



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Evolución y personalización de software en un marco  
Model-Driven Development

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Trujillo García, Sergio

Tutor/a: Letelier Torres, Patricio Orlando

CURSO ACADÉMICO: 2021/2022



# Agradecimientos

En primer lugar, agradecer a mi familia, ya que sin ellos no sería quien soy hoy en día, y no hubiera podido tener la oportunidad de estudiar a 2000 kilómetros de mi hogar, agradezco el sacrificio mostrado por su parte y el apoyo que me han brindado en esta experiencia.

Continuando, he de agradecer a los compañeros que he tenido en la carrera, empezando por las personas con las que he convivido estos cuatro años, por hacerme más amena la trayectoria y haber tenido una excelente convivencia, y a mis compañeros de clase, entre los que nos hemos apoyado increíblemente a lo largo del camino, algunos desde el primer momento, y otros nos hemos ido conociendo, pero todos nos hemos aportado algo entre nosotros.

Agradecer también al que ha sido mi tutor tanto de prácticas como del trabajo, Patricia, del cual obtuve además desde el principio la oportunidad de entrar en el departamento de I+D+i de la empresa donde he realizado prácticas. También a Francisco, el jefe de dicho departamento, por el trato y ayuda recibido, y en especial la oportunidad de realizar este trabajo que suponía un gran desafío.

Por último, hacer mención de mis compañeros de departamento, en especial a Juan Vicente, Víctor y Carlos, ya que han sido los que más me han ayudado, Juanvi siendo mi mentor al principio del proceso de prácticas, y Víctor y Carlos mostrándome a fondo las *DSL Tools*.

## Resum

El present treball ha sigut realitzat en context de pràctiques en una empresa, el producte principal de la qual és un ERP per al sector soci-sanitari, pertanyent l'autor d'aquest document al departament d'I+D+i, participant en el desenvolupament d'una nova versió d'aquest basada en *MDD*.

La finalitat d'aquest TFG és desenvolupar un algorisme de mescla de models generats amb les *DSL Tools* de Microsoft en format XML per a una aplicació amb arquitectura de microserveis i que basa el seu desenvolupament en generació automàtica de codi a partir de models amb les anteriorment esmentades *DSL Tools*.

Aquest algorisme serà l'encarregat de mesclar de manera intel·ligent els models que hagen rebut canvis en les versions del producte dels clients, amb la versió del producte estàndard, per a així poder incloure els seus canvis desitjats mentre es tenen els nous facilitats pel desenvolupament estàndard de l'aplicació. Així es permetrà la personalització del *programari* per part dels clients alhora que mantenen la seua versió del producte actualitzada.

A més, s'implementaran en les *DSL Tools* proteccions de models, per a així evitar que en els canvis afegits per a les versions del client no s'alteren elements procedients del producte estàndard de formes indegudes, limitant alhora els canvis que es poden afegir a la versió del client, tot en funció de l'especificació establida.

En el desenvolupament, de l'algorisme s'ha emprat com a tecnologia principal .NET Core, amb els aspectes que ens ofereix C#, el llenguatge de programació utilitzat, i s'ha fet especial ús de les biblioteques bàsiques de tractat de XML facilitades per C#. Per a les proteccions dels models s'han modificat les *DSL Tools* creades pel propi equip d'I+D+i.

**Paraules clau:** MDD, Mescla de models, DSL Tools, Generació automàtica de codi, microserveis Protecció de models, .NET, C#, XML

---

## Resumen

El presente trabajo se realiza en contexto de prácticas en una empresa cuyo producto principal es un ERP para el sector socio-sanitario, perteneciendo el autor de este documento al departamento de I+D+i, participando en el desarrollo de una nueva versión de este basada en *MDD*.

La finalidad de este TFG es desarrollar un algoritmo de mezcla de modelos generados con las *DSL Tools* de Microsoft en formato XML para una aplicación con arquitectura de microservicios y que basa su desarrollo en generación automática de código a partir de modelos con las anteriormente mencionadas *DSL Tools*.

Este algoritmo es el encargado de mezclar de manera inteligente los modelos que hayan recibido cambios en las versiones del producto de los clientes, con la versión del producto estándar, para así poder incluir sus cambios deseados mientras se tienen los nuevos facilitados por el desarrollo estándar de la aplicación. Así se permitirá la personalización del *software* por parte de los clientes a la vez que mantienen su versión del producto actualizada.

Además, se implementan en las *DSL Tools* protecciones de modelos, para así evitar que en los cambios añadidos para las versiones del cliente no se alteren elementos procedentes del producto estándar de formas indebidas, limitando a la vez los cambios que se pueden añadir a la versión del cliente, todo en función de la especificación establecida.

En el desarrollo del algoritmo se ha empleado como tecnología principal .NET Core, con los aspectos que nos ofrece C#, el lenguaje de programación utilizado, y se ha hecho especial uso de las bibliotecas básicas de tratado de XML facilitadas por C#. Para las protecciones de los modelos se han modificado las *DSL Tools* creadas por el propio equipo de I+D+i.

**Palabras clave:** MDD, Mezcla de Modelos, DSL Tools, Generación automática de código, Microservicios, Protección de modelos, .NET, C#, XML

---

## Abstract

The present work has been carried out in the context of internships in a company, whose main product is an ERP for the social-health sector, the author of this document belonging to the R+D+i department, helping in the development of a new version of this based on *MDD*.

The purpose of this FDP is to develop an algorithm for mixing models generated with Microsoft's *DSL Tools* in XML format for an application with microservices architecture and that bases its development on automatic code generation from models with the aforementioned *DSL Tools*.

This algorithm will be in charge of intelligently mixing the models that have received changes in the product versions of the clients, with the version of the standard product, in order to be able to include their desired changes while having the new ones provided by the standard development of the application. This will allow customers to customize the *software* while keeping their version of the product up to date.

In addition, model protections will be implemented in the *DSL Tools*, in order to prevent changes added for client versions from altering elements from the standard product in undue ways, while limiting the changes that can be made. add to the client version, all based on the stated specification.

In the development of the algorithm, .NET Core has been used as the main technology, with the aspects offered by C#, the programming language used, and special use has been made of the basic XML treatise libraries provided by C . For the protection of the models, the *DSL Tools* created by the R+D+i team itself have been modified.

**Key words:** MDD, Models mixing, DSL Tools, Automated code generation, Microservices, Models protection, .NET, C#, XML

---



# Índice general

---

<b>Índice general</b>	<b>VII</b>
<b>Índice de figuras</b>	<b>IX</b>
<b>Índice de tablas</b>	<b>XII</b>
<hr/>	
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación	1
1.2 Objetivos	3
1.3 Estructura de la memoria	3
<b>2 Estado del arte</b>	<b>5</b>
2.1 ERP existentes en el mercado	5
2.1.1 SAP	5
2.1.2 Oracle ERP	6
2.1.3 Microsoft Dynamics 365	8
2.2 ERP en el sector socio-sanitario	9
2.2.1 Ekon Health	9
2.2.2 Izaro ERP	10
2.3 Mezcla de modelos	11
2.3.1 Git	11
2.3.2 Mezcla de XML	12
2.4 Conclusiones acerca del estado del arte	12
<b>3 Contexto tecnológico</b>	<b>13</b>
3.1 Tecnologías usadas	13
3.2 System.Xml	13
<b>4 Ejemplo de generación de código con DSL Tools</b>	<b>15</b>
4.1 Instalación de las DSL Tools	15
4.2 Creación del proyecto	15
4.3 Elementos más importantes en el desarrollo del DSL	18
4.4 Desarrollo del DSL	19
4.5 Modelado y generación de código	19
4.6 Conclusiones	21
<b>5 Desarrollo</b>	<b>23</b>
5.1 Glosario	23
5.2 Especificación de requisitos	23
5.2.1 Casos de uso	23
5.2.2 Requisitos no funcionales	25
5.3 Diseño	28
5.3.1 Diseño del algoritmo para mezclar modelos	29
5.3.2 Diseño de las modificaciones de las DSL Tools y las protecciones de los modelos	30
5.4 Programación	35
5.4.1 Programación del algoritmo para mezclar modelos	35

5.4.2	Programación de las modificaciones de las <i>DSL Tools</i> y las protecciones de los modelos . . . . .	37
5.5	Pruebas . . . . .	42
5.5.1	Pruebas del algoritmo . . . . .	42
5.5.2	Pruebas de las modificaciones de las <i>DSL Tools</i> y las protecciones de los modelos . . . . .	49
5.6	Metodología . . . . .	51
<b>6</b>	<b>Caso de estudio</b>	<b>55</b>
6.0.1	Explicación del caso . . . . .	55
6.0.2	Versión estándar desplegada al cliente . . . . .	55
6.0.3	Evolución de la versión estándar . . . . .	56
6.1	Evolución de la versión del cliente . . . . .	57
6.2	Resultado de la aplicación del algoritmo de mezcla de modelos . . . . .	58
<b>7</b>	<b>Conclusiones y trabajo futuro</b>	<b>61</b>
	<b>Referencias</b>	<b>63</b>
<hr/>		
	Apéndices	
<b>A</b>	<b>Acrónimos empleados</b>	<b>65</b>
<b>B</b>	<b>Tablas de casuística</b>	<b>67</b>
B.1	Dominio . . . . .	67
B.1.1	<i>Domain Classes</i> . . . . .	67
B.1.2	<i>Domain Relationships</i> . . . . .	73
B.2	Aplicación . . . . .	78
B.2.1	<i>Domain Classes</i> . . . . .	78
B.2.2	<i>Domain Relationships</i> . . . . .	81
B.3	UI . . . . .	89
B.3.1	<i>Domain Classes</i> . . . . .	89
B.3.2	<i>Domain Relationships</i> . . . . .	97
<b>C</b>	<b>Objetivos de desarrollo sostenible</b>	<b>113</b>



# Índice de figuras

---

4.1	Creación de una solución de diseñador de lenguajes específicos de dominio en <i>Visual Studio</i> . . . . .	16
4.2	Selección de plantilla para proyecto de diseñador de lenguaje específico de dominio en <i>Visual Studio</i> . . . . .	16
4.3	Creación de extensión de archivo para proyecto de diseñador de lenguaje específico de dominio en <i>Visual Studio</i> . . . . .	17
4.4	Establecimiento de detalles para proyecto de diseñador de lenguaje específico de dominio en <i>Visual Studio</i> . . . . .	17
4.5	Elección de un método de firma para proyecto de diseñador de lenguaje específico de dominio en <i>Visual Studio</i> . . . . .	17
4.6	Metamodelo de un proyecto de diseñador de lenguaje específico de dominio en <i>Visual Studio</i> . . . . .	18
4.7	Explorador de DSL en <i>Visual Studio</i> . . . . .	18
4.8	Detalles de DSL y ventana de propiedades en <i>Visual Studio</i> . . . . .	18
4.9	Metamodelo de DSL de entidades y atributos. . . . .	19
4.10	Cuadro de herramientas de la <i>DSL Tool</i> . . . . .	20
4.11	Modelo de la <i>DSL Tool</i> creada sobre personas . . . . .	20
4.12	Ejemplo de código generado con el DSL creado. . . . .	21
5.1	Diagrama de casos de uso. . . . .	24
5.2	Modelo de calidad ISO/IEC 25010. . . . .	25
5.3	Estructura de la solución del algoritmo . . . . .	29
5.4	Tabla de requerimientos de la clase <i>Entity</i> . . . . .	30
5.5	Primera tabla de requerimientos de la relación <i>Generalization</i> . . . . .	31
5.6	Segunda tabla de requerimientos de la relación <i>Generalization</i> . . . . .	31
5.7	Tabla de requerimientos de la clase <i>DefaultDTO</i> . . . . .	32
5.8	Tabla de requerimientos de la relación <i>AdHocActionReferencesInputOutputDTO</i> . . . . .	32
5.9	Tabla de requerimientos de la clase <i>ExternalFormReference</i> . . . . .	33
5.10	Tabla de requerimientos de la relación <i>ExternalCommandReferenceReferencesForm</i> . . . . .	34
5.11	Diagrama de flujo del algoritmo de mezcla de modelos . . . . .	35
5.12	Cabecera del método <i>MixModels</i> . . . . .	36
5.13	Bucle de carga de los modelos XML . . . . .	36
5.14	Ejemplo de método recursivo . . . . .	36
5.15	Escritura de los modelos XML . . . . .	37
5.16	Elemento base de la jerarquía . . . . .	37
5.17	Como establecer método <i>CanMerge</i> manual . . . . .	38
5.18	Método <i>CanMerge</i> de <i>Entity</i> . . . . .	38
5.19	Como establecer <i>TypeDescriptor</i> personalizado. . . . .	38
5.20	Método <i>IsPropertyDescriptorReadOnly</i> . . . . .	38
5.21	Editor <i>ModelElementReferenceWrapperEditor</i> . . . . .	39
5.22	Validación sobre <i>Entity</i> . . . . .	39

5.23	Interfaz común a las relaciones. . . . .	40
5.24	Plantillas de generación de código relacionadas con las relaciones. . . . .	40
5.25	Como establecer <i>CanAccept</i> personalizado para las relaciones. . . . .	40
5.26	Método <i>CanAccept</i> personalizado en <i>ConnectionBuilder</i> . . . . .	40
5.27	Método auxiliar auto generado de clase <i>Helper</i> . . . . .	41
5.28	Método de clase parcial con la lógica para permitir o no el agregado de la relación. . . . .	41
5.29	Como establecer serializado personalizado de las relaciones. . . . .	41
5.30	Lista de tests del algoritmo de mezcla de modelos. . . . .	42
5.31	Lista de tests de dominio del algoritmo de mezcla de modelos. . . . .	43
5.32	Lista de tests de aplicación del algoritmo de mezcla de modelos. . . . .	43
5.33	Lista de tests de UI del algoritmo de mezcla de modelos. . . . .	44
5.34	Lista de modelos para los tests del algoritmo de mezcla de modelos. . . . .	44
5.35	Modelo de dominio de test del algoritmo de mezcla de modelos. . . . .	45
5.36	Modelo de aplicación de la versión de estándar de test del algoritmo de mezcla de modelos. . . . .	45
5.37	Modelo de UI de la versión de estándar de test del algoritmo de mezcla de modelos. . . . .	46
5.38	Modelo de aplicación de la versión de cliente de test del algoritmo de mezcla de modelos. . . . .	46
5.39	Modelo de UI de la versión de cliente de test del algoritmo de mezcla de modelos. . . . .	46
5.40	Modelo de aplicación esperado como resultado de test del algoritmo de mezcla de modelos. . . . .	47
5.41	Modelo de UI esperado como resultado de test del algoritmo de mezcla de modelos. . . . .	47
5.42	Test del algoritmo de mezcla de modelos. . . . .	48
5.43	Ejemplo de tabla de restricciones. . . . .	50
5.44	Test de restricción. . . . .	50
5.45	Línea temporal del proyecto. . . . .	52
6.1	Modelo de dominio de la versión estándar desplegada al cliente. . . . .	55
6.2	Modelo de aplicación de la versión estándar desplegada al cliente. . . . .	56
6.3	Modelo de UI de la versión estándar desplegada al cliente. . . . .	56
6.4	Modelo de dominio de la versión estándar evolucionada. . . . .	57
6.5	Modelo de UI de la versión estándar evolucionada. . . . .	57
6.6	Modelo de dominio de la versión de cliente evolucionada. . . . .	57
6.7	Modelo de UI de la versión de cliente evolucionada. . . . .	58
6.8	Modelo de dominio de la versión final. . . . .	58
6.9	Modelo de UI de la versión final. . . . .	59
B.1	Tabla de la clase <i>Entity</i> . . . . .	67
B.2	Tabla de la clase <i>Enumeration</i> . . . . .	67
B.3	Tabla de la clase <i>Field</i> . . . . .	68
B.4	Tabla de la clase <i>EnumerationValue</i> . . . . .	68
B.5	Tabla de la clase <i>Validation</i> . . . . .	68
B.6	Tabla de la clase <i>PostAction</i> . . . . .	69
B.7	Tabla de la clase <i>CalculatedPropertyTestCaseCollection</i> . . . . .	69
B.8	Tabla de la clase <i>CalculatedPropertyTestCase</i> . . . . .	69
B.9	Tabla de la clase <i>ValidationTestCaseCollection</i> . . . . .	70
B.10	Tabla de la clase <i>ValidationTestCase</i> . . . . .	70
B.11	Tabla de la clase <i>PostActionTestCaseCollection</i> . . . . .	70
B.12	Tabla de la clase <i>PostActionTestCase</i> . . . . .	71

B.13	Tabla de la clase <i>TestCaseEntityDatabase</i>	71
B.14	Tabla de la clase <i>TestCaseEntityEntryParameter</i>	71
B.15	Tabla de la clase <i>ExternalEntityReference</i>	72
B.16	Tabla de la clase <i>ExternalEnumerationReference</i>	72
B.17	Primera tabla de la relación <i>Association</i>	73
B.18	Segunda tabla de la relación <i>Association</i>	74
B.19	Primera tabla de la relación <i>Generalization</i>	75
B.20	Segunda tabla de la relación <i>Generalization</i>	75
B.21	Primera tabla de la relación <i>EntityReferencesEnumerationBases</i>	76
B.22	Segunda tabla de la relación <i>EntityReferencesEnumerationBases</i>	77
B.23	Tabla de la clase <i>DefaultDTO</i>	78
B.24	Tabla de la clase <i>SimpleDTO</i>	78
B.25	Tabla de la clase <i>ExternalDtoBase</i>	78
B.26	Tabla de la clase <i>ParameterDTO</i>	79
B.27	Tabla de la clase <i>ExternalParameterDtoReference</i>	79
B.28	Tabla de la clase <i>PrivateAdHocAction</i>	79
B.29	Tabla de la clase <i>PrivateAdHocActionContainer</i>	80
B.30	Tabla de la clase <i>PublicAdHocAction</i>	80
B.31	Tabla de las relaciones <i>AdHocActionReferencesInputParameterDTO</i> , <i>AdHocActionReferencesOutputParameterDTO</i> , <i>AdHocActionReferencesInputDtoBase</i> y <i>AdHocActionReferencesOutputDtoBase</i>	81
B.32	Tabla de las relaciones <i>AdHocActionReferencesOutputExternalDtoBaseReference</i> y <i>AdHocActionReferencesInputExternalDtoBaseReference</i>	82
B.33	Tabla de las relaciones <i>AdHocActionReferencesOutputExternalParameterDTO</i> y <i>AdHocActionReferencesInputExternalParameterDTO</i>	83
B.34	Tabla de la relación <i>HierarchicalDtoReferencesDTO</i>	83
B.35	Tabla de la relación <i>MessageProcessorProcessesParameterDTO</i>	84
B.36	Primera tabla de la relación <i>ReportHasCustomLabelGroupBases</i>	84
B.37	Segunda tabla de la relación <i>ReportHasCustomLabelGroupBases2</i>	85
B.38	Tabla de la relación <i>ReportHasInputExternalDataDtoBaseReference</i>	86
B.39	Tabla de la relación <i>ReportHasInputExternalParameterDTO</i>	87
B.40	Tabla de la relación <i>ReportHasInputParameterDTO</i>	87
B.41	Primera tabla de la relación <i>SubReport</i>	88
B.42	Segunda tabla de la relación <i>SubReport2</i>	88
B.43	Tercera tabla de la relación <i>SubReport3</i>	89
B.44	Tabla de la clase <i>Command</i>	89
B.45	Tabla de la clase <i>CommandGroup</i>	90
B.46	Tabla de la clase <i>CrudCommandGroup</i>	90
B.47	Tabla de la clase <i>CustomControl</i>	90
B.48	Tabla de la clase <i>Dashboard</i>	91
B.49	Tabla de la clase <i>DashboardContainer</i>	91
B.50	Tabla de la clase <i>ExternalCommandReference</i>	91
B.51	Tabla de la clase <i>ExternalFormContainerReference</i>	92
B.52	Tabla de la clase <i>ExternalFormReference</i>	92
B.53	Tabla de la clase <i>ExternalIndicator</i>	93
B.54	Tabla de la clase <i>Field</i>	93
B.55	Tabla de la clase <i>Form</i>	93
B.56	Tabla de la clase <i>FormContainer</i>	94
B.57	Tabla de la clase <i>Layout</i>	94
B.58	Tabla de la clase <i>Navigation</i>	94
B.59	Tabla de la clase <i>NavigationOption</i>	95
B.60	Tabla de la clase <i>Selector</i>	95

B.61	Tabla de la clase <i>Table</i>	95
B.62	Tabla de la clase <i>Template</i>	96
B.63	Tabla de la clase <i>Workspace</i>	96
B.64	Tabla de la clase <i>WorkspaceArea</i>	96
B.65	Tabla de la relación <i>CommandReferencesExternalFormContainerReference</i>	97
B.66	Tabla de la relación <i>CommandReferencesExternalFormReference</i>	98
B.67	Tabla de la relación <i>CommandReferencesTargetForm</i>	99
B.68	Tabla de la relación <i>CommandReferencesTargetFormContainer</i>	99
B.69	Tabla de la relación <i>CrudCommandGroupReferencesExternalFormReference</i>	100
B.70	Tabla de la relación <i>CrudCommandGroupReferencesTargetForm</i>	101
B.71	Tabla de la relación <i>DashboardContainerIncludesDashboards</i>	101
B.72	Tabla de la relación <i>DashboardIncludesExternalIndicator</i>	102
B.73	Tabla de la relación <i>ExternalCommandReferenceReferencesForm</i>	103
B.74	Tabla de la relación <i>FormContainerReferencesExternalFormReference</i>	104
B.75	Tabla de la relación <i>FormContainerReferencesForm</i>	105
B.76	Tabla de la relación <i>NavigationOptionReferencesExternalFormReference</i>	106
B.77	Tabla de la relación <i>NavigationOptionReferencesTargetForm</i>	107
B.78	Tabla de la relación <i>NavigationOptionReferencesTemplate</i>	108
B.79	Tabla de la relación <i>NavigationOptionReferencesWorkspace</i>	109
B.80	Tabla de la relación <i>NavigationReferencesFormContainer</i>	110
B.81	Tabla de la relación <i>TableReferencesFormContainer</i>	110
B.82	Tabla de la relación <i>TableReferencesTargetForm</i>	111
B.83	Tabla de la relación <i>WorkspaceAreaIncludesDashboardContainers</i>	111
B.84	Tabla de la relación <i>WorkspaceAreaIncludesUIContainerElements</i>	112

## Índice de tablas

---

5.1	Caso de uso modificar modelo	24
5.2	Caso de uso modificar modelo de cliente	24
5.3	Caso de uso actualizar versión del cliente mezclando los modelos	25
5.4	Requisito no funcional completitud funcional respecto a las <i>DSL Tools</i>	25
5.5	Requisito no funcional velocidad de ejecución del algoritmo	26
5.6	Requisito no funcional velocidad de desempeño de las <i>DSL Tools</i>	26
5.7	Requisito no funcional robustez de las protecciones de los modelos	26
5.8	Requisito no funcional tecnología empleada	27
5.9	Requisito no funcional el algoritmo debe ser genérico	27
5.10	Requisito no funcional capacidad del algoritmo de ser modificado	27
5.11	Requisito no funcional capacidad del algoritmo de ser probado	27
5.12	Requisito no funcional reusabilidad del código de las <i>DSL Tools</i>	28
5.13	Requisito no funcional capacidad de ser modificado del código de las <i>DSL Tools</i>	28
5.14	Requisito no funcional capacidad de ser probado del código de las <i>DSL Tools</i>	28

---

---

# CAPÍTULO 1

## Introducción

---

### 1.1 Motivación

---

El mundo está en constante informatización, por lo que cada vez se requieren más aplicaciones *software*, lo cual conlleva una mayor exigencia para la industria *software*. Esto lleva a buscar soluciones que faciliten y agilicen el desarrollo. Una de las principales y más prometedoras soluciones es la generación automática de código. Y es que esta es una gran aliada en el desarrollo de *software*, pues ofrece grandes ventajas como pueden ser:

- **Abstracción.** Al poseer un mayor nivel de abstracción, la programación resulta de mayor entendimiento.
- **Estandarización.** Las herramientas de generación facilitan la estandarización del código fuente, la interfaz de usuario, las bases de datos, etc.
- **Cambios en el código.** Debido a cómo se realiza, el desarrollo con herramientas de generación automática de código consigue que resulte mucho más sencilla la modificación de este, pues se hace a través de los modelos.
- **Productividad.** Aumenta la productividad del desarrollo.
- **Énfasis en el análisis.** Debido a que el tiempo requerido en el desarrollo es menor, se permite emplear más recursos en la fase de análisis, lo cual ofrece claras ventajas.

