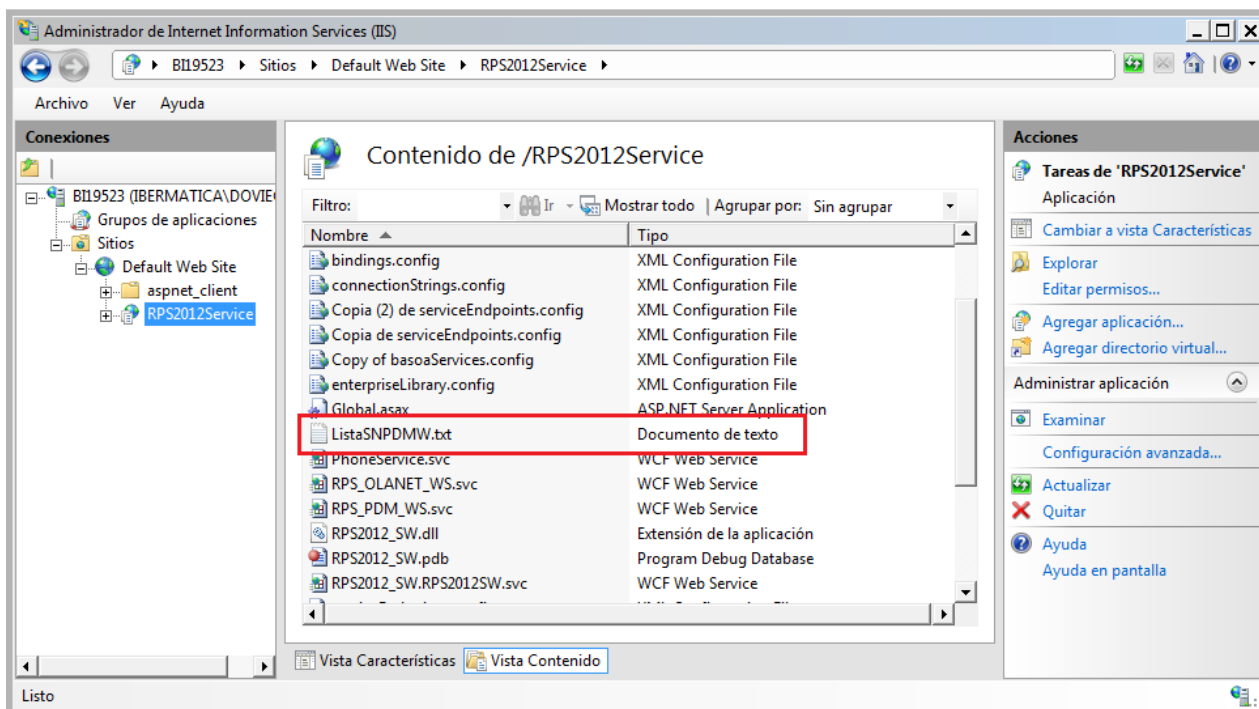
Private Function Desencriptar(ByVal v_strEncriptado As String) As String
'Desencripta un string empleando el método de Cifrado Base64 y la clave
P_CTE_CLAVE_BASE64.
```

Exceptuando la función *DameNumeroSeriePDMW*, el resto de funciones se agregan al código fuente del servicio RPS2012; el número de serie se pasa como parámetro adicional de la función

ValidarUsuario:

```
Boolean ValidarUsuario(
    string usuario,
    string password,
    string idioma,
    string CodCompany,
    string CodPlanta,
    string NumSerie);
'Comprueba que las credenciales de acceso a RPS2012 son válidas. Si la validación falla,
no 'podrá utilizarse el servicio.
```

NumSerie se pasa como parámetro a la función *_ChequearSeguridad*, construida a partir de la función original de la integración *ChequearSeguridadSNPDMW*. Dado que la ruta del archivo de claves será siempre la carpeta raíz del servicio RPS2012, no necesitamos especificarla.



```
private int _ChequearSeguridad(string numSerie)
```

'Coteja el número de serie con las claves encriptadas en el archivo de claves. Si el número es válido, el código de retorno es 0

- numSerie: cadena de caracteres con el número de serie enviado desde el cliente conectado al servicio.

El resto de funciones no sufren modificaciones (C#):

```
private string _Encriptar(string v_strOrigen)
```

- v_strOrigen: cadena de caracteres de longitud 16 sin encriptar

```
private string _Desencriptar(string v_strEncriptado)
```

- v_strEncriptado: cadena de caracteres encriptada según el cifrado Base64

Tras comprobar las credenciales de acceso a RPS2012, se verifica el número de serie. Si este falla, se realiza una nueva verificación contra las direcciones MAC de los adaptadores de red del servidor, para determinar si se trata de un servidor autorizado.

La verificación de la dirección MAC se realiza de forma análoga a la verificación de un número de serie, ya que las MAC encriptadas se almacenan junto con el resto de claves en el mismo archivo ListaSNPDMW.txt

```
foreach (string DirMAC in _ObtenerDireccionesMAC())
{
    if (_ChequearSeguridad(DirMAC) == 0) {return true;}
}
```

El nuevo método `_ObtenerDireccionesMAC()` genera una colección de cadenas de caracteres con las direcciones MAC de los adaptadores de red operativos del servidor:

```
private Collection _ObtenerDireccionesMAC()
{
```

```
NetworkInterface[] interfaces = NetworkInterface.GetAllNetworkInterfaces();
Collection resul = new Collection();
foreach (NetworkInterface interfaz in interfaces)
{'iterar entre todos los adaptadores de red del equipo servidor

    string aux = ""; 'variable para almacenar la dirección MAC del adaptador
    'si es un adaptador virtual, omitir
    if (interfaz.NetworkInterfaceType == NetworkInterfaceType.Loopback)
continue; 'si el adaptador esta desconectado, omitir
    if (interfaz.OperationalStatus != OperationalStatus.Up) continue;
    'obtener dirección física del adaptador
    aux = interfaz.GetPhysicalAddress().ToString();
    'limpiar caracteres de separación
    aux = aux.Replace(":", "").Replace("-", "");
    'rellenar con ceros hasta la longitud deseada (16 caracteres)
    aux = aux.PadLeft(16, '0');
    'agregar dirección MAC a la colección de retorno
    resul.Add(aux, interfaz.Name);
}
return resul;
}
```