La forma más usual de utilizar la generación automática de código es a partir de modelos, y aquí es donde entra el desarrollo de *software* guiado por modelos, conocido también por su acrónimo en inglés, MDD (*Model-Driven Development*) [1]. El MDD ofrece numerosas ventajas alineadas con la generación automática de código, entre ellas [2]:

- **Rapidez.** Al aumentar el nivel de abstracción, el desarrollo experimenta un aumento de la velocidad en su proceso, pues, un elemento de un modelo representa diversas líneas de código, lo que desemboca en poder crear un mayor número de funcionalidades en la misma cantidad de tiempo.
- **Economía.** Como el desarrollo es más veloz, el producto se saca a mercado en un lapso menor de tiempo, además, se experimenta un ahorro en personal, pues se requiere de menos especialistas. Aunque la mejora en la economía puede verse afectada por el coste de aprendizaje del MDD o del coste de adquisición de la herramienta empleada.

- **Menor propensión a errores.** El MDD asegura que el equipo de desarrollo se puede enfocar en el resto de las funcionalidades de la aplicación, pues los problemas técnicos son cubiertos por las herramientas. Además, al reducir el código escrito a mano, se reduce a su vez la probabilidad de introducir errores humanos en el código.
- **Menor sensibilidad.** Esto incluye diversos aspectos, pues el MDD produce una menor sensibilidad a los cambios en los requerimientos de negocio, gracias al resto de ventajas ofrecidas, una debilidad en menor cantidad a los cambios en la tecnología, pues los modelos no tienen que ser cambiados, sino el generador de código, y también una menor sensibilidad a los cambios en el personal, gracias de nuevo a la elevación del nivel de abstracción.

El DSD, o *Domain-Specific Development*[3] se basa en la observación de que muchos de los problemas del desarrollo de *software* se pueden solventar fácilmente diseñando un lenguaje para un propósito en especial. El método habitual en un contexto DSD de enfrentar estos problemas suele ser la utilización de los mencionados DSL o *Domain Specific Language*.

Pensemos por ejemplo, en el problema de encontrar cada ocurrencia de un patrón en específico en un conjunto de caracteres de un fichero, y una acción a realizar sobre cada una de estas ocurrencias, es un trabajo ciertamente complejo y tedioso. Para ello existen los motores de expresiones regulares, como por ejemplo la clase de .NET <sup>1</sup> *System.Text.RegularExpressions.Regex* [4], que facilita enormemente este tipo de tareas. Sin estas, el programador tendría que desarrollar un programa especial para reconocer patrones y tratarlos de la forma deseada, lo cual es una tarea más ardua y propensa a errores.

El DSD aplica este mismo principio a una amplia variedad de problemas, delegando en la capacidad imaginativa y de abstracción del equipo de desarrollo.

Microsoft ofrece unas herramientas de desarrollo de DSL ligadas a su IDE, Visual Studio <sup>2</sup>, las llamadas *DSL Tools* <sup>3</sup>. Estas son empleadas en el desarrollo del proyecto en el que este trabajo se ve envuelto, son engranaje principal de todo el desarrollo, ya que a partir de estas, el equipo diseña y programa diversos DSL, cada uno con su propio propósito. Luego, con las *DSL Tools* desarrolladas, se modelan los distintos microservicios[5] que componen la aplicación, en los cuales se basa la arquitectura del producto.

Los microservicios son un estilo de arquitectura que desarrolla una única aplicación como el centro de pequeños servicios, los cuales se ejecutan en sus propios procesos, con mecanismos propios y normalmente, su propia API. Estos servicios se especializan en atender cierto conjunto de necesidades del negocio y se puedan desplegar de forma automática e independiente.

Si bien se mencionaba la creciente exigencia en el desarrollo del *software*, existe otra exigencia en aumento, y es su personalización[6]. El cliente desea adquirir un producto que pueda adecuar en la medida de lo posible a sus gustos, preferencias y necesidades.

Por ejemplo, un doctor que está utilizando la aplicación desea observar en la ficha de un paciente, la última presión arterial medida, sin embargo, el programa no la muestra por defecto. Este problema se soluciona fácilmente permitiendo la personalización del *software*

La aplicación posee mecanismos de personalización para el usuario. Y aquí es donde convergen las dos principales motivaciones de este proyecto, la generación automática

<sup>1</sup> Documentación oficial de .NET: <https://docs.microsoft.com/es-es/dotnet/>

<sup>2</sup> Página oficial de Visual Studio de Microsoft: <https://visualstudio.microsoft.com/es/>

<sup>3</sup> Introducción a las DSL Tools de Visual Studio: <https://docs.microsoft.com/es-es/visualstudio/modeling/getting-started-with-domain-specific-languages?view=vs-2022>

de código mediante modelos y el deseo del usuario por personalizar su versión de la aplicación.

El hecho de que la aplicación sea configurable por el usuario, genera el requisito de que la aplicación en desarrollo ofrezca también este servicio.

Por lo anterior, en este TFG se plantea el desarrollo de un algoritmo que mezcle modelos. De este modo, la personalización por parte del cliente se realizará modificando estos, y que además, obtenga las actualizaciones aplicadas en el producto estándar, empleando este algoritmo.

Además, el algoritmo debe considerar que en un futuro el propio cliente podría realizar estas modificaciones con una herramienta de alto nivel y amigable. Sin embargo, por el momento, los cambios deberán ser solicitados al equipo de desarrollo, los cuales modificarán directamente los modelos. Por este motivo, se establecen restricciones para proteger los modelos, lo que conlleva a que no se realicen cambios indeseados en el producto.

## 1.2 Objetivos

---

El trabajo tiene dos objetivos principales:

- **Desarrollar un algoritmo para mezclar modelos** que permita modificar el producto del cliente según los cambios deseados además de obtener las nuevas actualizaciones del desarrollo estándar.
- **Implementar protecciones de los modelos** para impedir modificaciones no deseadas.

Además, se hará especial hincapié en la realización de pruebas que otorguen una buena cobertura de código en el algoritmo de mezcla de modelos y las protecciones de modelos mencionadas.

## 1.3 Estructura de la memoria

---

El documento se establece acorde a 5 capítulos:

- El primero es el actual, el cual introduce el trabajo, nos indica las motivaciones del mismo, introduce los objetivos a cumplir y finaliza con esta explicación de la estructura en la que se establece la memoria.
- El segundo capítulo es el estado del arte, en el cual se explica como se encuentra el terreno en cuanto al problema a abarcar, posibles soluciones y alternativas comparadas.
- El tercer bloque tiene como objetivo introducirnos en el contexto tecnológico del problema, y las herramientas aplicadas.
- El cuarto capítulo se compone de un pequeño tutorial abarcando todo el proceso de desarrollo de una *DSL Tool* de forma sencilla.
- El capítulo quinto es el que más se extiende, ya que abarca el proceso de desarrollo de la solución, el cual incluye, claro, la especificación de requisitos, el diseño de la

propia solución, la programación como tal, las pruebas realizadas y la metodología empleada.

- El sexto capítulo presenta una demo del algoritmo de mezcla de modelos con un caso que simula la realidad.
- El capítulo que abarca el último lugar incluye unas conclusiones acerca del trabajo realizado, expone si se han cumplido los objetivos de forma exitosa y añade una visión de como se podrá expandir el proyecto realizado, y futuras posibles mejoras del programa relacionadas con lo desarrollado.

Además, se incluye una sección que recopila las referencias utilizadas a lo largo del trabajo, además de un apéndice con los acrónimos utilizados, otro con documentación del proyecto y un último apéndice con objetivos de desarrollo sostenible.



---

---

## CAPÍTULO 2

# Estado del arte

---

Como se ha mencionado, el producto en el que se encuentra comprendido el trabajo es un ERP, más concretamente del sector socio-sanitario, por lo cual se describirán los ERP más reconocidos de forma general, y a continuación se tendrán en cuenta algunos competidores más directos.

Además, más tarde se comentarán posibles soluciones existentes al problema que aborda el trabajo actual que se podrían haber tenido en cuenta, pero no han resultado aplicables.

### 2.1 ERP existentes en el mercado

---

Comentaremos ahora algunos de los sistemas de planificación de recursos empresariales más reconocidos del mercado, y cómo estos abarcan problemáticas similares a la de nuestro contexto.

#### 2.1.1. SAP

*"SAP es uno de los principales productores mundiales de software para gestión de procesos de negocio, y desarrolla soluciones que facilitan el procesamiento eficaz de datos y el flujo de información entre las organizaciones."* <sup>1</sup>

Con fundación en 1972, su nombre original denominaba al desarrollo de programas de sistemas de análisis, y se acabó denominando simplemente SAP <sup>2</sup>. Se expandió desde ser un pequeño emprendimiento hasta contar con miles de empleados a lo largo del mundo.

Con su *software* SAP R/2 y SAP R/3, la empresa estableció prácticamente un estándar global para los ERP. En la actualidad, la versión del producto vigente, S/4HANA incluye la computación *in-memory*, y soporte de tecnologías avanzadas como inteligencia artificial y *machine learning*.

Hoy en día, las distintas aplicaciones integradas forman parte de una suite en lugar de la plataforma heredada impulsada por procesos.

---

<sup>1</sup>Página oficial de presentación de SAP <https://www.sap.com/spain/about/company/what-is-sap.html>

<sup>2</sup>Página principal de SAP España <https://www.sap.com/spain/index.html>

*SAP* además posee una gama de áreas:

- ERP y finanzas.
- CRM y experiencia del cliente.
- Gestión de red y gastos.
- Cadena de suministro digital.
- RR.HH. e interacción con el personal.
- Gestión de experiencias.
- Business Technology Platform.
- Transformación digital.
- Pequeñas y medianas empresas.
- Soluciones para la industria.

Siendo que *SAP* es un sistema empleado por un elevado número de empresas internacionalmente, requiere de alta flexibilidad y adaptabilidad, por lo que era de esperar que incluyera opciones de personalización de *software*.

Si bien se puede encontrar en internet, que existía un sistema de personalización para las antiguas versiones del producto, denominado *SPRO*, no hay mucha información al respecto. En contra parte, para la personalización del producto actual, *SAP S/4HANA* se puede hallar algo más de contenido [7].

El producto ofrece herramientas para la adaptación de la interfaz de usuario, añadir formularios, reportes, campos personalizados e incluso lógica.

### 2.1.2. Oracle ERP

*"Los líderes del futuro serán los que mejor se adapten al cambio en el presente. Oracle Fusion Cloud ERP es un conjunto de soluciones de ERP en la nube completo y moderno que proporciona a sus equipos capacidades avanzadas, como la IA para automatizar los procesos manuales que los ralentizan, los análisis para reaccionar a los cambios del mercado en tiempo real y las actualizaciones automáticas necesarias para mantenerse actualizados y ganar una ventaja competitiva."*<sup>3</sup>

*Oracle* es una compañía estadounidense fundada en el año 1977, comienzan siendo una consultoría de *software* que en sus primeros años se dedicó principalmente al apartado de las bases de datos. Como es sabido, *Oracle* tiene importantes productos de *software* aparte de *Oracle ERP*, como lo son *MySQL* <sup>4</sup> o el lenguaje de programación orientado a objetos *Java* <sup>5</sup>.

Finalmente, en 1989 sacan su propio ERP al mercado, conocido como *Oracle Financials*. A su producto actual se le conoce como *Oracle Cloud ERP*, que nos ofrece también una amplia gama de áreas como pueden ser:

- Finanzas.
- Gestión de proyectos.

<sup>3</sup>Página principal de Oracle ERP España <https://www.oracle.com/es/erp/>

<sup>4</sup>Página principal de MySQL España <https://www.oracle.com/es/mysql/>

<sup>5</sup>Página principal de Java España <https://www.oracle.com/es/java/>

- Compras.
- Gestión de riesgos y cumplimiento.
- Gestión del rendimiento empresarial.
- Cadena de suministro y fabricación.
- Analítica ERP.

Al igual que *SAP*, el ERP de *Oracle* es usado por muchas empresas, por lo cual debe incluir múltiples opciones de personalización para incrementar la flexibilidad y permitir al cliente tener la mejor versión según su punto de vista del producto.

Se puede fácilmente encontrar documentación oficial acerca de este apartado [8]. Cabe mencionar a *Oracle Fusión*, se trata de un *middleware* que anexiona la familia de aplicaciones empresariales de *Oracle*, el sistema de personalización de *Oracle* gira en torno este *middleware*.

Existen personalizaciones en tiempo de ejecución que se guardan en un *sandbox*, que se trata de un entorno de *testing* que aísla los cambios para que puedan ser validados antes de ser incluidos al entorno común. Los diálogos de administración de personalizaciones permiten identificar donde y para qué capa se han realizado estas, y también es posible traer al producto personalizaciones que otros hayan realizado o exportar las propias.

Las aplicaciones que abarca *Oracle Fusion* están construidas usando un modelo de datos común, gracias a esto es que se puede añadir un cambio a un objeto existente y que este esté disponible para otros objetos distintos.

Existen tres escenarios en lo que a personalización y extensibilidad [9] se refiere [10]:

- **Personalización:** referente a los cambios que un usuario final del producto puede realizar a distintos artefactos en la UI en tiempo de ejecución, los cuales se recuperan para este mismo usuario cada vez que inicia en la aplicación.
- **Extensiones en tiempo de ejecución:** abarcan las personalizaciones y extensiones que un analista del negocio puede realizar a la aplicación usando compositores basados en explorados. Estos pueden ser usados por todo el mundo o por un subconjunto de usuarios. Pueden ir desde cambiar la visualización y experiencia de una página, hasta alterar los objetos del dominio, elementos asociados, *workflows*...
- **Extensiones en tiempo de diseño:** estas incluyen cambios más complicados como por ejemplo crear una aplicación basada en SOA o trabajo por lotes. Además, estos requieren de despliegue al entorno en tiempo de ejecución. Los cambios mencionados son normalmente realizados por desarrolladores de Java utilizando *Oracle JDeveloper*<sup>6</sup>.

Cabe mencionar que las personalizaciones de aplicaciones de *Oracle Fusion* son realizadas por capas, y orientadas a distintos roles de acceso y autorización de cambios.

---

<sup>6</sup>Página principal de *JDeveloper* <https://www.oracle.com/application-development/technologies/jdeveloper.html>

### 2.1.3. *Microsoft Dynamics 365*

"Adáptese e innove con una empresa hiperconectada: ofrezca a todos la información y la libertad que necesitan para prosperar, conectando sus datos, procesos y equipos con aplicaciones empresariales inteligentes."<sup>7</sup>

*Microsoft* también posee entre sus filas un ERP. *Microsoft* es una empresa estadounidense cofundada por el reconocido Bill Gates en el año 1975. Aunque no fue hasta 2006 que la empresa decidió sacar su propio *software* de planificación de recursos empresariales, entrando así en el mundo que se está exponiendo.

Al igual que los otros productos que hemos mencionado, *Microsoft Dynamics 365* también ofrece un repertorio de áreas a disposición del cliente:

- Ventas.
- Marketing.
- Servicio.
- Finanzas.
- Comercio.
- Cadena de suministro

*Microsoft Dynamics 365* es una suite en la nube que agrupa las mejores funcionalidades de Ventas (CRM) y de planificación (ERP) de *Microsoft Dynamics* junto con herramientas ofimáticas y colaborativas, además de Business Intelligence (BI).<sup>8</sup>

En su solución, este producto incluye, principalmente:

- Su ERP *Dynamics Business Central* o *D365 for Finance an Operations*.
- *Microsoft Dynamics for Sales* para la gestión de ventas y marketing.
- *Office 365* conocido por la inmensa mayoría por ser la solución de ofimática más popular del mercado.
- *Power BI*, avanzada herramienta de Business Intelligence.
- *Cortana Intelligence* para el apoyo con inteligencia predictiva y análisis.
- *Azure* aportando datos sobre IOT.
- Conectores con aplicaciones y servicios externos.

El producto en cuestión ofrece también mecanismos de personalización. Es más, *Microsoft Dynamics 365* ofrece personalización para el usuario sin necesidad de escribir código, diseñadores y editores WYSIWYG, lo que hacen esta tarea más accesible y sencilla [11]. En la documentación se pueden encontrar diferentes categorías de componentes a usar para este desarrollo:

<sup>7</sup>Página principal de *Microsoft Dynamics 365* España <https://dynamics.microsoft.com/es-es/>

<sup>8</sup>¿Qué es *Microsoft Dynamics 365*? <https://www.quonext.com/microsoft-dynamics-365s>

- **Datos:** los componentes que determinan en qué datos se basará la aplicación.
- **UI:** los componentes que determinan cómo interactúan los usuarios con la aplicación.
- **Lógica:** los componentes que determinan los procesos de negocio, reglas y automatización que la aplicación tendrá.
- **Visualizaciones:** los componentes que determinan qué tipo de visualizaciones y creación de informes estarán a disposición.

*Microsoft* enuncia la capacidad de extensibilidad y portabilidad, además de la posibilidad de instalar soluciones de terceros para solucionar incapacitaciones o limitaciones en el proceso de desarrollo.

## 2.2 ERP en el sector socio-sanitario

---

Una vez analizadas las alternativas generales y más conocidas que existen en el mercado al producto de la empresa donde se desarrolla este trabajo, es importante analizar algunas otras más especializadas en el sector socio-sanitario.

### 2.2.1. *Ekon Health*

*Ekon*<sup>9</sup> se autodeclara como *el fabricante de referencia en tecnología cloud para la PYME española*<sup>10</sup>. *Ekon ERP* y *Cloud* surge en el año 2000. *Ekon Cloud ERP* se enfoca en el mercado de las pymes españolas, y es mencionado en este apartado pues, aunque se trata de un programa que abarca diversos sectores como son: industria, distribución, construcción o despachos profesionales, posee un producto especializado en salud.

*Ekon Health* es una solución para la gestión socio-sanitaria de residencias. En su página web [12] nos indican que su objetivo actual es reducir el impacto con la situación actual de pandemia del COVID-19, además del crecimiento del envejecimiento de la población, las cuales son oportunidades de negocio tanto para los productos de este apartado como para el de la empresa donde se desarrolla este trabajo.

*Ekon* hace un breve resumen en su página de lo que ofrece en su solución del sector socio-sanitario:

- **Funcionalidades de salud:** *Ekon Health* es un software de gestión HIS que facilita la gestión integral en el ámbito de la sanidad, ofreciendo soluciones para garantizar eficiencia y el mejor trato al paciente.
- **Información del paciente al instante:** La aplicación permite la fácil circulación de la información de cada paciente por todo el ecosistema, lo que facilita el trato con los pacientes gracias al PRM y mejora la calidad del servicio.
- **Visión global - Historial:** *Ekon Health* ofrece una visión global de las historias de salud, gestionando la información del sistema de manera integral, facilitando así la labor de los trabajadores.

---

<sup>9</sup>Página principal de *Ekon*: <https://www.ekon.es/recursos/ekon-health-gestion-sociosanitaria-y-residencias-con-ekon-health-erp/#>

<sup>10</sup>Página de presentación de *Ekon* <https://www.ekon.es/quienes-somos/>

- **HIS totalmente integrado:** *Ekon* comunica que hay pocas soluciones disponibles en el mercado que ofrezcan un HIS totalmente integrado, como es su producto *Health*.
- **Tu inversión no caduca:** *Ekon* enuncia que el cliente es capaz de recibir de forma rápida y fácil las últimas funcionalidades tecnológicas que surgen y son implementadas en la aplicación.
- **El menor coste de propiedad** Según la empresa, este producto es la solución del mercado que ofrece una funcionalidad versátil y adaptable al menor precio de propiedad garantizado.

*Ekon* también comenta algunas características generales de una aplicación como la digitalización, la movilidad, la integración, o la mejora continua.

*Ekon* también ofrece personalización, sin embargo, se puede observar un defecto en la manera que ellos ofrecen este servicio en su propia página. Ofrecen dos maneras de gestionar el negocio, estándar o personalizada, y el conflicto proviene de que según ellos mismos dicen, estas son excluyentes entre si. Por lo tanto, aquí vemos un problema que es la no posibilidad de desarrollar una personalización del producto mientras se disfruta de las ventajas del estándar, problemática cuya solución es el principal objetivo del trabajo del alumno.

Sin embargo, pese a las posibilidades de personalización del programa, es difícil encontrar información de cómo esta tarea es realizada.

### 2.2.2. *Izaro ERP*

*Zucchetti* es una empresa de *software* a medida y TIC, que posee entre su larga lista de soluciones *software* un ERP conocido como *Izaro*.

*Izaro* ofrece solución para diversos sectores como [13]:

- Fabricación discreta.
- Proyectos.
- Fabricantes de componentes de automoción.
- Distribución.
- Servicios.
- Sanidad.
- Alimentación.
- Gran consumo

Además, *Izaro ERP* ofrece diversas características técnicas y componentes como pudieran ser:

- **Grupos empresariales.** *Izaro* resuelve la gestión de grupos empresariales extendidos en el extranjero.
- **Accesible desde internet.**
- **Gestión de la calidad.** *Zucchetti* comenta que la gestión de la calidad es esencial en su producto *Izaro*, por lo cual se puede apreciar incluida en todo el proceso.

- **Todo bajo control.** *Izaro ERP* promete cubrir todo el ciclo de la gestión aportando una visibilidad completa.
- **Document Management System.** Gestión documental para eliminar los ficheros en físico de la empresa y digitalizar todos los informes.
- **Business Process Management.** Gestión de los procesos para garantizar su cumplimiento y control.
- **Busicess Intelligence.** Cuadros de mando que son posibles de navegar para tener conocimiento acerca de la empresa y el mercado.

Por desgracia, pese a ofrecer un rico conjunto de servicios y opciones, *Izaro ERP* no parece tener posibilidad de personalización para el cliente, por lo cual no es posible discutir acerca de este aspecto para el producto.

## 2.3 Mezcla de modelos

---

El siguiente apartado a tener en cuenta, tras haber comentado los principales productos competidores, son las posibilidades ya existentes para la mezcla de modelos, objetivo principal del trabajo presente.

### 2.3.1. *Git*

La primera opción cuando hablamos de mezcla de ficheros de cualquier tipo, es la herramienta *Git* <sup>11</sup>

*Git* es un sistema distribuido de control de versiones, gratis y de código abierto que soporta proyectos de gran magnitud [14]. *Git* es la principal opción a la hora de desarrollar un proyecto de *software* para tener un control de versiones, siendo prácticamente total su aplicación en los procesos de desarrollo existentes.

*Git* ofrece una serie de ventajas frente a los sistemas tradicionales [15]:

- **Sistema distribuido** sin punto central de fallo, que permite también el trabajo en local sin conexión.
- **Superrápido y ligero**, optimizado para poder realizar operaciones de control muy rápidas.
- **Sistema de ramas** para creación y mezclado de estas, poco propenso a problemáticas.
- **Integración de la información** gracias al modelo de almacenamiento que predice problemas de este tipo.
- **Flujos de trabajo muy flexibles.**
- **Staging.** Sistema que permite realizar selecciones sobre lo que se sube en el momento o no.
- Como ya se ha mencionado, es **gratis y de código abierto.**

---

<sup>11</sup>Página principal de *Git*: <https://git-scm.com/>

Sin embargo, pese a todas las capacidades que ofrece *Git* y lo eficaz que es, no permite un mezclado de modelos de las *DSL Tools* correcto y de la forma deseada. Esto es principalmente debido a que cuando trata de mezclar dos modelos, se generan numerosos conflictos, algunos incluso ilógicos, lo cual no es admisible para un proceso que se desea realizar de manera automática para permitir así el fácil despliegue del servicio a los clientes.

### 2.3.2. Mezcla de XML

Otra opción posible sería alguna forma de mezcla de ficheros XML de forma general, sin embargo, cuando se busca información acerca de esto, no se encuentra nada en específico, lo único son pequeños programas manuales para esta mezcla. Esto no es una opción factible, pues la mezcla de modelos deseada es bastante compleja.

## 2.4 Conclusiones acerca del estado del arte

---

Se ha estudiado la personalización que se ofrece en ERP de grandes empresas de desarrollo de *software*, y otros competidores especializados en el sector socio-sanitario. Si bien, la solución de este trabajo resulta más interesante que estas, son algo importante a tener en cuenta.

Se ha podido observar la diferencia entre las diferentes soluciones planteadas al problema de personalización de *software* ERP, o incluso la inexistencia de estos mecanismos, por lo cual podemos concluir que en la actualidad no existe una solución estandarizada y es un tema a medio resolver.

La solución planteada por la nueva versión del *software* de la empresa donde se realiza este trabajo, es muy innovadora, se basa en la mezcla de modelos, pues se trabaja en un marco MDD con DSL propios.

También se han estudiado las escasas alternativas que existen para la mezcla de los modelos de las *DSL Tools*, y como ya se dijo, el conocido sistema *Git* no es suficiente para esta ardua tarea, y no se han encontrado bibliotecas o herramientas suficientes para realizar el trabajo de forma general en cuanto a ficheros XML.



---

---

## CAPÍTULO 3

# Contexto tecnológico

---

En este capítulo se realizará una breve descripción de las tecnologías utilizadas para este trabajo. Se explicará más en profundidad la biblioteca *System.Xml*, utilizada a lo largo del desarrollo para el tratado de ficheros XML.

### 3.1 Tecnologías usadas

---

Para el desarrollo se han utilizado tecnologías de *Microsoft* principalmente, comenzando por el *framework .NET Core*<sup>1</sup>, heredero de *.NET Framework*<sup>2</sup>. El lenguaje de programación usado ha sido C#<sup>3</sup>. Para la generación automática de código mediante MDD se utilizan las *DSL Tools*, herramienta de creación de DSL para la creación de modelos y su comportamiento personalizado.

### 3.2 *System.Xml*

---

*System.Xml*<sup>4</sup> es la biblioteca principal de C# utilizada a lo largo del desarrollo del trabajo. *System.Xml* incluye una variedad de clases, métodos, interfaces... además de permitir estándares como XML 1.0<sup>5</sup>, espacios de nombres XML, esquemas XML, expresiones *XPath*<sup>6</sup> o transformaciones *XSLT*<sup>7</sup>.

La clase que se ha utilizado para tratar los modelos como un documento XML ha sido *XmlDocument*. Según la documentación oficial[17], esta *representa un documento XML. Puede usar esta clase para cargar, validar, editar, agregar y posicionar el código XML en un documento*". Un *XmlDocument* puede verse formado por un árbol de *XmlNode*.

Según la documentación, un *XmlNode* *representa un único nodo en el documento XML.*", los modelos generados por las *DSL Tools* no dejan de ser elementos contenidos dentro de otros sucesivamente, y todos englobados dentro de un elemento que representa el modelo.

---

<sup>1</sup>Página de introducción de *.NET Core* <https://docs.microsoft.com/es-es/dotnet/core/introduction>

<sup>2</sup>Página de introducción de *.NET Framework* <https://docs.microsoft.com/es-es/dotnet/framework/get-started/>

<sup>3</sup>Documentación oficial de C# <https://docs.microsoft.com/es-es/dotnet/csharp/>

<sup>4</sup>Documentación oficial de *System.Xml*: <https://docs.microsoft.com/es-es/dotnet/api/system.xml?view=net-6.0>

<sup>5</sup>Página oficial de W3C de XML 1.0: <https://www.w3.org/TR/xml/>

<sup>6</sup>Página principal de W3C sobre *XPath*: <https://www.w3.org/TR/xpath/>

<sup>7</sup>Página principal de W3C acerca de *XSLT*: <https://www.w3.org/TR/xslt/>

A lo largo del desarrollo se utilizará la idea conceptual de árbol al tratar los modelos como documentos XML.

Se ha empleado alguna clase más, por ejemplo *XmlAttribute*, la cual *representa un atributo. Los valores válidos y predeterminados del atributo se definen en una definición de tipo de documento (DTD) o un esquema.*". Esta ha sido empleada para representar atributos de los nodos del documento.

Como ya ha sido expuesto, la clase principal en el uso ha sido *XmlNode*, la cual tiene numerosas propiedades útiles para el tratado de modelos como puedan ser:

- *ChildNodes*, que obtiene los nodos hijos.
- *FirstChild*, que obtiene el primer hijo del nodo.
- *InnerXml* que muestra el contenido XML del interior del nodo.
- *OuterXml* que obtiene el mismo resultado que el atributo anterior, añadiendo las propiedades del propio nodo.

Cabe destacar también algunos métodos como *AppendChild* que añade un nuevo nodo al propio, *SelectNodes* que devuelve una lista de nodos que cumplen una expresión *XPath* especificada e incluso *InsertBefore* e *InsertAfter* para insertar un nodo antes o después correspondientemente, de otro nodo especificado.

---

---

## CAPÍTULO 4

# Ejemplo de generación de código con *DSL Tools*

---

En este capítulo se presenta un ejemplo de uso de las *DSL Tools*. Se generará código sencillo realizando todo el proceso, desde la creación de la herramienta y la plantilla hasta observar el código generado. Para este apartado se ha utilizado la guía oficial de introducción a las *DSL Tools* de *Microsoft*[18].

### 4.1 Instalación de las *DSL Tools*

---

Para poder desarrollar un proyecto con las *DSL Tools* de Microsoft primero necesitaremos algunos elementos instalados en la máquina donde desarrollar el proyecto, como son, el IDE de Microsoft Visual Studio<sup>1</sup>, el SDK de *Visual Studio*<sup>2</sup> y el SDK de modelado para *Visual Studio* que se puede descargar desde el instalador de Visual Studio.

### 4.2 Creación del proyecto

---

Como en cualquier proyecto con *Visual Studio*, el flujo estándar es crear una solución, cuando se crea un nuevo proyecto. En función de una plantilla escogida, se añadirán elementos necesarios para el tipo de proyecto seleccionado. En este caso, seleccionaremos la opción del diseñador de lenguajes específicos de dominio, o DSL, como se puede apreciar en la figura 4.1.

---

<sup>1</sup>Página de instalación de Visual Studio: <https://visualstudio.microsoft.com/es/downloads/>

<sup>2</sup>Página de guía de instalación oficial del SDK de *Visual Studio*: <https://docs.microsoft.com/es-es/visualstudio/extensibility/installing-the-visual-studio-sdk?view=vs-2022>

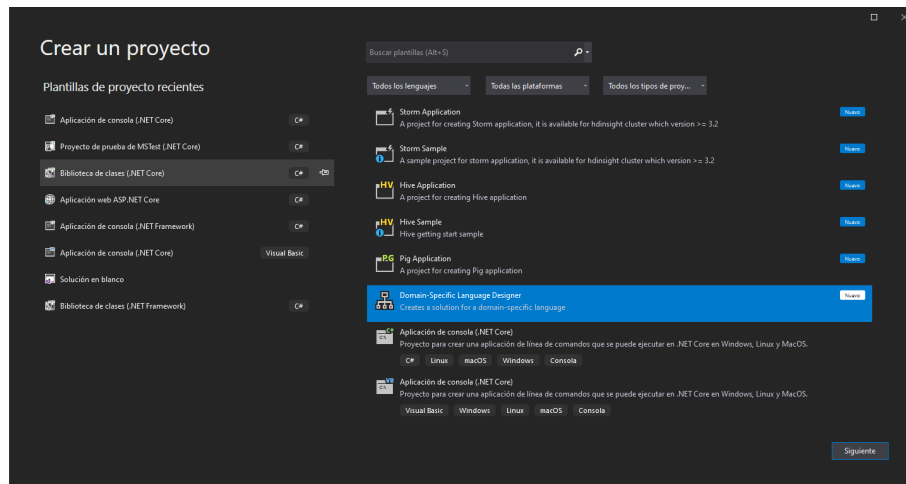


Figura 4.1: Creación de una solución de diseñador de lenguajes específicos de dominio en *Visual Studio*

Además, se debe escoger una plantilla de DSL, que sirve como puntos de partida útiles, para este caso se utilizará la plantilla de lenguaje mínimo, como se muestra en la figura 4.2.

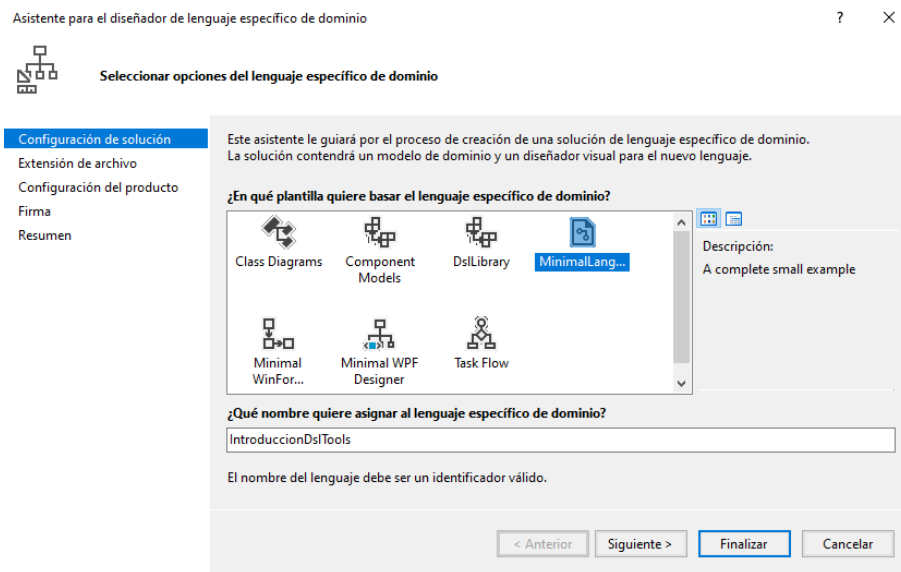


Figura 4.2: Selección de plantilla para proyecto de diseñador de lenguaje específico de dominio en *Visual Studio*

También se debe escoger una extensión para el DSL, esta extensión tiene que ser distinta a la de extensiones de archivos ya existentes, por ejemplo *docx* o *html*, en este caso se utilizará la extensión *tdsl* haciendo alusión a "Tutorial DSL", tal y como se enseña en la figura 4.3.

El asistente registrará un nuevo tipo de archivo para los archivos de modelo en el nuevo lenguaje.  
Elija una extensión de archivo y asegúrese de que sea única en el sistema.

**¿Qué extensión deben usar los archivos de modelo?**

Otras herramientas y aplicaciones registradas para controlar esta extensión:

Ninguna aplicación ni editor de Visual Studio usa esta extensión.

Nota: Puede editar la extensión más adelante en la solución de lenguaje específico de dominio.

**Figura 4.3:** Creación de extensión de archivo para proyecto de diseñador de lenguaje específico de dominio en *Visual Studio*

Junto a estos datos, además de un nombre de producto (en este caso se utilizará "TutorialDslTools"), un nombre para la compañía del proyecto (en este caso "SergioTrujillo" debido al nombre del autor del trabajo) para generar el espacio de nombres base, como en la figura 4.4.

Especifique el nombre del producto, el nombre de la compañía y el espacio de nombres de nivel superior que el asistente usa para crear la solución.

**¿Cuál es el nombre del producto al que pertenece el nuevo lenguaje específico de dominio?**

**¿Cuál es el nombre de la compañía a la que pertenece el producto?**

**¿Cuál es el espacio de nombres raíz de los proyectos en la solución?**

Nota: Puede editar esta información más adelante en la solución de lenguaje específico de dominio.

**Figura 4.4:** Establecimiento de detalles para proyecto de diseñador de lenguaje específico de dominio en *Visual Studio*

Y finalmente escoger un método de firma (como ejemplo se elegirá la opción de crear un archivo de clave de nombre seguro), como se observa en la figura 4.5.

Los procedimientos recomendados para la extensibilidad de Visual Studio requieren que los ensamblados de paquete tengan nombres seguros.

**Crear un archivo de clave de nombre seguro**  
Se creará un archivo de clave de nombre seguro (Key.snk) y se agregará a la solución.  
Los ensamblados se firmarán con la nueva clave.

**Usar clave de nombre seguro existente**

El archivo de clave de nombre seguro especificado se copiará en la solución.  
Los ensamblados se firmarán normalmente o con retraso con la clave especificada.

Nota: Si el archivo solo contiene la parte de la clave pública, los ensamblados se firmarán con retraso, lo que puede impedir que la solución se ejecute correctamente.

**Figura 4.5:** Elección de un método de firma para proyecto de diseñador de lenguaje específico de dominio en *Visual Studio*

### 4.3 Elementos más importantes en el desarrollo del DSL

Una vez creado el proyecto, se abrirá automáticamente un modelo, el metamodelo de la DSL Tool creada, nombrado como *DslDefinition.dsl*, aquí es donde se deberán establecer las metaentidades y relaciones entre estas que compondrán el lenguaje específico de dominio del proyecto, lo cual se puede observar en la figura 4.6.

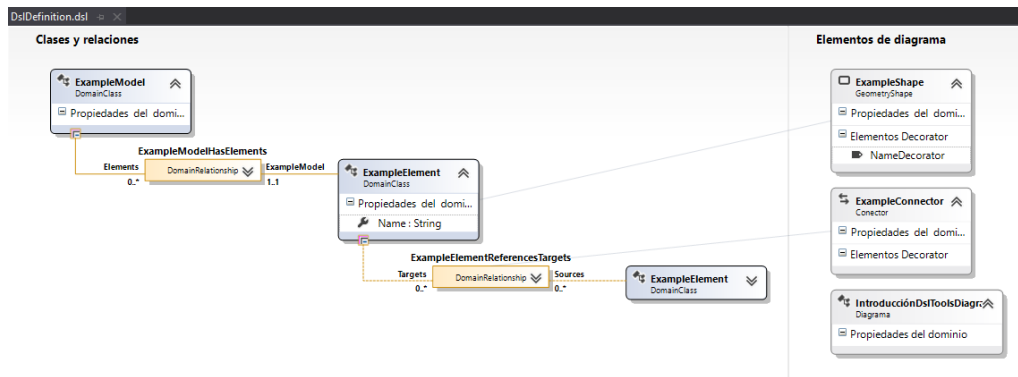


Figura 4.6: Metamodelo de un proyecto de diseñador de lenguaje específico de dominio en *Visual Studio*

A la derecha del programa se encuentra el explorador de DSL, ventana donde se realizarán acciones sobre el metamodelo, desde las más sencillas como añadir elementos hasta las más técnicas y profundas. Podemos visualizar el explorador en la figura 4.7.

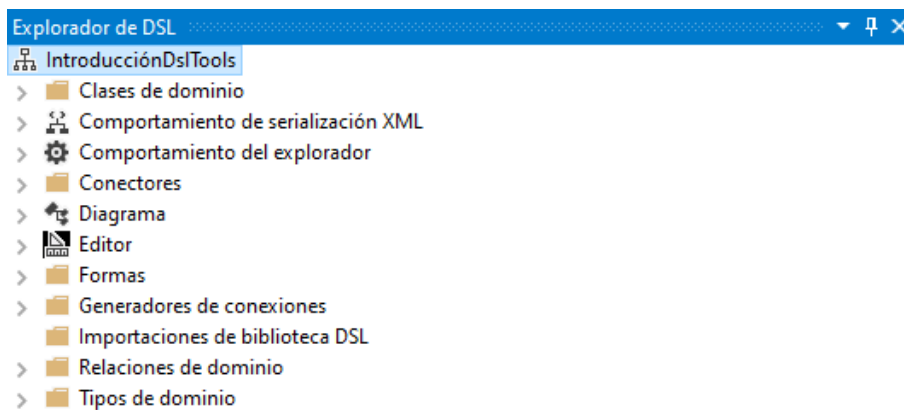


Figura 4.7: Explorador de DSL en *Visual Studio*

En la figura 4.8 se puede apreciar la ventana de detalles del DSL y la ventana de propiedades, ambas importantes para el desarrollo también, pero que no se entrará en detalle más allá de lo necesario.

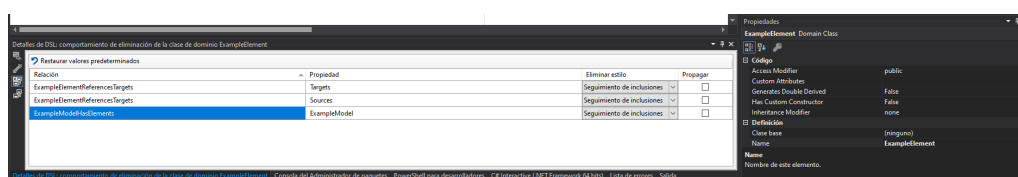


Figura 4.8: Detalles de DSL y ventana de propiedades en *Visual Studio*

Cabe mencionar que con la creación del proyecto se han incluido dos proyectos, por un lado, el proyecto *Dsl* que contiene código que define el lenguaje específico del dominio, y el proyecto *DslPackage* el cual incluye código que permite abrir y editar instancias del DSL en Visual Studio. Estos dos proyectos junto al metamodelo son los más importantes del desarrollo de un DSL.

En el desarrollo de las *DSL Tools* también se deben tener en cuenta las plantillas, que son los documentos que dictan la generación del código según las pautas establecidas.

## 4.4 Desarrollo del DSL

Para la creación del DSL se va a generar un dominio simple, se metamodelarán entidades y atributos de estos. Con estos datos resulta un metamodelo como el de la figura 4.9.

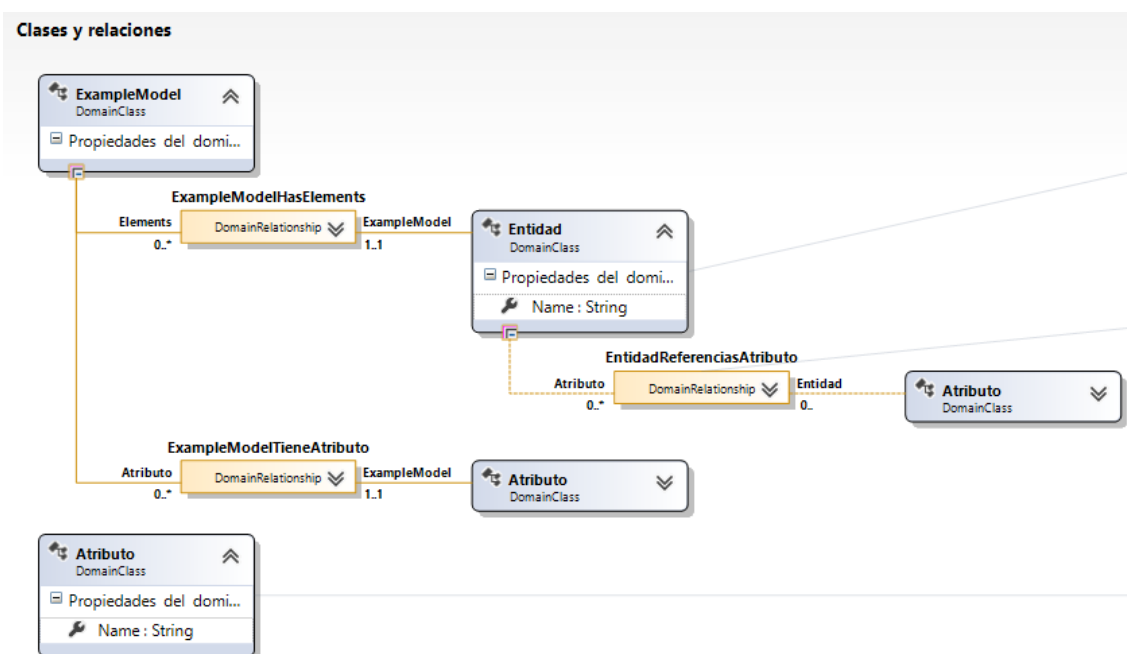


Figura 4.9: Metamodelo de DSL de entidades y atributos.

Cada vez que se cambie en metamodelo se debe volver a transformar las plantillas, esto se puede hacer utilizando la opción *Transformar todas las plantillas T4* del menú *Compilar* de la barra principal de opciones.

Si se deseara generar código de manera personalizada se deberían crear plantillas adicionales, las plantillas tienen la extensión *tt* y mezclan código propio con código *C#*. Mencionar además que si se desearan realizar acciones más técnicas como pueden ser la lógica de *mergeado*, modificar la creación de los elementos y mucho más, se deberían realizar cambios más profundos en el metamodelo...

## 4.5 Modelado y generación de código

Una vez se obtiene el metamodelo con los cambios deseados y las plantillas transformadas, se debe presionar la tecla *F5* para compilar la solución y que se lance a ejecución una instancia experimental de *Visual Studio* con la versión compilada de la *DSL Tool*. Una

vez ejecutada la instancia aparece disponible entre otros elementos un modelo de nombre *test*, en este caso se tratará de *test.tdsl*, teniendo, claro, la extensión utilizada para el DSL.

A la izquierda se puede apreciar el cuadro de herramientas con los elementos del DSL, como en la figura 4.10.

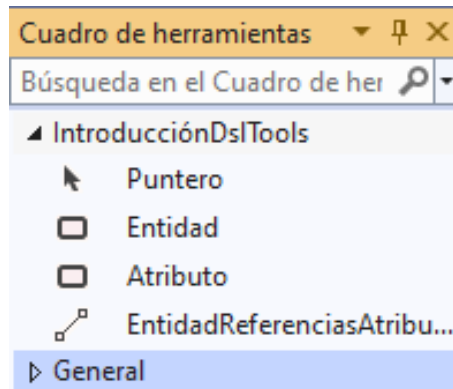


Figura 4.10: Cuadro de herramientas de la DSL Tool

Para el ejemplo se creará simplemente la entidad persona, a la cual adjudicaremos algunos atributos como cualidades, pertenencias, motivaciones y aficiones. Esto resultaría como la figura 4.11.

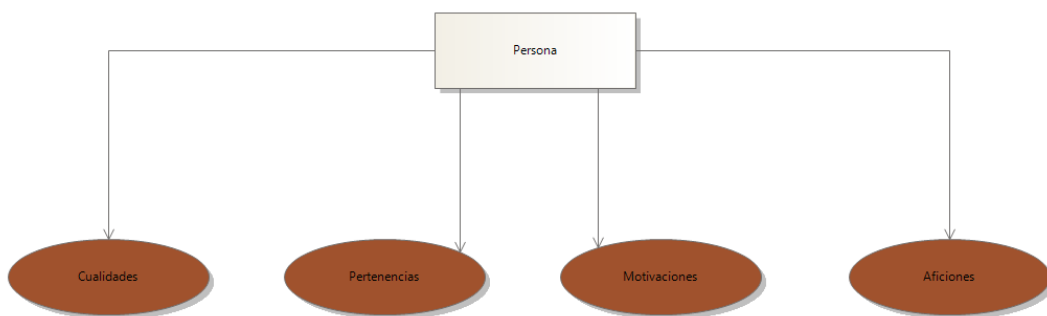


Figura 4.11: Modelo de la DSL Tool creada sobre personas

Se pueden realizar algunas modificaciones a la plantilla que se encuentra en la solución *IntroducciónDslToolsReport.tt*. En este caso se modifican las plantillas para generar clases con las entidades determinadas, con su nombre y sus listas de atributos. Además de las clases de los atributos con sus correspondientes nombres. Con todo lo anterior se obtendrá un archivo de código C# tal que así 4.12.



```
using System.Collections.Generic;
0 referencias
public class Persona
{
    0 referencias
    string Nombre { get; set; }
    0 referencias
    IEnumerable<Cualidad> cualidad { get; set; }
    0 referencias
    IEnumerable<Pertenencia> pertenencia { get; set; }
    0 referencias
    IEnumerable<Motivacion> motivacion { get; set; }
    0 referencias
    IEnumerable<Aficion> aficion { get; set; }
}
1 referencia
public class Cualidad
{
    0 referencias
    string Nombre { get; set; }
}
1 referencia
public class Pertenencia
{
    0 referencias
    string Nombre { get; set; }
}
1 referencia
public class Motivacion
{
    0 referencias
    string Nombre { get; set; }
}
1 referencia
public class Aficion
{
    0 referencias
    string Nombre { get; set; }
}
```

Figura 4.12: Ejemplo de código generado con el DSL creado.

Con las *DSL Tools* se puede generar gran parte del código, incluso de lógica, sin embargo, en ocasiones será necesario crear mecanismos para conectar el código autogenerado con código manual a implementar posteriormente.

## 4.6 Conclusiones

---

Se ha podido apreciar a lo largo de este capítulo cómo generar código empleando las *DSL Tools* con detalle de todo el procedimiento, desde la instalación de estas y creación del proyecto hasta el desarrollo del DSL y modelado utilizando este.



---

---

# CAPÍTULO 5

## Desarrollo

---

En este capítulo se explicará el desarrollo de la solución planteada. Se comenzará con un glosario de términos empleados en el desarrollo, se continuará con la especificación de requisitos, para dar paso al diseño, más tarde explicar como se realizó la programación de la misma solución y acto seguido se expondrán las pruebas aplicadas a lo programado en el anterior paso, culminando con la exposición de la metodología empleada. Para este capítulo, en diversas ocasiones se separarán la secciones en subsecciones a su vez, dividiendo el desarrollo del algoritmo con el de las modificaciones de las *DSL Tools* y protecciones de los modelos.

### 5.1 Glosario

---

Se presenta el glosario establecido en la fase de diseño con el fin de esclarecer algunos términos utilizados sobre todo en el capítulo de desarrollo.

- **Versión estándar:** Producto con el que se trabaja en el equipo de desarrollo y del que dispondrán los clientes que no hayan realizado modificaciones en los modelos.
- **Versión del cliente:** Versión en la cual el cliente ha modificado algún modelo.
- **Master:** Rama de la versión estándar.
- **Rama cliente:** Rama perteneciente al cliente en la cual aplica sus modificaciones.
- **Modelo estándar:** Última versión del modelo en la versión estándar.
- **Modelo cliente:** Última versión de un modelo que ha sido modificado en la versión del cliente.
- **Modelo puro de cliente:** Modelo que crea desde cero el cliente en el cual tendrá total libertad.

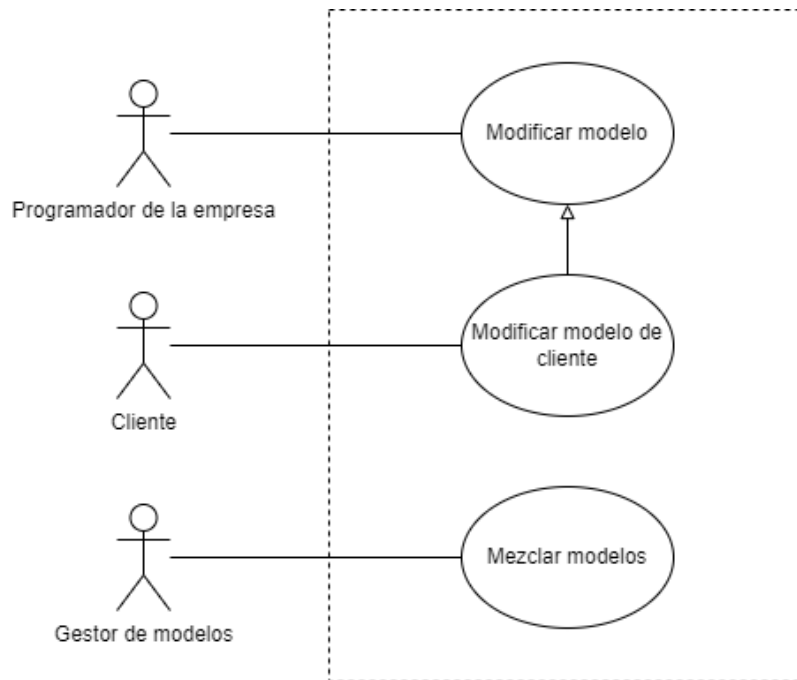
### 5.2 Especificación de requisitos

---

#### 5.2.1. Casos de uso

En el sencillo diagrama de casos de uso que se presenta a continuación [5.1](#) se pretende explicar de forma básica las principales funcionalidades de la propuesta, por un lado, los programadores de la empresa podrán modificar los modelos de las *DSL Tools* como

han estado haciendo hasta ahora, mientras que los clientes podrán modificar sus propias versiones, con algunas limitaciones. Aparte, cuando alguna de las dos versiones es modificada, se lanzará a ejecución el algoritmo de mezcla de modelos para poder actualizar la versión del cliente. Cabe mencionar que este último punto se implementará en el futuro, de ahí la abstracción existente. Estos casos de uso serán expuestos también en forma de tabla.



**Figura 5.1:** Diagrama de casos de uso.

<b>Identificador</b>	CU01
<b>Nombre</b>	Modificar modelo
<b>Actor</b>	Programador empresa
<b>Descripción</b>	El programador modifica el modelo en las <i>DSL Tools</i> como procede en el desarrollo habitual del producto

**Tabla 5.1:** Caso de uso modificar modelo

<b>Identificador</b>	CU02
<b>Nombre</b>	Modificar modelo de cliente
<b>Actor</b>	Cliente
<b>Descripción</b>	El cliente modifica su versión del modelo en las <i>DSL Tools</i> similar a como se realiza en el desarrollo habitual del producto pero con algunas limitaciones establecidas

**Tabla 5.2:** Caso de uso modificar modelo de cliente

<b>Identificador</b>	CU03
<b>Nombre</b>	Actualizar versión del cliente mezclando los modelos
<b>Actor</b>	Sistema
<b>Descripción</b>	Tras haberse actualizado la versión estándar o del cliente, en algún momento se lanzará de alguna manera no establecida aún el algoritmo para mezclar los modelos y así actualizar la versión del producto del cliente

**Tabla 5.3:** Caso de uso actualizar versión del cliente mezclando los modelos

### 5.2.2. Requisitos no funcionales

Los requisitos no funcionales definen las cualidades, características y restricciones del *software*, y a diferencia de los funcionales no determinan las funcionalidades del sistema a desarrollar. Los requisitos no funcionales se caracterizan también por ser específicos, cuantificables y verificables. Usualmente, estos son analizados en función del modelo ISO/IEC 25010 [19], también conocido por su siglas SQuARE:



**Figura 5.2:** Modelo de calidad ISO/IEC 25010.

Para el trabajo se toman en cuenta los siguientes requisitos no funcionales, explicados a través de un identificador, nombre, la característica del modelo de calidad correspondiente, una descripción y la motivación por la cual tomarlos en cuenta en el desarrollo:

<b>Identificador</b>	RNF01
<b>Nombre</b>	Complejidad funcional respecto a las <i>DSL Tools</i>
<b>Característica</b>	Adecuación funcional
<b>Descripción</b>	El algoritmo debe ser capaz de mezclar cualquier elemento existente en las <i>DSL Tools</i> de las cuales se vayan a juntar modelos.
<b>Motivación</b>	Para poder realizar un proceso automatizado que es lo que se desea, el algoritmo debe ser capaz de mezclar los modelos a modo de una caja negra a la que se le pasen estos y se obtenga el resultado automáticamente, por lo cual debe abarcar todos los casos posibles.

**Tabla 5.4:** Requisito no funcional complejidad funcional respecto a las *DSL Tools*

<b>Identificador</b>	RNF02
<b>Nombre</b>	Velocidad de ejecución del algoritmo
<b>Característica</b>	Eficiencia de desempeño
<b>Descripción</b>	El algoritmo debe de cumplir unos requisitos de velocidad de ejecución para poder ser considerado óptimo, aunque la ejecución dependerá del número de modelos mezclados y el tamaño de estos, en los casos más sencillos se debería resolver en cuestión de segundos.
<b>Motivación</b>	El algoritmo se debe ejecutar a la mayor velocidad posible pues es un proceso que se va a desplegar por cada cliente que posea evoluciones propias, si no fuera rápido no sería escalable con el número de clientes.

**Tabla 5.5:** Requisito no funcional velocidad de ejecución del algoritmo

<b>Identificador</b>	RNF03
<b>Nombre</b>	Velocidad de desempeño de las <i>DSL Tools</i> tras los cambios
<b>Característica</b>	Eficiencia de desempeño
<b>Descripción</b>	Tras los cambios implementados en las <i>DSL Tools</i> para las protecciones de los modelos, estas deben funcionar a una velocidad que perceptivamente sea la misma que antes de los cambios.
<b>Motivación</b>	Pese a que estos cambios sean necesarios para las protecciones de los modelos, no se desea que la eficiencia de desempeño de las <i>DSL Tools</i> se vea afectada, pues son esenciales en el desarrollo del producto.

**Tabla 5.6:** Requisito no funcional velocidad de desempeño de las *DSL Tools*

<b>Identificador</b>	RNF04
<b>Nombre</b>	Robustez de las protecciones de los modelos
<b>Característica</b>	Usabilidad
<b>Descripción</b>	Se desea un seguro de que las protecciones de los modelos sean aplicadas, por ello se implementarán de forma que se reafirmen.
<b>Motivación</b>	Resultaría de mala manera que los clientes realizaran cambios no deseados en los modelos, pues podrían variar el comportamiento de su versión del producto de forma no deseada por el proveedor.

**Tabla 5.7:** Requisito no funcional robustez de las protecciones de los modelos

<b>Identificador</b>	RNF05
<b>Nombre</b>	Tecnología empleada
<b>Característica</b>	Mantenibilidad
<b>Descripción</b>	Pese a que el algoritmo recibe como entrada y devuelve como salida modelos en formato XML, se requiere del uso de tecnología <i>.NET</i> y <i>C#</i> como lenguaje de programación.
<b>Motivación</b>	Se desea homogeneidad con el resto del producto, pues el equipo de desarrollo trabaja con estas tecnologías, no sería conveniente para el desarrollo que se realizara con unas distintas.

**Tabla 5.8:** Requisito no funcional tecnología empleada

<b>Identificador</b>	RNF06
<b>Nombre</b>	El algoritmo debe ser genérico
<b>Característica</b>	Mantenibilidad
<b>Descripción</b>	El algoritmo debe ser reusable por todas las <i>DSL Tools</i> .
<b>Motivación</b>	Si se desea que el algoritmo funcione de forma escalable, debe de ser lo más genérico posible para así mezclar modelos de las distintas <i>DSL Tools</i> .

**Tabla 5.9:** Requisito no funcional el algoritmo debe ser genérico

<b>Identificador</b>	RNF07
<b>Nombre</b>	Capacidad del algoritmo de ser modificado
<b>Característica</b>	Mantenibilidad
<b>Descripción</b>	El algoritmo debe ser capaz de ser modificado.
<b>Motivación</b>	Con cambios en las <i>DSL Tools</i> pueden surgir excepciones que se salgan de la genericidad, por lo tanto, el código debe ser escalable con estos cambios.

**Tabla 5.10:** Requisito no funcional capacidad del algoritmo de ser modificado

<b>Identificador</b>	RNF08
<b>Nombre</b>	Capacidad del algoritmo de ser probado
<b>Característica</b>	Mantenibilidad
<b>Descripción</b>	El algoritmo se debe realizar de forma adecuada a las reglas comunes del <i>software</i> de calidad para así poder ser probado.
<b>Motivación</b>	El algoritmo de mezcla de modelos es algo que se va a utilizar para desplegar nuevas versiones a los clientes, por lo cual debe tener una alta cobertura de pruebas.

**Tabla 5.11:** Requisito no funcional capacidad del algoritmo de ser probado

<b>Identificador</b>	RNF09
<b>Nombre</b>	Reusabilidad del código de las <i>DSL Tools</i>
<b>Característica</b>	Mantenibilidad
<b>Descripción</b>	Las modificaciones sobre las <i>DSL Tools</i> para las protecciones de los modelos deben utilizar código lo más reutilizable posible.
<b>Motivación</b>	La reusabilidad implica eficiencia en el tiempo de desarrollo, no hay que reescribir código de forma innecesaria cuando se puede realizar de forma genérica.

**Tabla 5.12:** Requisito no funcional reusabilidad del código de las *DSL Tools*

<b>Identificador</b>	RNF10
<b>Nombre</b>	Capacidad de ser modificado del código de las <i>DSL Tools</i>
<b>Característica</b>	Mantenibilidad
<b>Descripción</b>	Las modificaciones sobre las <i>DSL Tools</i> para las protecciones de los modelos deben ser modificables.
<b>Motivación</b>	Esto ocurre, ya que las <i>DSL Tools</i> están en constante cambio, por lo cual el código añadido debe ser modificable para adecuarse a posibles cambios en las decisiones de las protecciones de los modelos.

**Tabla 5.13:** Requisito no funcional capacidad de ser modificado del código de las *DSL Tools*

<b>Identificador</b>	RNF11
<b>Nombre</b>	Capacidad de ser probado del código de las <i>DSL Tools</i>
<b>Característica</b>	Mantenibilidad
<b>Descripción</b>	Las modificaciones sobre las <i>DSL Tools</i> para las protecciones de los modelos deben poder ser objetivo de pruebas.
<b>Motivación</b>	Como ya se ha dejado claro, se desea que las protecciones de los modelos sean lo más seguras posibles para evitar comportamientos indeseados tras las evoluciones del producto de los clientes, para ello se debe de probar bien el código previo a producción.

**Tabla 5.14:** Requisito no funcional capacidad de ser probado del código de las *DSL Tools*

### 5.3 Diseño

---

El apartado del diseño de la solución se abordará en dos subsecciones principales, para así poder separar de forma clara el diseño del algoritmo y, por otro lado, lo relativo al diseño de las protecciones de los modelos y modificaciones de las *DSL Tools*. Para la primera sección se recomienda antes leer el segundo apéndice del trabajo, donde se incluye un glosario con términos relevantes a la solución.



### 5.3.1. Diseño del algoritmo para mezclar modelos

Para el diseño del algoritmo se realizaron numerosas reuniones con varios integrantes del equipo en las cuales se fue refinando poco a poco. Entre otras cosas se incluyó un glosario para los términos más relevantes, pero no se va a entrar en detalles, para ello, leer el segundo apéndice disponible al final del documento.

Desde un inicio, el objetivo del diseño era que el algoritmo fuera lo más genérico posible (acorde al RNF06 5.9). Además, crear una *DSL Tool* de alto nivel gráfica en la cual el cliente modifique sus modelos, para así no emplear *Visual Studio*. Sin embargo, por el momento las modificaciones serán realizadas sobre los modelos directamente. De momento el actor Cliente de los casos de uso es realmente un programador del equipo que representa al cliente realizando estos cambios a medida.

Finalmente, se decidió, entre diversas opciones, que el algoritmo, lo que realizara, fuera coger las versiones del cliente de los modelos, y plasmar los elementos añadidos por este en las versiones estándar de los modelos. Esto se consigue añadiendo una propiedad que tengan todos los elementos de los modelos de las *DSL Tools* que declara si el elemento es procedente de versión estándar o de cliente. La nueva propiedad se calcula en función de si la rama utilizada.

#### Estructura de la solución

Para el algoritmo de mezcla de modelos se ha utilizado la estructura de proyectos *.NET* estándar del producto, para así corresponder al requisito no funcional RNF05 5.8. Por lo cual el desarrollo ha transcurrido en una solución con la estructura de la figura 5.3:

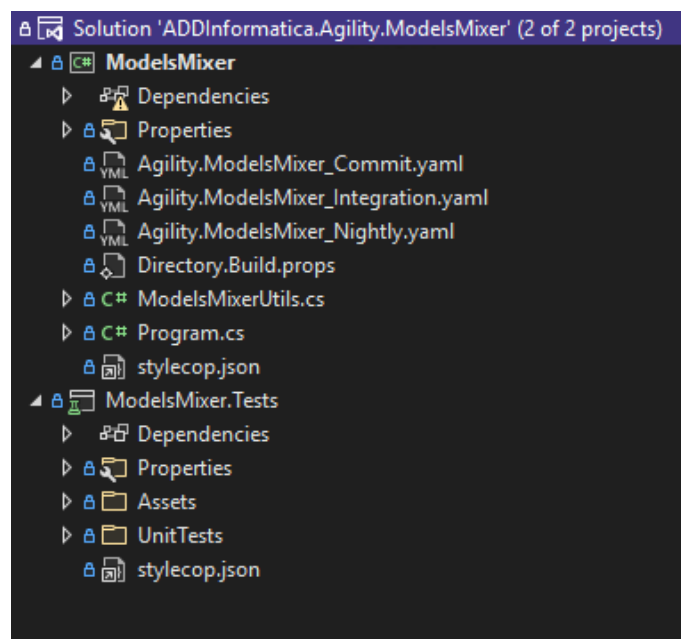


Figura 5.3: Estructura de la solución del algoritmo

En la figura 5.3 se pueden observar dos proyectos, el principal y el de tests. El principal incluye, una multitud de elementos estándar para una herramienta del producto, la clase principal y una clase de utilidad o de lógica. El segundo proyecto contiene una carpeta de *Assets* con los recursos necesarios para los tests y la carpetas con estos mismos.

### 5.3.2. Diseño de las modificaciones de las *DSL Tools* y las protecciones de los modelos

Una vez programado el algoritmo, se procedió al diseño de la parte complementaria a este, es decir, las modificaciones necesarias a las *DSL Tools*, lo que conlleva el añadido de la propiedad que declara si un elemento es de cliente o no, y las protecciones pertinentes de los modelos.

Para llevar a cabo esta tarea se realizaron reuniones con expertos de las *DSL Tools* del departamento. Se tomaron decisiones técnicas respecto de cómo implementar estas modificaciones. Además, se realizaron reuniones para declarar los requerimientos de las protecciones, para ello, se recogió en un documento, con tablas contemplando todas las posibilidades y una conclusión de si el cliente puede ejercer como acción cada posibilidad o no, este proceso fue realizado para cada elemento existente de las *DSL Tools*.

Se pueden observar ejemplos de estas tablas en las imágenes 5.4, B.19 y B.20, en donde se muestran las tablas de la clase que representa las entidades, y la relación que representa la herencia entre entidades, como ejemplos de la *DSL Tool* de dominio. También se incluyen ejemplos de aplicación con las figuras B.23 y 5.8, además, de dos ejemplos de UI, en las imágenes B.52 y B.73.

El resto de tablas se pueden encontrar en el apéndice b del documento.

Para interpretar las tablas de las clases de dominio, se debe tener en cuenta, para empezar, el título de la tabla, que representa el nombre de la clase, además, en la primera columna se declara si en la casuística se está intentando añadir el elemento a un modelo del cliente o del estándar. La segunda columna es una frase describiendo textualmente la casuística. Y las columnas tres y cuatro definen si la acción está permitida, y en algunos casos la explicación de la restricción. Para cada una de las filas existe un test.

Clase: Entity				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir una Entity?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir una Entity?	No	No	

Figura 5.4: Tabla de requerimientos de la clase *Entity*

Ahora, para comprender las tablas de las relaciones de dominio, se debe tener en cuenta, que hay algunas columnas más, las cuales son la segunda, tercera, y en algunos casos cuarta, en las cuales se declara el origen de los elementos a relacionar. Además, en el caso de las relaciones, no todas las entradas tienen un test correspondiente, pues hay casuística que no se puede dar a cabo. En estos casos, la columna del permiso tendrá el valor 'No existe'.

Relación Generalization							
Notas:							
Modelo	SubEntity	Entity	¿En este caso no existe un	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente		En un modelo de cliente, se pueden relacionar una SubEntity de cliente con una Entity de cliente?	Si	Si	
Cliente	Cliente	Estándar		En un modelo de cliente, se pueden relacionar una SubEntity de cliente con una Entity de estándar?	No existe	No puede haber una Entity de estándar en un modelo de cliente	
Cliente	Estándar	Cliente		En un modelo de cliente, se pueden relacionar una SubEntity de estándar con una Entity de cliente?	No existe	No puede haber una SubEntity de estándar en un modelo de cliente	
Cliente	Estándar	Estándar		En un modelo de cliente, se pueden relacionar una SubEntity de estándar con una Entity de estándar?	No existe	No puede haber una SubEntity de estándar en un modelo de cliente ni una Entity de estándar	
Estándar	Cliente	Cliente		En un modelo de estándar, se pueden relacionar una SubEntity de cliente con una Entity de cliente?	No existe	No puede haber una SubEntity de cliente en un modelo de estándar ni una Entity de cliente	
Estándar	Cliente	Estándar		En un modelo de estándar, se pueden relacionar una SubEntity de cliente con una Entity de estándar?	No existe	No puede haber una SubEntity de cliente en un modelo de estándar	
Estándar	Estándar	Cliente		En un modelo de estándar, se pueden relacionar una SubEntity de estándar con una Entity de cliente?	No existe	No puede haber una una Entity de cliente en un modelo de estándar	
Estándar	Estándar	Estándar		En un modelo de estándar, se pueden relacionar una SubEntity de estándar con una Entity de estándar?	No	No	

Figura 5.5: Primera tabla de requerimientos de la relación *Generalization*

Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de cliente?	Si	Si
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de estándar?	No	No
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una External Entity de estándar en un modelo de cliente
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una External Entity de estándar en un modelo de cliente
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de cliente?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente ni una External Entity de estándar
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente ni una External Entity de estándar
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de cliente?	No existe	No puede haber una ParentEntity de cliente en un modelo de estándar
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de cliente en un modelo de estándar
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se puede relacionar una ParentEntity de cliente con una External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una ParentEntity de cliente en un modelo de estándar ni una External Entity de estándar que apunte a una Entity de cliente
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se puede relacionar una ParentEntity de cliente con una External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de cliente en un modelo de estándar
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se puede relacionar una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de	Si	Si
Estándar	Estándar	Cliente	Estándar	una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de estándar?	No	No
Estándar	Estándar	Estándar	Cliente	una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una External Entity de estándar que apunte a una Entity de cliente
Estándar	Estándar	Estándar	Estándar	una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de estándar?	No	No

Figura 5.6: Segunda tabla de requerimientos de la relación *Generalization*

<b>Clase:</b>	DefaultDTO			
<b>Notas:</b>	Se elimina lo relacionado con la validación de mergear en modelo estándar o cliente pues se /			
	<b>Entidad apuntada</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	Se puede añadir un DefaultDTO que apunta a una Entidad de cliente?	Si	Si
	Estándar	Se puede añadir un DefaultDTO que apunta a una Entidad de estándar?	No	No

Figura 5.7: Tabla de requerimientos de la clase *DefaultDTO*

<b>Relación:</b>	AdHocActionReferenciasInputParameterDTO / AdHocActionReferenciasOutputParameterDTO / AdHocActionReferenciasInputDtoBase / A						
<b>Notas:</b>	Cuando se habla de DTO en este contexto, se trata de tanto DTOBase como ParameterDTO.						
	<b>Modelo</b>	<b>AdHocAction</b>	<b>DTO</b>	<b>(En este caso no existe un tercer</b>	<b>Casuística</b>	<b>¿Permitido?</b>	<b>Explicación</b>
	Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un DTO de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un DTO de estándar?	No existe	No puede haber un DTO de estándar en un modelo de cliente.
	Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un DTO de cliente?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente.
	Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un DTO de estándar?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente, ni un DTO de estándar.
	Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un DTO de cliente?	No existe	No puede haber un DTO de cliente en un modelo estándar.
	Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un DTO de estándar?	No existe	No puede haber una AdHocAction de cliente en un modelo estándar.
	Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un DTO de cliente?	No existe	No puede haber un DTO de cliente en un modelo estándar.
	Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un DTO de estándar?	No	No

Figura 5.8: Tabla de requerimientos de la relación *AdHocActionReferenciasInputOutputDTO*

Clase:		ExternalFormReference		
Notas:				
Modelo	Form apuntado	Casuística	¿Se permite?	Explicación
Cliente	Cliente	En un modelo de cliente, se puede añadir un ExternalFormReference que apunta a un Form de cliente?	Si	Si
Cliente	Estándar	En un modelo de cliente, se puede añadir un ExternalFormReference que apunta a un Form de estándar?	Si	Si
Estándar	Cliente	En un modelo de estándar, se puede añadir un ExternalFormReference que apunta a un Form de cliente?	Si	Si
Estándar	Estándar	En un modelo de estándar, se puede añadir un ExternalFormReference que apunta a un Form de estándar?	Si	Si

Figura 5.9: Tabla de requerimientos de la clase *ExternalFormReference*

Relación	ExternalCommandReference	References	Form				
Modelo	ExternalCommandReference	Command apuntado	Form	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente	
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de cliente?	No.	No	
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de cliente?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de estándar?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente ni un Form de estándar	
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de cliente?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente	
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de estándar?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente ni un Form de estándar	
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de estándar?	Si	Si	
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de estándar?	No	No	
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de estándar?	No existe	No puede haber un ExternalFormReference de estándar que apunte a un Command de cliente	
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de estándar?	No	No	

Figura 5.10: Tabla de requerimientos de la relación *ExternalCommandReferenceReferencesForm*

Además, cabe mencionar que se realizó el proceso de forma iterativa por cada una de las *DSL Tools*, es decir, se declaraban las restricciones para una, se implementaban y se procedía a recoger las de la siguiente. Este procedimiento se realizó de una forma ordenada por capas para las principales *DSL Tools*, que en este caso se trata de dominio, aplicación y UI.

## 5.4 Programación

Para el apartado de la programación de la solución se va a proceder como en el apartado anterior del diseño, es decir, se van a separar en subsecciones la programación del algoritmo y en una distinta la programación de las modificaciones de las *DSL Tools* y las protecciones de los modelos.

### 5.4.1. Programación del algoritmo para mezclar modelos

Primero, se presentará el algoritmo de una forma más visual, con un diagrama de flujo. Este diagrama se puede observar en la figura 5.11.

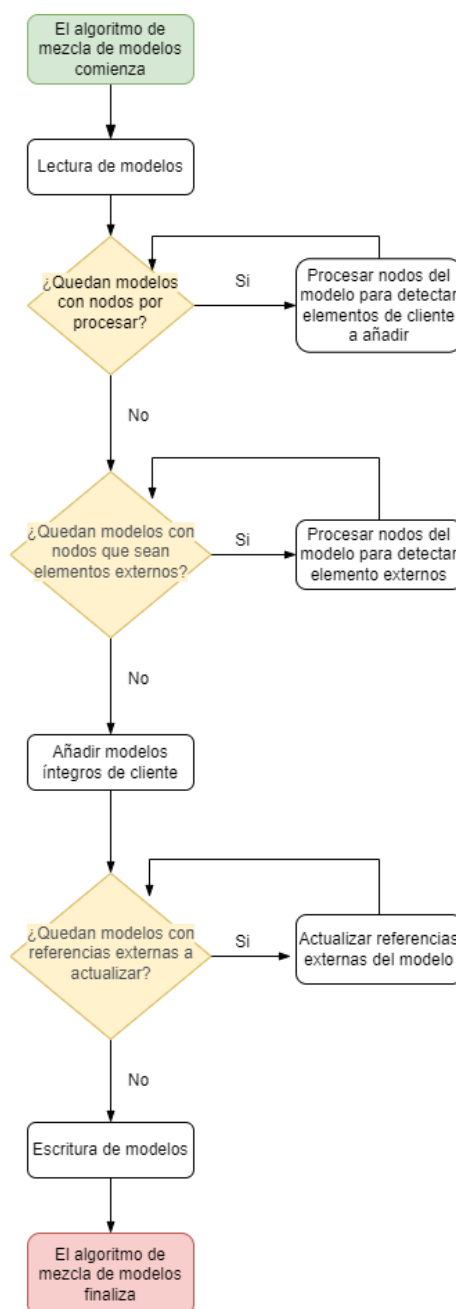


Figura 5.11: Diagrama de flujo del algoritmo de mezcla de modelos

El algoritmo cuenta con un método a modo de lanzadera que inicia todo el proceso, se trata del método *main* de la clase ejecutable de la solución. Este método realiza funciones de aplicación, y a la vez llama al método *MixModels* de la clase auxiliar, mostrado en la figura 5.12.

```
public static void MixModels(IEnumerable<string> standardModelPaths, IEnumerable<string> customerModelPaths, IEnumerable<string> pureCustomerModelsPaths)
```

Figura 5.12: Cabecera del método *MixModels*

En el capítulo 3 ya se mencionó que la librería principal usada era *System.Xml*, pues este método es el que se ocupa de coger las rutas pasadas por parámetro y cargarlas como documentos XML 5.13.

```
for (int i = 0; i < standardModelPaths.Count(); i++)
{
    string standardModelContentString = File.ReadAllText(standardModelPaths.ElementAt(i));
    string customerModelContentString = File.ReadAllText(customerModelPaths.ElementAt(i));

    XmlDocument modelCreatedFromStandardWhereCustomerChangesWillBeMixed = new XmlDocument();
    XmlDocument customerModelToTakeCustomerChanges = new XmlDocument();

    modelCreatedFromStandardWhereCustomerChangesWillBeMixed.Load(new StringReader(standardModelContentString));
    customerModelToTakeCustomerChanges.Load(new StringReader(customerModelContentString));

    StandardVersionOfModifiedXmlDocuments.Add(modelCreatedFromStandardWhereCustomerChangesWillBeMixed);
    CustomerVersionOfModifiedXmlDocuments.Add(customerModelToTakeCustomerChanges);
}
```

Figura 5.13: Bucle de carga de los modelos XML

Cabe mencionar que se pasan por separado las rutas de las versiones estándar de los modelos y las versiones de cliente de los mismos, además de los modelos que son propios de cliente.

Los modelos se procesan llamando a métodos con ellos, en este procesado es bastante común la recursión, pues al verse los documentos XML como árboles el método más sencillo era usar recursiones 5.14.

```
private static void ProcessNodeForDetectingCustomerElements(XmlDocument standardModel, XmlNode customerNodeToProcess)
{
    string isMadeByCustomerTag = "isMadeByCustomer=" + Quote + bool.TrueString + Quote;

    // The node is not a customer element, and does not contain any either, so we exit.
    if (!customerNodeToProcess.OuterXml.Contains(isMadeByCustomerTag, StringComparison.OrdinalIgnoreCase))
    {
        return;
    }
    else if (customerNodeToProcess.OuterXml.Split('>')[0].Contains(isMadeByCustomerTag, StringComparison.OrdinalIgnoreCase))
    {
        // The node is a customer element, so we process it to calculate if it should be added or not.
        ProcessCustomerElement(standardModel, customerNodeToProcess);
    }
    else if (customerNodeToProcess.HasChildNodes)
    {
        // The node is not a customer element, but it does contain any, so we look for it's children.
        foreach (XmlNode childNode in customerNodeToProcess.ChildNodes)
        {
            ProcessNodeForDetectingCustomerElements(standardModel, childNode);
        }
    }
}
```

Figura 5.14: Ejemplo de método recursivo

Y finalmente se escriben en la rutas de salida pasadas como parámetros 5.15



```
// Write process.
for (int i = 0; i < FinalXmlDocuments.Count; i++)
{
    using StreamWriter writer = File.CreateText(path: customerVersionOfModifiedModelsAndPureCustomerModelsPaths.ElementAt(i));
    using XmlWriter xmlWriter = XmlWriter.Create(writer, xmlSettings);
    FinalXmlDocuments[i].WriteTo(xmlWriter);
}
```

Figura 5.15: Escritura de los modelos XML

### 5.4.2. Programación de las modificaciones de las *DSL Tools* y las protecciones de los modelos

Se procederá a dividir el proceso seguido para las *Domain Classes* y para las *Domain Relationships*. Para el primer caso, por regla general se tomarán ejemplos de código de la meta-entidad *Entity* que representa las entidades del dominio de la aplicación, y para el segundo, la meta-relación *Association* que conecta las anteriormente mencionadas, ambas pertenecientes a la *DSL Tool* de dominio.

#### *Domain Classes*

Para las clases de dominio de las diferentes *DSL Tools* se procedió a añadir un elemento del cual colgara toda la jerarquía de los metamodelos de todos los DSL, este elemento se añadió en la *DSL Tool* de *Common*, que contiene elementos generales para todos los demás DSL como puedan ser por ejemplo las clases de casos de prueba o los elementos documentables, dicha clase se observa en la figura 5.16.

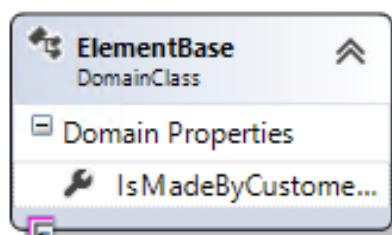


Figura 5.16: Elemento base de la jerarquía

A este elemento base se le añadió la propiedad que declara si un elemento es de cliente o estándar, por lo cual todos los elementos de las distintas *DSL Tools* tendrían esta propiedad. Al heredar todas las *Domain Classes* del elemento base, una gran parte del código solo necesitaría ser realizada una única vez de manera genérica.

Para evitar el *drag & drop* de clases de dominio por parte del cliente en los modelos, en situaciones no deseadas según los requerimientos establecidos en la fase de diseño se implementaron los métodos *CanMerge* que ofrecen las *DSL Tools* permitiendo así establecer la lógica de las protecciones esencial. Para ello primero se debe establecer en el *DSL Explorer* del metamodelo que el método es de implementación personalizada, como es posible visualizar en la figura 5.17:

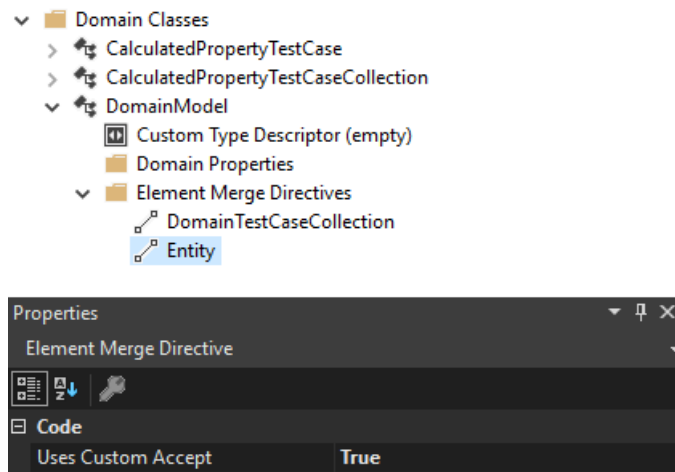


Figura 5.17: Como establecer método *CanMerge* manual

Una vez realizado esto ya se puede implementar el código en una clase parcial, como en la figura 5.18:

```

/// <summary> Queries if <see cref="Entity"/> 'rootElement' can be merged. </summary>
/// <param name="rootElement"> The root element. </param>
/// <param name="elementGroupPrototype"> The element group prototype. </param>
/// <returns> True if it succeeds, false if it fails. </returns>
2 references | - changes | - authors, - changes
private bool CanMergeEntity(ProtoElementBase rootElement, ElementGroupPrototype elementGroupPrototype)

```

Figura 5.18: Método *CanMerge* de *Entity*

Luego, para bloquear propiedades se utilizaron los *Type Descriptors* personalizados de las mismas herramientas que permiten establecer, según la lógica que se establezca, las propiedades que se deseen como de solo lectura, evitando así su modificación. Para ello primero hay que erigir el *TypeDescriptor* como personalizado en el *DSL Explorer* del metamodelo, como vemos seleccionado en la figura 5.19:

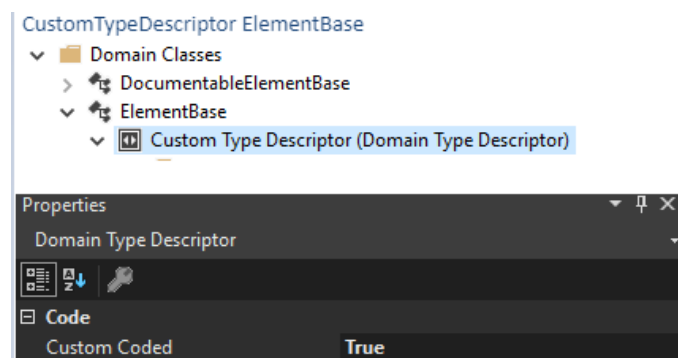


Figura 5.19: Como establecer *TypeDescriptor* personalizado.

Tras esto, se puede implementar de forma manual el método *IsPropertyDescriptorReadOnly* con la lógica que se desee según las propiedades a bloquear bajo condiciones establecidas 5.20

```

/// <inheritdoc/>
15 references | Sergio Trujillo Garcia, 136 days ago | 1 author, 8 changes
protected override bool IsPropertyDescriptorReadOnly(ElementPropertyDescriptor propertyDescriptor)

```

Figura 5.20: Método *IsPropertyDescriptorReadOnly*

Se crea también un editor base para distintos editores, *AddModelbusReferenceWrapperEditor*, que realiza la lógica necesaria para evitar o no que se despliegue el mismo, evitando así la modificación de propiedades desde los editores.

Se crean también editores a modo de recubrimiento que realizan la lógica necesaria para evitar o no que se despliegue el mismo, evitando así la modificación de propiedades desde los editores, como es el caso de *ModelElementReferenceWrapperEditor*, que impide su despliegue si la propiedad es de solo lectura, actuando como recubrimiento para el editor estándar de la selección de referencias 5.21.

```
public class ModelElementReferenceWrapperEditor : ModelElementReferenceEditor
{
    /// <inheritdoc/>
    0 references | Sergio Trujillo Garcia, 86 days ago | 1 author, 1 change
    public override object EditValue(ITypeDescriptorContext context, IServiceProvider provider, object value)
    {
        // This validation is added in order to avoid opening the text editor if the property is
        // marked as readonly.
        if (context.PropertyDescriptor.IsReadOnly)
        {
            return null;
        }

        return base.EditValue(context, provider, value);
    }
}
```

Figura 5.21: Editor *ModelElementReferenceWrapperEditor*

Finalmente, para otorgar mas robustez y funcionar acorde al RNF04 5.7, se añaden validaciones con lógica similar a lo codificado en los métodos *Can Merge* para así evitar que modificaciones no deseadas se realicen manualmente en los XML y por si los métodos *Can Merge* no funcionaran 5.22.

```
/// <summary>
/// ... Validates that a customer <see cref="Entity"/> cannot be merged in a standard <see cref="DomainModel"/>.
/// </summary>
/// <param name="context"> The validation context. </param>
[ValidationMethod(ValidationCategories.Menu | ValidationCategories.Save)]
2 references | 0/2 passing | - changes | -authors, -changes
internal void ValidateCustomerEntityCannotBeMergedInStandardModel(ValidationContext context)
```

Figura 5.22: Validación sobre *Entity*

Como pequeña mención, nombrar que la lógica mencionada en este apartado gira en torno a si la instancia de la *Domain Class* es de cliente o estándar, ídem para el modelo, además del modo en el que estén funcionando las *DSL Tools*, lo cual viene dado por parte de la rama de *Git*, que según su nombre se puede conocer si esta misma pertenece a algún cliente o si forma parte del desarrollo del producto estándar.

## Domain Relationships

La programación de las relaciones del dominio resultó algo más complicada, pues no se podía establecer una relación padre de toda la jerarquía como si para las clases de dominio. La solución resultó en crear una interfaz que contuviera la propiedad necesaria y algunos elementos genéricos que también se necesitarían [5.23](#)

```
public interface IRelationshipBase : ICustomModelElement
{
    /// <summary>
    /// Gets or sets the value that states if the element was made by the customer.
    /// </summary>
    /// <value> The value that states if the element was made by the customer. </value>
    /// <remarks>
    /// It is needed to catch the value of the property for making the method not to
    /// calculate it every time a model is opened, because it would be changed if the DSL
    /// Tool changes to another mode.
    /// </remarks>
    99+ references | 0/71 passing | Sergio Trujillo Garcia, 153 days ago | 1 author, 2 changes
    new bool? IsCustomerInstance { get; set; }
}
```

Figura 5.23: Interfaz común a las relaciones.

Para realizar todo de una forma un tanto más genérica y evitar despistes por parte del programador, se añadieron plantillas de generación automática de código para producir el código común a las relaciones necesario [5.24](#).

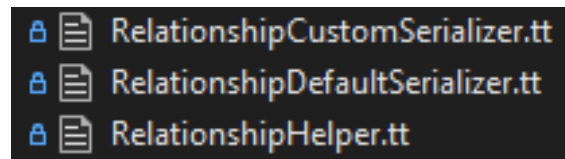


Figura 5.24: Plantillas de generación de código relacionadas con las relaciones.

En el caso de las *Domain Relationships*, no se utilizaron métodos *Can Merge* como para las *Domain Classes* debido al funcionamiento de las *DSL Tools*, sino que se debían utilizar *Connection Builders*, que contienen el método para designar si una relación puede conectar los elementos origen y destino recibidos en el momento de la creación. Para emplear esta utilidad, una vez más, hay que especificarlo en el *DSL Explorer* del metamodelo [5.25](#):



Source role directives	Target role directives	Custom accept	Custom connect
Domain Class	Path	<input checked="" type="checkbox"/>	<input type="checkbox"/>
EntityBase		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 5.25: Como establecer *CanAccept* personalizado para las relaciones.

Tras esto, se podrá implementar el método personalizado en el *ConnectionBuilder* [5.26](#)

```
/// <summary> Indicates if this relationship can be merged or not. </summary>
/// <returns> True if the relationship can be merged, false if not. </returns>
/// reference | - changes | - authors, - changes
private static bool CanMergeRelationshipImplementation(EntityBase sourceModelElement, EntityBase targetModelElement)
```

Figura 5.26: Método *CanAccept* personalizado en *ConnectionBuilder*.

Aquí es donde entra en juego el código autogenerado, entre las plantillas creadas se encuentran los *Relationship Helpers*, clases *Helper* por cada relación, las cuales contienen un método con lógica común a todas las relaciones y que a su vez llaman a un método que deberá ser implementado por el programador manualmente y que funcione acorde a la restricciones establecidas en el diseño 5.27.

```

/// <summary> Indicates if this <see cref="Association"/> can be merged or not. </summary>
/// <param name="sourceModelElement"> The source element for the relationship. </param>
/// <param name="targetModelElement"> The target element for the relationship. </param>
/// <returns> True if the Association can be merged, false if not. </returns>
1 reference | - changes | - authors, - changes
public static bool CanMergeRelationship(EntityBase sourceModelElement, EntityBase targetModelElement)
{
    string modelAbsolutePath = sourceModelElement.GetModelAbsolutePath();
    bool isCustomerMode = CustomerOrStandardModeUtils.CheckIfCurrentBranchIsCustomer(modelAbsolutePath);
    if (!isCustomerMode || Path.GetFileName(modelAbsolutePath).Contains(CustomerOrStandardModeUtils.CustomerModelString))
    {
        return true;
    }
    return CanMergeRelationshipImplementation(sourceModelElement, targetModelElement);
}

```

Figura 5.27: Método auxiliar auto generado de clase *Helper*.

Al realizar la llamada mencionada, se asegura que el programador implementará el método en una clase parcial 5.28, pues la clase *Helper* no compilará al llamar a un método inexistente. A su vez, deberá implementar que en el *Connection Builder* se llame al método de la clase *Helper*, cerrando así el flujo.

```

/// <summary> Indicates if this relationship can be merged or not. </summary>
/// <returns> True if the relationship can be merged, false if not. </returns>
1 reference | - changes | - authors, - changes
private static bool CanMergeRelationshipImplementation(EntityBase sourceModelElement, EntityBase targetModelElement)

```

Figura 5.28: Método de clase parcial con la lógica para permitir o no el agregado de la relación.

Ya que la propiedad que declara si la relación ha sido añadida en el producto cliente o estándar ha sido autogenerada, para evitar tener que implementar los serializadores manualmente, se crearon las plantillas correspondientes a estos 5.24. Para usar estos, hay que establecer como personalizado el serializado de la clase en el *DSL Explorer* del meta-modelo 5.29:

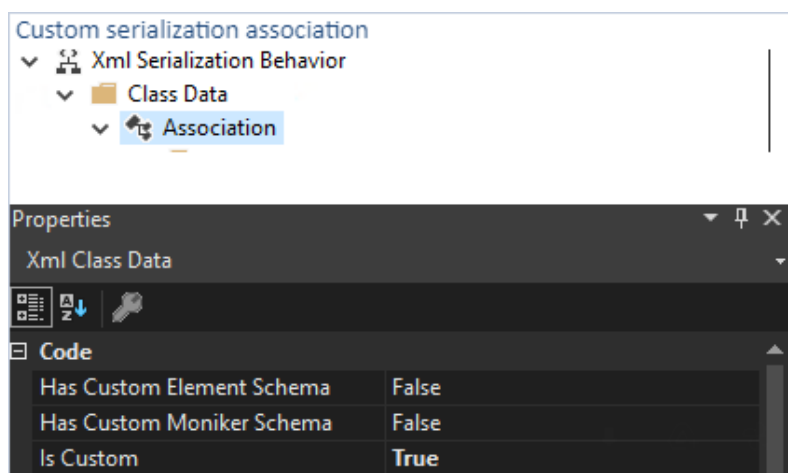


Figura 5.29: Como establecer serializado personalizado de las relaciones.

Con esto, se utilizarán los serializadores autogenerados con las plantillas.

Al igual que para las clases de dominio, se crearon validaciones para otorgar más robustez y evitar cambios indeseados introducidos manualmente en los XML. Estas en

este caso contendrían lógica similar a la de los *Connection Builders* en lugar de los *Can Merge*.

Mencionar también que la lógica implementada de las relaciones depende en gran parte de los elementos que intenta conectar.

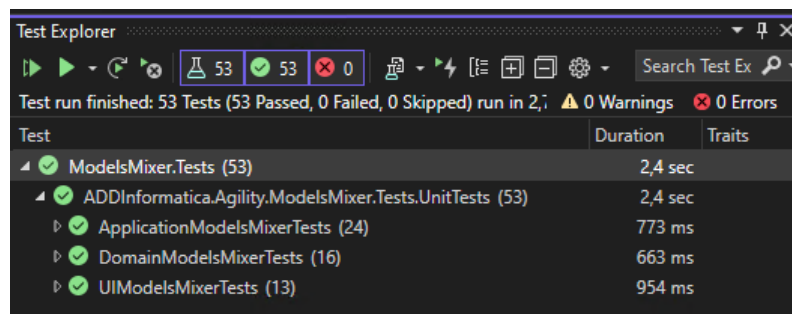
## 5.5 Pruebas

El apartado de pruebas, para esclarecer más el documento, se va a dividir una vez más en dos subsecciones, una dedicada a las pruebas del algoritmo y otra dedicada al testeo de las modificaciones de las *DSL Tools* y las protecciones de los modelos.

### 5.5.1. Pruebas del algoritmo

Para realizar las pruebas al código correspondiente al algoritmo, se realizó un análisis de caja negra.

Se recogieron las posibilidades existentes al evolucionar los modelos de la versión de cliente y/o la de estándar. La lista de tests existentes se puede observar en las figuras 5.30, 5.31, 5.32 y 5.33. Se modelaron tanto las versiones de los modelos de estándar como las de cliente, además del resultado deseado tras la mezcla. La lista de modelos para estos tests se puede visualizar en la imagen de la figura 5.34.



Test	Duration	Traits
✓ ModelsMixer.Tests (53)	2,4 sec	
✓ ADDInformatica.Agility.ModelsMixer.Tests.UnitTests (53)	2,4 sec	
✓ ApplicationModelsMixerTests (24)	773 ms	
✓ DomainModelsMixerTests (16)	663 ms	
✓ UIModelsMixerTests (13)	954 ms	

Figura 5.30: Lista de tests del algoritmo de mezcla de modelos.

Test Name	Duration (ms)
DomainModelsMixerTests (16)	663 ms
MixModels_GivenModelsAddedExternalCustomerEnumer...	57 ms
MixModels_GivenModelsAddedRefactoredEnumeratedTo...	59 ms
MixModels_GivenModelsCustomerAddFieldToStandardE...	20 ms
MixModels_GivenModelsCustomerAddRelationFromCust...	63 ms
MixModels_GivenModelsCustomerAddRelationFromCust...	92 ms
MixModels_GivenModelsCustomerAddRelationFromCust...	51 ms
MixModels_GivenModelsCustomerAddRelationFromCust...	56 ms
MixModels_GivenModelsCustomerEntityMadeAsSubTyp...	66 ms
MixModels_GivenModelsRefactoredStandardEntityUsedI...	44 ms
MixModels_GivenModelWithAnEntityWithAFieldAddedC...	11 ms
MixModels_GivenModelWithAnEntityWithAFieldAddedFi...	29 ms
MixModels_GivenModelWithAnEntityWithAFieldAddedFi...	28 ms
MixModels_GivenModelWithAnEntityWithAFieldAddedFi...	20 ms
MixModels_GivenModelWithAnEntityWithAFieldAddedFi...	17 ms
MixModels_GivenModelWithAnEntityWithAFieldAddedP...	26 ms
MixModels_GivenModelWithAnEntityWithAFieldAddedV...	24 ms

Figura 5.31: Lista de tests de dominio del algoritmo de mezcla de modelos.

Test Name	Duration (ms)
ApplicationModelsMixerTests (24)	773 ms
MixModels_GivenModelsAddedExternalInputCustomerSi...	40 ms
MixModels_GivenModelsAddedRefactoredParameterDto...	38 ms
MixModels_GivenModelsAddedRefactoredSimpleDtoAsI...	45 ms
MixModels_GivenModelsCustomerCalculatedDtoUsesAs...	66 ms
MixModels_GivenModelsCustomerCompositeDtoUsesRe...	106 ms
MixModels_GivenModelsCustomerReportUsesRefactored...	49 ms
MixModels_GivenModelsSameReferenceToEntityFieldAd...	31 ms
MixModels_GivenModelsStandardEntityUsedByCustomer...	48 ms
MixModels_GivenModelWithACalculatedDtoWithAFieldA...	146 ms
MixModels_GivenModelWithACalculatedDtoWithAFieldA...	16 ms
MixModels_GivenModelWithACalculatedDtoWithAFieldA...	30 ms
MixModels_GivenModelWithACalculatedDtoWithAFieldA...	11 ms
MixModels_GivenModelWithACompositeDtoWithAField...	13 ms
MixModels_GivenModelWithACompositeDtoWithAField...	10 ms
MixModels_GivenModelWithAParameterDtoWithAFieldA...	9 ms
MixModels_GivenModelWithAParameterDtoWithAFieldA...	17 ms
MixModels_GivenModelWithAPrivateAdHocContainerWi...	14 ms
MixModels_GivenModelWithAPrivateAdHocContainerWi...	8 ms
MixModels_GivenModelWithASimpleDtoWithAFieldAdd...	10 ms
MixModels_GivenModelWithCalculatedDtoWithAFieldAd...	10 ms
MixModels_GivenModelWithCalculatedDtoWithAFieldAd...	9 ms
MixModels_GivenModelWithCompositeDtoWithAFieldA...	12 ms
MixModels_GivenModelWithCompositeDtoWithAFieldA...	15 ms
MixModels_GivenModelWithSimpleDtoWithAFieldAdded...	20 ms

Figura 5.32: Lista de tests de aplicación del algoritmo de mezcla de modelos.

Test Name	Duration (ms)
UIModelsMixerTests (13)	954 ms
MixModels_GivenModelAddedFieldInCustomerToRefacto...	72 ms
MixModels_GivenModelAddedFieldInCustomerToRefacto...	80 ms
MixModels_GivenModelAddedFieldInStandardThatChan...	73 ms
MixModels_GivenModelAddedFieldToLayoutInCustomer...	81 ms
MixModels_GivenModelAddedFieldToTableInCustomer_R...	55 ms
MixModels_GivenModelAddedLayoutInCustomer_Return...	117 ms
MixModels_GivenModelAddedLayoutInCustomerToRefac...	15 ms
MixModels_GivenModelsCustomerFieldReferencesNotAd...	37 ms
MixModels_GivenModelsCustomerFormUsesRefactored...	111 ms
MixModels_GivenModelsSameReferenceToDtoFieldAdde...	35 ms
MixModels_GivenModelWithALayoutAddedFieldAndAdd...	143 ms
MixModels_GivenModelWithALayoutAddedLayoutAndA...	19 ms
MixModels_GivenModelWithATableAddedFieldAndAdde...	116 ms

Figura 5.33: Lista de tests de UI del algoritmo de mezcla de modelos.

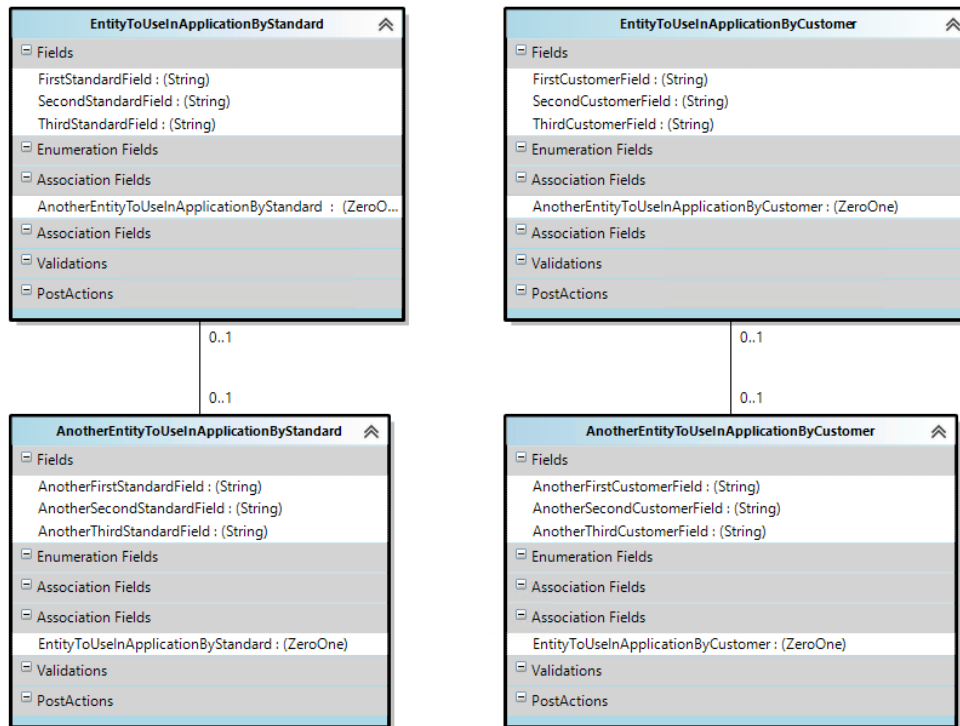
Model Name	Files
Assets	
Models	
Application	
AChangeAddedInStandardAndADifferentOneInCustomer	
SameChangeAddedInStandardAndCustomer	
SameReferenceAddedInStandardAndCustomer	
StandardIsRefactored	
DomainModel.ddsl	
DomainModel.ddsl.diagram	
Domain	
AChangeAddedInStandardAndADifferentOneInCustomer	
SameChangeAddedInStandardAndCustomer	
StandardIsRefactored	
UI	
AChangeAddedInStandardAndADifferentOneInCustomer	
SameChangeAddedInStandardAndCustomer	
SameReferenceAddedInStandardAndCustomer	
StandardIsRefactored	
ApplicationModel.adsl	
ApplicationModel.adsl.diagram	
DomainModel.ddsl	
DomainModel.ddsl.diagram	

Figura 5.34: Lista de modelos para los tests del algoritmo de mezcla de modelos.

Para ejemplificar, se tomará de muestra un caso de prueba en el que el cliente propaga un campo proveniente de la versión estándar, a la capa de aplicación, para así poder mostrarlo en su interfaz, al mismo tiempo se realizará el mismo cambio pero con un campo distinto en la versión estándar.

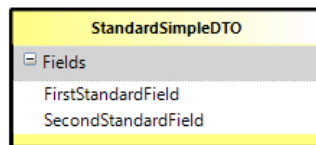
El modelo de dominio, que no recibe cambios, por lo cual solo se mostrará por esta vez, se puede visualizar en la figura 5.35.





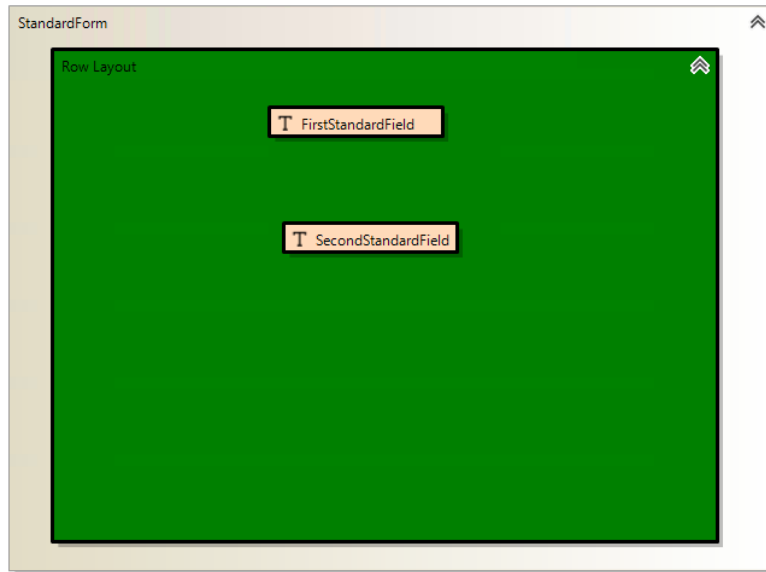
**Figura 5.35:** Modelo de dominio de test del algoritmo de mezcla de modelos.

A continuación, se enseñan los modelos de aplicación 5.36 y UI 5.37 de la versión estándar, con los cambios mencionados.



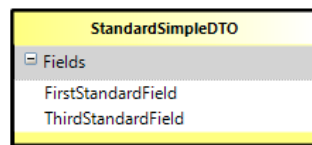
**Figura 5.36:** Modelo de aplicación de la versión de estándar de test del algoritmo de mezcla de modelos.

Los modelos de UI son una abstracción de como es la interfaz final del usuario, es decir, se modelan con la *DSL Tool*, y después se genera la interfaz final con los modelos.

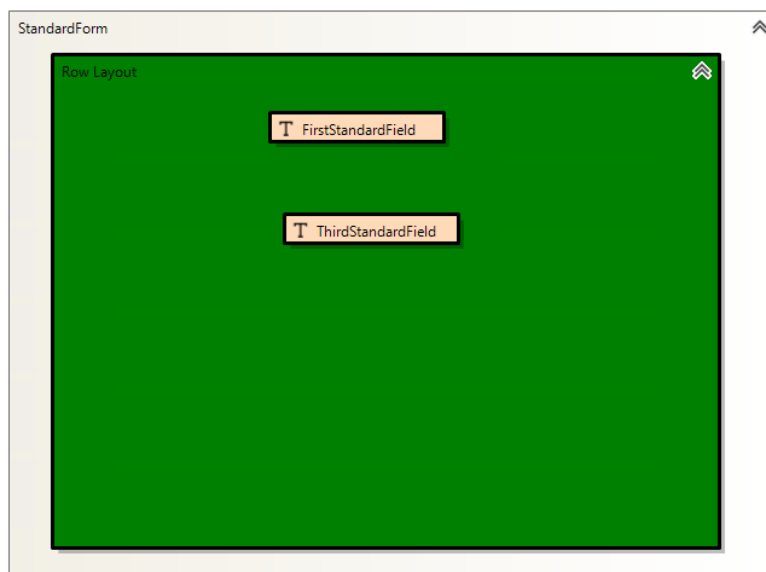


**Figura 5.37:** Modelo de UI de la versión de estándar de test del algoritmo de mezcla de modelos.

Ahora, se muestran los modelos de aplicación 5.38 y UI 5.39 de la versión de cliente, también con los cambios pertinentes.

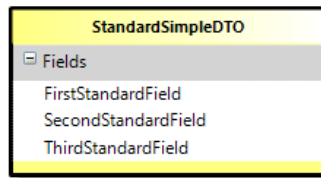


**Figura 5.38:** Modelo de aplicación de la versión de cliente de test del algoritmo de mezcla de modelos.

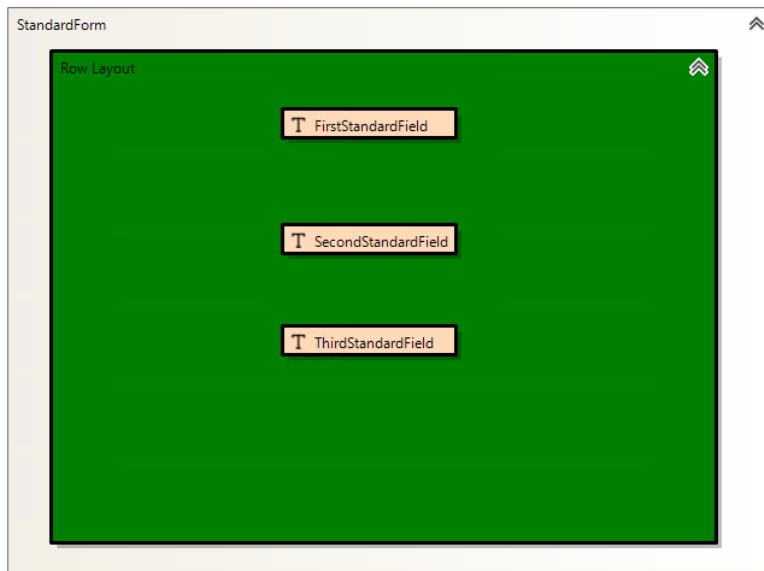


**Figura 5.39:** Modelo de UI de la versión de cliente de test del algoritmo de mezcla de modelos.

Y finalmente, se exponen los modelos de aplicación 5.40 y UI 5.41 que se esperan como resultado en el test.



**Figura 5.40:** Modelo de aplicación esperado como resultado de test del algoritmo de mezcla de modelos.



**Figura 5.41:** Modelo de UI esperado como resultado de test del algoritmo de mezcla de modelos.

Para cada casuística, una vez modelada, se creaba un test que recibía las versiones de los modelos estándar, de cliente, y resultante deseada, obtenía del algoritmo los modelos producto de la mezcla, y los comparaba con el resultante deseado para mostrar resultados positivos o negativos. El test del ejemplo anterior se puede ver en la figura 5.42.

```

</summary>
// ... The method: MixModels, when a standard model is given, and fields are added at the
// ... same time in customer and standard to layout, the method mix this models.
</summary>
[TestMethod]
|0 references | Sergio Trujillo Garcia, 344 days ago | 1 author, 6 changes
public void MixModels_GivenModelWithALayoutAddedFieldAndAddedFieldToLayout_ReturnsMixedModels()
{
    // Arrange.
    string destinyModelDirectory = @"..\..\..\Assets\Models\UI\SameChangeAddedInStandardAndCustomer\AddedFieldToLayout\MixedModels\";

    if (!Directory.Exists(destinyModelDirectory))
    {
        Directory.CreateDirectory(destinyModelDirectory);
    }

    try
    {
        string standardModelDirectory = @"..\..\..\Assets\Models\UI\SameChangeAddedInStandardAndCustomer\AddedFieldToLayout\Standard\";
        string standardDomainModelPath = standardModelDirectory + "DomainTestModel.ddsl";
        string standardApplicationModelPath = standardModelDirectory + "ApplicationTestModel.adsl";
        string standardUIModelPath = standardModelDirectory + "UITestModel.uifdsl";

        string customerModelDirectory = @"..\..\..\Assets\Models\UI\SameChangeAddedInStandardAndCustomer\AddedFieldToLayout\Customer\";
        string customerDomainModelPath = customerModelDirectory + "DomainTestModel.ddsl";
        string customerApplicationModelPath = customerModelDirectory + "ApplicationTestModel.adsl";
        string customerUIModelPath = customerModelDirectory + "UITestModel.uifdsl";

        string expectedMixedModelDirectory = @"..\..\..\Assets\Models\UI\SameChangeAddedInStandardAndCustomer\AddedFieldToLayout\ExpectedMixedModels\";
        string expectedDomainMixedModelPath = expectedMixedModelDirectory + "DomainTestModel.ddsl";
        string expectedApplicationMixedModelPath = expectedMixedModelDirectory + "ApplicationTestModel.adsl";
        string expectedUIMixedModelPath = expectedMixedModelDirectory + "UITestModel.uifdsl";

        string destinyDomainModelPath = destinyModelDirectory + "DomainTestModel.ddsl";
        string destinyApplicationModelPath = destinyModelDirectory + "ApplicationTestModel.adsl";
        string destinyUIModelPath = destinyModelDirectory + "UITestModel.uifdsl";

        List<string> standardModelPaths = new List<string>();
        standardModelPaths.Add(standardDomainModelPath);
        standardModelPaths.Add(standardApplicationModelPath);
        standardModelPaths.Add(standardUIModelPath);

        List<string> pureCustomerModelPaths = new List<string>();

        List<string> mixedModelsDestinyPaths = new List<string>();
        mixedModelsDestinyPaths.Add(destinyDomainModelPath);
        mixedModelsDestinyPaths.Add(destinyApplicationModelPath);
        mixedModelsDestinyPaths.Add(destinyUIModelPath);

        File.Copy(customerDomainModelPath, destinyDomainModelPath, true);
        File.Copy(customerApplicationModelPath, destinyApplicationModelPath, true);
        File.Copy(customerUIModelPath, destinyUIModelPath, true);

        // Act.
        ModelsMixerUtils.MixModels(standardModelPaths, mixedModelsDestinyPaths, pureCustomerModelPaths);

        // Assert.
        string expectedDomainMixedModelString = File.ReadAllText(expectedDomainMixedModelPath);
        XmlDocument expectedDomainMixedModelDocument = new XmlDocument();
        expectedDomainMixedModelDocument.Load(new StringReader(expectedDomainMixedModelString));

        string mixedDomainModelString = File.ReadAllText(destinyDomainModelPath);
        XmlDocument mixedDomainModelDocument = new XmlDocument();
        mixedDomainModelDocument.Load(new StringReader(mixedDomainModelString));

        mixedDomainModelDocument.OuterXml.Should().Be(expectedDomainMixedModelDocument.OuterXml);

        string expectedApplicationMixedModelString = File.ReadAllText(expectedApplicationMixedModelPath);
        XmlDocument expectedApplicationMixedModelDocument = new XmlDocument();
        expectedApplicationMixedModelDocument.Load(new StringReader(expectedApplicationMixedModelString));

        string mixedApplicationModelString = File.ReadAllText(destinyApplicationModelPath);
        XmlDocument mixedApplicationModelDocument = new XmlDocument();
        mixedApplicationModelDocument.Load(new StringReader(mixedApplicationModelString));

        mixedApplicationModelDocument.OuterXml.Should().Be(expectedApplicationMixedModelDocument.OuterXml);

        string expectedUIMixedModelString = File.ReadAllText(expectedUIMixedModelPath);
        XmlDocument expectedUIMixedModelDocument = new XmlDocument();
        expectedUIMixedModelDocument.Load(new StringReader(expectedUIMixedModelString));

        string mixedUIModelString = File.ReadAllText(destinyUIModelPath);
        XmlDocument mixedUIModelDocument = new XmlDocument();
        mixedUIModelDocument.Load(new StringReader(mixedUIModelString));

        mixedUIModelDocument.OuterXml.Should().Be(expectedUIMixedModelDocument.OuterXml);
    }
    finally
    {
        // Clean up.
        if (Directory.Exists(destinyModelDirectory))
        {
            Directory.Delete(destinyModelDirectory, true);
        }
    }
}

```

Figura 5.42: Test del algoritmo de mezcla de modelos.

Una vez creados todos los tests, se comprobó la cobertura de estos, obteniendo un resultado del 100 %, por lo cual el algoritmo aparentemente estaba probado correctamente,

a no ser que surgiera alguna excepción no esperada que provocara un comportamiento no deseado.

Cabe mencionar además que se plantea la posibilidad de probar los métodos del algoritmo mediante caja blanca, analizándolos por separado y codificando los casos de prueba necesarios, sin embargo, por el momento se optó por incluir solo las pruebas de caja negra mencionadas.

### 5.5.2. Pruebas de las modificaciones de las *DSL Tools* y las protecciones de los modelos

Para el testeo de las modificaciones de las *DSL Tools* y las protecciones de los modelos, se empleó el mismo documento mencionado en la fase de diseño que contenía las tablas de verdad explicando la posible casuística de las *Domain Classes* y *Domain Relationships*.

Se planteó crear un test por cada caso existente, una vez más, utilizando la técnica de caja negra, es decir, utilizando el código con desconocimiento acerca de su funcionamiento, así, se tendrían ya implementados todos los casos posibles, y si en algún momento hubiera un cambio en las restricciones se conocería gracias a los tests si la lógica implementada no funciona de la forma deseada.

Estos tests fueron programados sobre las validaciones implementadas en la fase anterior, pues el código puro de las *DSL Tools* era muy complicado de probar.

En los tests se crean los elementos necesarios para el test y se llama a la validación, esperando errores o no de estas.

Por ejemplo, en la figura 5.43 se puede observar un caso de prueba, resuelto en el test de la imagen 5.44.



De importante mención es que todos los tests implementados fueron añadidos en el documento para dejar constancia y así poder realizar un mantenimiento limpio y organizado. En total se realizaron unas 150 pruebas automatizadas.

## 5.6 Metodología

---

La metodología empleada en este trabajo es una adaptación de *Scrum*.

En este caso, al ser un producto en desarrollo por parte de un departamento de I+D+i, y la migración de otro que si está en producción, no se han realizado comprobaciones con el cliente ni pruebas de aceptación con este, sin embargo, si se ha podido ver presente una figura actuando de *Producto Owner*. Este papel lo ha desempeñado el jefe del departamento de I+D+i.

Se convocaban reuniones de forma pertinente para establecer requisitos, diseño, dudas, etc. Y con los expertos necesarios del departamento para afrontar cada desafío. Primero se trabajó en el algoritmo y luego se trabajó por capas de las *DSL Tools* para las restricciones y protecciones de los modelos, empezando por las *DSL Tools* de más bajo nivel. Para pasar al proceso de la siguiente capa se debía haber finalizado la anterior, no dejando así cabos sueltos. Sin embargo, solo se realizó el proceso para las tres *DSL Tools* principales, pues el equipo cuenta con más de estas herramientas entre sus filas, pero se consideró que no se abarcarían de momento.

El autor empleó unas 230 horas en el desarrollo, abarcando desde principios de septiembre de 2021 hasta febrero de 2022. En la figura 5.45 se puede observar la línea de tiempo del proyecto.

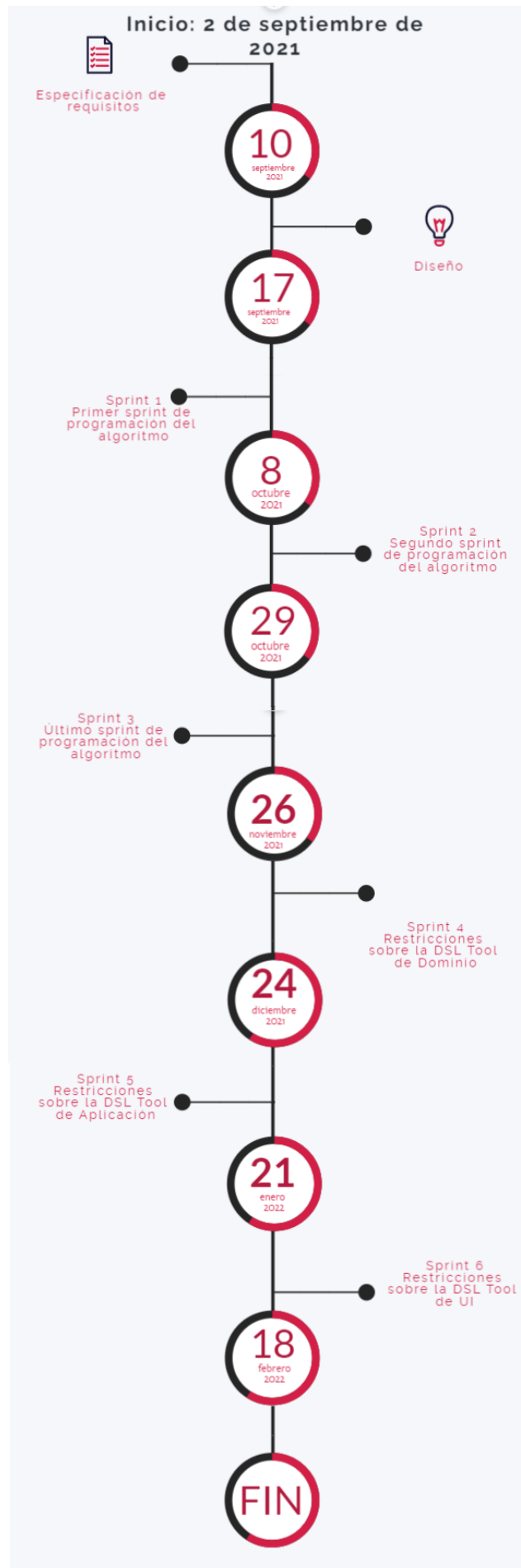


Figura 5.45: Línea temporal del proyecto.



Cabe mencionar también que el control del trabajo realizado y el almacenamiento de los documentos pertinentes se gestionaba a través de un sistema de control de incidencias interno, y el trabajo por capas era efectuado por *sprints* de unas tres semanas a través de un sistema de *Pull Requests*.

Las tareas fueron organizadas a través de un *Backlog*, ubicado en las libretas de *One-Note* del departamento, manteniendo un control constante sobre las tareas.



---

## CAPÍTULO 6

# Caso de estudio

---

A continuación se expondrá un caso de estudio del algoritmo de mezcla de modelos, presentando un caso, evolucionando los modelos, simulando actualizaciones del producto tanto estándar como de un cliente, para finalmente mostrar el resultado.

### 6.0.1. Explicación del caso

En este supuesto caso, partiremos de una situación en la que se le ha desplegado a un cliente la versión del estándar actual.

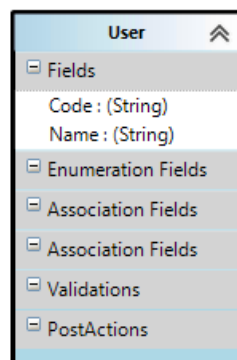
Una vez el producto está desplegado, los analistas del equipo de desarrollo consideran necesario incluir a los usuarios una propiedad que declare la fecha en la que estos entraron a la residencia. A su vez, esta propiedad se deberá ver reflejada en las fichas de usuario.

Simultáneamente, el cliente considera que sus usuarios tengan asociados el nombre de un familiar, y que a su vez, este se muestre en sus fichas.

Tras aplicar estos cambios, se desplegarán los mecanismos necesarios para aplicar a la versión del cliente el algoritmo de mezcla de modelos, resultando en un producto con ambas mejoras.

### 6.0.2. Versión estándar desplegada al cliente

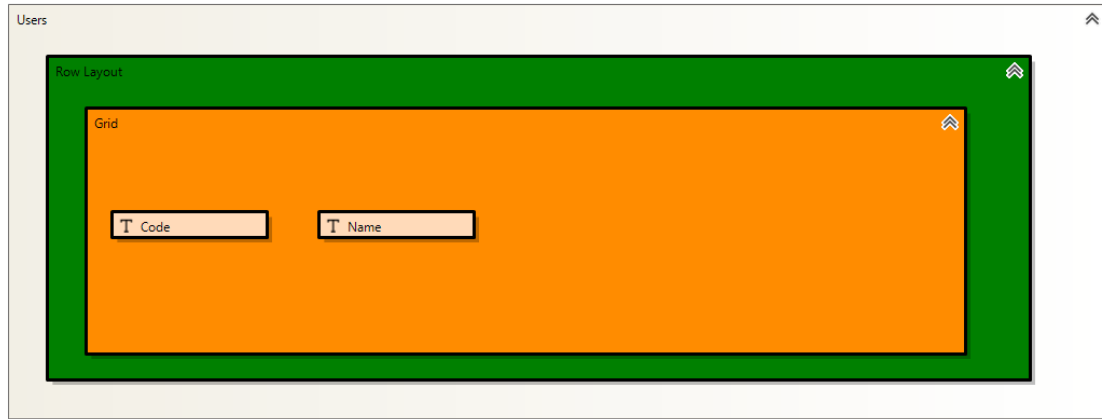
En las figuras 6.1, 6.2 y 6.3 se muestran los modelos de la versión estándar desplegada al cliente inicialmente. El modelo de aplicación se omitirá de aquí en adelante, pues visualmente no muestra los cambios aplicados al dominio.



**Figura 6.1:** Modelo de dominio de la versión estándar desplegada al cliente.



**Figura 6.2:** Modelo de aplicación de la versión estándar desplegada al cliente.



**Figura 6.3:** Modelo de UI de la versión estándar desplegada al cliente.

### 6.0.3. Evolución de la versión estándar

Ahora se aplica la propiedad de la fecha de entrada a la residencia de los usuarios en la versión estándar del producto. Los cambios se pueden ver reflejados en las figuras 6.4 y 6.5.

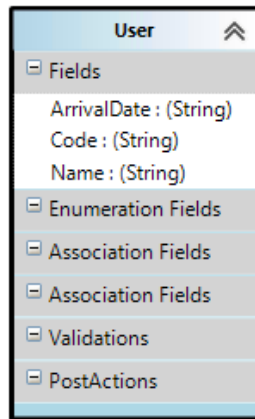


Figura 6.4: Modelo de dominio de la versión estándar evolucionada.

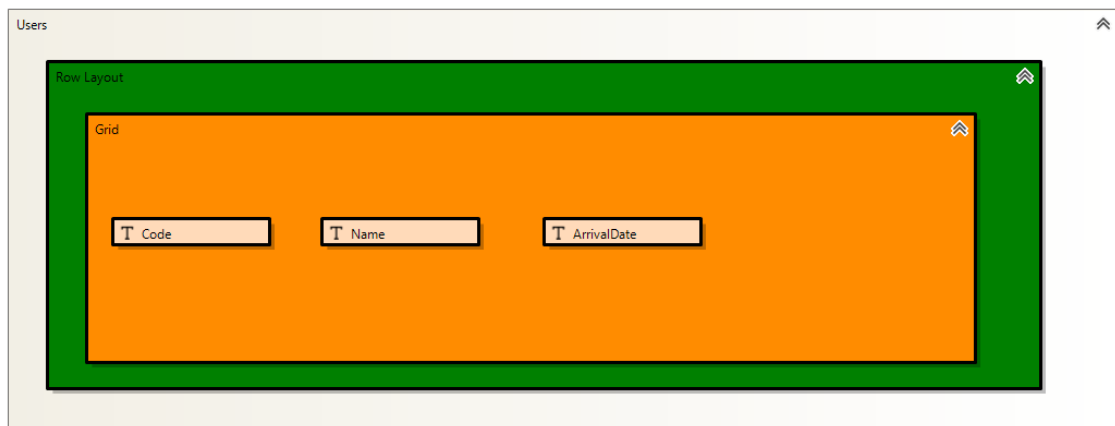


Figura 6.5: Modelo de UI de la versión estándar evolucionada.

## 6.1 Evolución de la versión del cliente

A continuación, se simula que el cliente añada el nombre de un familiar a los usuarios en su propia versión del producto. Los cambios son apreciables en las figuras 6.6 y 6.7.

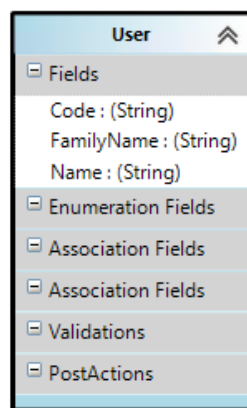


Figura 6.6: Modelo de dominio de la versión de cliente evolucionada.

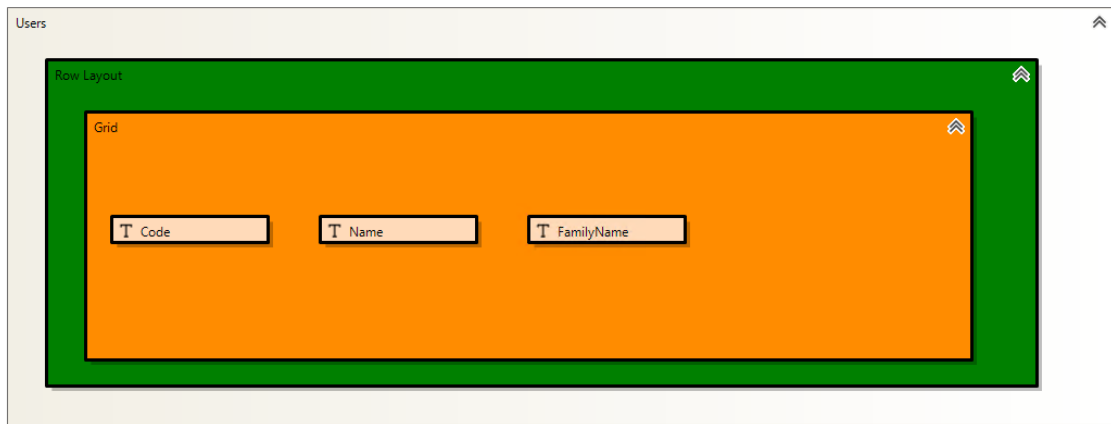


Figura 6.7: Modelo de UI de la versión de cliente evolucionada.

## 6.2 Resultado de la aplicación del algoritmo de mezcla de modelos

Finalmente, se despliega al cliente el algoritmo de mezcla de modelos, haciendo así que su versión del producto posea las mejoras implementadas en los anteriores pasos. Esto es observable en las figuras 6.8 y 6.9.

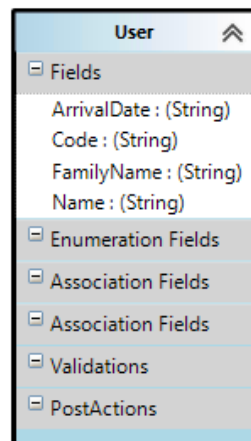
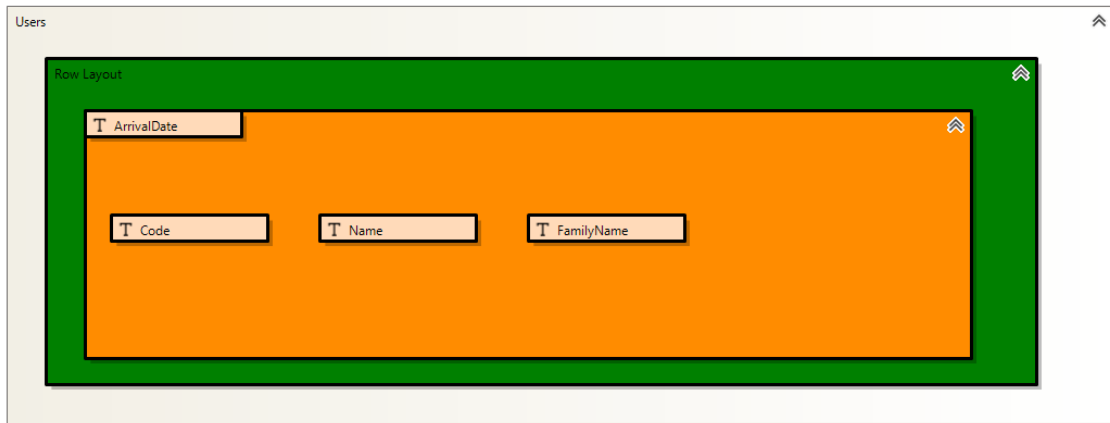


Figura 6.8: Modelo de dominio de la versión final.



**Figura 6.9:** Modelo de UI de la versión final.

Cabe mencionar que el campo traído desde la versión estándar, en la UI está descolocado, pues el algoritmo por el momento no contempla los diagramas visualmente, pues se consideró que no era algo prioritario, ya que el resultado es el mismo.





# Conclusiones y trabajo futuro

---

Consideramos que los objetivos planteados se han cumplido. Se ha desarrollado un algoritmo de mezcla de modelos totalmente funcional, además se han desarrollado una serie de modificaciones sobre las herramientas de desarrollo del departamento.

El proyecto cuenta con una gran cantidad de validaciones y exhaustivas pruebas. Además, se han realizado algunas presentaciones a clientes. Actualmente, se están desarrollando nuevas restricciones debido a la adición de nuevos elementos a las *DSL Tools*.

Este trabajo me ha dado la oportunidad de afrontar importantes desafíos. He podido mejorar mis capacidades con tecnologías de *Microsoft*, especialmente las *DSL Tools*, también el manejo de documentos XML. Además de ser el responsable de un módulo del nuevo producto.

Algunas de las asignaturas que más me han sido de ayuda para la realización del proyecto y la redacción de este trabajo han sido:

- Proceso de *software*.
- Proyecto de ingeniería de *software*.
- Desarrollo de software dirigido por modelos.
- Análisis y especificación de requisitos.
- Calidad de *software*.

Como trabajo futuro y antes de que se comercialice la nueva versión del producto, se tendrán que realizar las siguientes tareas:

- Pruebas de aceptación para verificar más exhaustiva y formalmente los requisitos no funcionales.
- Despliegue de ramas de clientes.
- *Builds* que lancen a ejecución el algoritmo de mezcla de modelos para producir las versiones de cliente.
- Automatizar la generación de código en las versiones de cliente.



# Referencias

---

- [1] *model-driven development (MDD)*. Alexander S. Gillis, 2018 (consultado en 02/2022): <https://searchsoftwarequality.techtarget.com/definition/model-driven-development>
- [2] *15 reasons why you should start using Model Driven Development*. Johaan Den Haan, 2009 (consultado en 02/2022): <http://www.theenterprisearchitect.eu/blog/2009/11/25/15-reasons-why-you-should-start-using-model-driven-development/#:~:text=MDD%20is%20faster&text=While%20the%20used%20model%20is,functionality%20in%20the%20same%20time>.
- [3] *Domain-Specific Development with Visual Studio DSL Tools*. Steve Cook, Gareth Jones, Stuart Kent y Allan Cameron Wills, Addison Wesley, 2017. ISBN: 0321398203
- [4] Documentación oficial System.Text.RegularExpressions.Regex de .NET (consultado en 02/2022): <https://docs.microsoft.com/es-es/dotnet/api/system.text.regularexpressions.regex?view=net-6.0>
- [5] *Microservices*. J. Lewis y M. Fowler, 2014 (consultado en 02/2022): <https://martinfowler.com/articles/microservices.html>
- [6] *Customized Software: What is it, Types, and Examples*. Shardul Bhatt, 2021 (consultado en 02/2022): <https://www.botreetechnologies.com/blog/customized-software-what-is-it-types-and-examples/>
- [7] *S/4HANA Extensibility: Customization, Modifications, Custom Code*. Erich Schneider, 2015 (consultado en 03/2022): <https://blogs.sap.com/2015/07/16/s4hana-extensibility-customization-modifications-custom-code/>
- [8] *Fusion Applications Extensibility Guide for Developers* (consultado en 03/2022): <https://docs.oracle.com/applications/falcm12/OADTC/GUID-7A38E4EA-551C-407D-8E7C-A07AD3887D7A.htm#OADTC152>
- [9] *Extensibilidad* (consultado en 06/2022): <https://hmong.es/wiki/Extendable>
- [10] *Información general sobre la creación y personalización para la versión 9.x de Dynamics 365 Customer Engagement (on-premises)* (consultado en 03/2022): <https://docs.microsoft.com/es-es/dynamics365/customerengagement/on-premises/customize/overview?view=op-9-1>

- [11] *Customizing and Extending Oracle Fusion Applications* (consultado en 03/2022):  
<https://docs.oracle.com/applications/falcm12/OADTC/GUID-7A38E4EA-551C-407D-8E7C-A07AD3887D7A.htm#OADTC152>
- [12] *Página principal de Ekon Health* (consultado en 03/2022):  
<https://www.ekon.es/recursos/ekon-health-gestion-sociosanitaria-y-residencias-con-ekon-health-erp/#>
- [13] *Página principal de Iزارo ERP* (consultado en 03/2022):  
<https://izaroerp.com/>
- [14] *Página principal de Git* (consultado en 03/2022):  
<https://git-scm.com/>
- [15] *Qué es Git, ventajas e inconvenientes y por qué deberías aprenderlo (bien)*. José Manuel Alarcón, 2020 (consultado en 03/2022):  
<https://www.campusmvp.es/recursos/post/que-es-git-ventajas-e-inconvenientes-y-por-que-deberias-aprenderlo-bien.aspx#:~:text=Git%20ofrece%20varias%20ventajas%20frente,operaciones%20de%20control%20muy%20r%C3%A1pidas.>
- [16] *Página principal de .NET* (consultado en 04/2022):  
<https://docs.microsoft.com/es-es/dotnet/>
- [17] *Documentación principal de System.XML* (consultado en 04/2022):  
<https://docs.microsoft.com/es-es/dotnet/api/system.xml?view=net-6.0>
- [18] *Introducción oficial a las DSL Tools* (consultado en 04/2022):  
<https://docs.microsoft.com/es-es/visualstudio/modeling/getting-started-with-domain-specific-languages?view=vs-2022>
- [19] *System and Software Quality Requirements and Evaluation* (consultado en 05/2022):  
<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

---

---

## APÉNDICE A

# Acrónimos empleados

---

- **MDD**: Desarrollo guiado por modelos.
- **DSL**: Lenguaje específico de dominio.
- **DSD**: Desarrollo basado en dominio específico.
- **IDE**: Entorno de desarrollo integrado.
- **API**: Interfaz de programación de aplicaciones.
- **ERP**: Sistema de planificación de recursos empresariales.
- **CRM**: Gestión o administración de relaciones con el cliente.
- **SPRO**: SAP Reference Project Object.
- **SOA**: Arquitectura orientada a servicios.
- **IOT**: Internet de las cosas.
- **WYSIWYG**: Lo que ves es lo que obtienes.
- **UI**: Interfaz de usuario.
- **PYME**: Pequeña y mediana empresa.
- **HIS**: Sistema de información hospitalario.
- **PRM**: Gestión de la relación con los socios.
- **TIC**: Tecnologías de la información y comunicación.
- **XML**: Lenguaje de marcado extensible.
- **HTML**: Lenguaje de marcado para elaboración de páginas web.
- **CSS**: Hojas de estilo en cascada.
- **W3C**: Consorcio de la *World Wide Web*.
- **XSLT**: Transformaciones de hojas de estilo de lenguaje extensible.
- **SDK**: Kit de desarrollo de *software*.



---

---

## APÉNDICE B

# Tablas de casuística

---

En este anexo se añadirán todas las tablas de los tests de las restricciones sobre las *DSL Tools* mencionados en el capítulo de desarrollo.

## B.1 Dominio

---

### B.1.1. *Domain Classes*

Clase: Entity				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir una Entity?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir una Entity?	No	No	

Figura B.1: Tabla de la clase *Entity*

Clase: Enumeration				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir un Enumeration?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir un Enumeration?	No	No	

Figura B.2: Tabla de la clase *Enumeration*

Clase: Field				
Notas:				
	Modelo	Casuística	¿Se permite?	Explicación
	Cliente	En un modelo de cliente, se puede añadir un Field?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un Field?	Si	Si

Figura B.3: Tabla de la clase *Field*

Clase: EnumerationValue				
Notas:				
	Enumerat ion	Casuística	¿Se permite?	Explicación
	Cliente	En un Enumeration de cliente, se puede añadir un EnumerationValue?	Si	Si
	Estándar	En un Enumeration de estándar, se puede añadir un EnumerationValue?	No	No

Figura B.4: Tabla de la clase *EnumerationValue*

Clase: Validation				
Notas:				
	Entity	Casuística	¿Se permite?	Explicación
	Cliente	En una Entity de cliente, se puede añadir una Validation?	Si	Si
	Estándar	En una Entity de estándar, se puede añadir una Validation?	Si	Si

Figura B.5: Tabla de la clase *Validation*



Clase: PostAction				
Notas:				
	Entity	Casuística	¿Se permite?	Explicación
	Cliente	En una Entity de cliente, se puede añadir una PostAction?	Si	Si
	Estándar	En una Entity de estándar, se puede añadir una PostAction?	Si	Si

Figura B.6: Tabla de la clase *PostAction*

Clase: CalculatedPropertyTestCaseCollection					
Notas:					
	Modelo	CalculatedProperty apuntada	Casuística	¿Se permite?	Explicación
	Cliente	Cliente	En un modelo de cliente, se puede añadir un CalculatedPropertyTestCaseCollection que apunta a una CalculatedProperty de cliente?	Si	Si
	Cliente	Estándar	En un modelo de cliente, se puede añadir un CalculatedPropertyTestCaseCollection que apunta a una CalculatedProperty de estándar?	No	No
	Estándar	Cliente	En un modelo de estándar, se puede añadir un CalculatedPropertyTestCaseCollection que apunta a una CalculatedProperty de cliente?	No	No
	Estándar	Estándar	En un modelo de estándar, se puede añadir un CalculatedPropertyTestCaseCollection que apunta a una CalculatedProperty de estándar?	No	No

Figura B.7: Tabla de la clase *CalculatedPropertyTestCaseCollection*

Clase: CalculatedPropertyTestCase				
Notas:				
	CalculatedPropertyTestCaseCollection	Casuística	¿Se permite?	Explicación
	Cliente	En un CalculatedPropertyTestCaseCollection de cliente, se puede añadir un CalculatedPropertyTestCase?	Si	Si
	Estándar	En un CalculatedPropertyTestCaseCollection de estándar, se puede añadir un CalculatedPropertyTestCase?	No	No

Figura B.8: Tabla de la clase *CalculatedPropertyTestCase*

Clase: ValidationTestCaseCollection					
Notas:					
	Modelo	Validation Apuntada	Casuística	¿Se permite?	Explicación
	Cliente	Cliente	En un modelo de cliente, se puede añadir un ValidationTestCaseCollection que apunta a una Validation de cliente?	Si	Si
	Cliente	Estándar	En un modelo de cliente, se puede añadir un ValidationTestCaseCollection que apunta a una Validation de estándar	No	No
	Estándar	Cliente	En un modelo de estándar, se puede añadir un ValidationTestCaseCollection que apunta a una Validation de cliente?	No	No
	Estándar	Estándar	En un modelo de estándar, se puede añadir un ValidationTestCaseCollection que apunta a una Validation de estándar	No	No

Figura B.9: Tabla de la clase *ValidationTestCaseCollection*

Clase: ValidationTestCase					
Notas:					
	ValidationTestCaseCollection	Casuística	¿Se permite?	Explicación	
	Cliente	En un ValidationTestCaseCollection de cliente, se puede añadir un ValidationTestCase?	Si	Si	
	Estándar	En un ValidationTestCaseCollection de estándar, se puede añadir un ValidationTestCase?	No	No	

Figura B.10: Tabla de la clase *ValidationTestCase*

Clase: PostActionTestCaseCollection					
Notas:					
	Modelo	PostAction apuntada	Casuística	¿Se permite?	Explicación
	Cliente	Cliente	En un modelo de cliente, se puede añadir un PostActionTestCaseCollection que apunta a una PostAction de cliente?	Si	Si
	Cliente	Estándar	En un modelo de cliente, se puede añadir un PostActionTestCaseCollection que apunta a una PostAction de estándar	No	No
	Estándar	Cliente	En un modelo de estándar, se puede añadir un PostActionTestCaseCollection que apunta a una PostAction de cliente?	No	No
	Estándar	Estándar	En un modelo de estándar, se puede añadir un PostActionTestCaseCollection que apunta a una PostAction de estándar	No	No

Figura B.11: Tabla de la clase *PostActionTestCaseCollection*

Clase: PostActionTestCase				
Notas:				
	PostActionTestCaseCollection	Casuística	¿Se permite?	Explicación
	Cliente	En un PostActionTestCaseCollection de cliente, se puede añadir un PostActionTestCase?	Si	Si
	Estándar	En un PostActionTestCaseCollection de estándar, se puede añadir un PostActionTestCase?	No	No

Figura B.12: Tabla de la clase *PostActionTestCase*

Clase: TestCaseEntityDatabase				
Notas:				
	TestCase	Casuística	¿Se permite?	Explicación
	Cliente	En un TestCase de cliente, se puede añadir un TestCaseEntityDatabase?	Si	Si
	Estándar	En un TestCase de estándar, se puede añadir un TestCaseEntityDatabase?	No	No

Figura B.13: Tabla de la clase *TestCaseEntityDatabase*

Clase: TestCaseEntityEntryParameter				
Notas:				
	TestCase	Casuística	¿Se permite?	Explicación
	Cliente	En un TestCase de cliente, se puede añadir un TestCaseEntryParameter?	Si	Si
	Estándar	En un TestCase de estándar, se puede añadir un TestCaseEntryParameter?	No	No

Figura B.14: Tabla de la clase *TestCaseEntityEntryParameter*

Clase: ExternalEntityReference					
Notas:					
	Modelo	Entity apuntada	Casuística	¿Se permite?	Explicación
	Cliente	Cliente	En un modelo de cliente, se puede añadir una ExternalEntityReference que apunta a una Entity de cliente?	Si	Si
	Cliente	Estándar	En un modelo de cliente, se puede añadir una ExternalEntityReference que apunta a una Entity de estándar?	Si	Si
	Estándar	Cliente	En un modelo de estándar, se puede añadir una ExternalEntityReference que apunta a una Entity de cliente?	Si	Si
	Estándar	Estándar	En un modelo de estándar, se puede añadir una ExternalEntityReference que apunta a una Entity de estándar?	No	No

Figura B.15: Tabla de la clase *ExternalEntityReference*

Clase: ExternalEnumerationReference					
Notas:					
	Modelo	Enumeration apuntado	Casuística	¿Se permite?	Explicación
	Cliente	Cliente	En un modelo de cliente, se puede añadir una ExternalEnumerationReference que apunta a un Enumeration de cliente?	Si	Si
	Cliente	Estándar	En un modelo de cliente, se puede añadir una ExternalEnumerationReference que apunta a un Enumeration de estándar?	Si	Si
	Estándar	Cliente	En un modelo de estándar, se puede añadir una ExternalEnumerationReference que apunta a un Enumeration de cliente?	Si	Si
	Estándar	Estándar	En un modelo de estándar, se puede añadir una ExternalEnumerationReference que apunta a un Enumeration de estándar?	No	No

Figura B.16: Tabla de la clase *ExternalEnumerationReference*

**B.1.2. Domain Relationships**

Relación/ Association							
Notas:							
	Modelo	Source Entity	Target Entity	(En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar una Source Entity de cliente con una Target Entity de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar una Source Entity de cliente con una Target Entity de estándar?	No existe	No puede haber una Source Entity de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar una Source Entity de estándar con una Target Entity de cliente?	No existe	No puede haber una Target Entity de estándar en un modelo de cliente
	Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar una Source Entity de estándar con una Target Entity de estándar?	No existe	No puede haber una Source Entity de estándar en un modelo de cliente ni una Target Entity de estándar
	Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar una Source Entity de cliente con una Target Entity de cliente?	No existe	No puede haber una Source Entity de cliente en un modelo de cliente, ni una Target Entity de cliente
	Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar una Source Entity de cliente con una Target Entity de estándar?	No existe	No puede haber una Target Entity de cliente en un modelo de estándar
	Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar una Source Entity de estándar con una Target Entity de cliente?	No existe	No puede haber una Source Entity de cliente en un modelo de estándar
	Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar una Source Entity de estándar con una Target Entity de estándar?	No	No

Figura B.17: Primera tabla de la relación Association

Cliente	Cliente	Cliente	Cliente	En un modelo cliente, se puede relacionar una Source Entity de cliente con una Target External Entity de cliente que apunta a una Entity de cliente?	Si	Si
Cliente	Cliente	Cliente	Estándar	En un modelo cliente, se puede relacionar una Source Entity de cliente con una Target External Entity de cliente que apunta a una Entity de estándar?	Si	Si
Cliente	Cliente	Estándar	Cliente	En un modelo cliente, se puede relacionar una Source Entity de cliente con una Target External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una Target External Entity de estándar en un modelo cliente
Cliente	Cliente	Estándar	Estándar	En un modelo cliente, se puede relacionar una Source Entity de cliente con una Target External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una Target External Entity de estándar en un modelo cliente
Cliente	Estándar	Cliente	Cliente	En un modelo cliente, se puede relacionar una Source Entity de estándar con una Target External Entity de cliente que apunta a una Entity de cliente?	No existe	No puede haber una Source Entity de estándar en un modelo de cliente
Cliente	Estándar	Cliente	Estándar	En un modelo cliente, se puede relacionar una Source Entity de estándar con una Target External Entity de cliente que apunta a una Entity de estándar?	No existe	No puede haber una Source Entity de estándar en un modelo de cliente
Cliente	Estándar	Estándar	Cliente	En un modelo cliente, se puede relacionar una Source Entity de estándar con una Target External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una Source Entity de estándar en un modelo de cliente ni una Target External Entity de estándar
Cliente	Estándar	Estándar	Estándar	En un modelo cliente, se puede relacionar una Source Entity de estándar con una Target External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una Source Entity de estándar en un modelo de cliente ni una Target External Entity de estándar
Estándar	Cliente	Cliente	Cliente	En un modelo estándar, se puede relacionar una Source Entity de cliente con una Target External Entity de cliente que apunta a una Entity de cliente?	No existe	No puede haber una Source Entity de cliente en un modelo de estándar
Estándar	Cliente	Cliente	Estándar	En un modelo estándar, se puede relacionar una Source Entity de cliente con una Target External Entity de cliente que apunta a una Entity de estándar?	No existe	No puede haber una Source Entity de cliente en un modelo de estándar
Estándar	Cliente	Estándar	Cliente	En un modelo estándar, se puede relacionar una Source Entity de cliente con una Target External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una Source Entity de cliente en un modelo de estándar ni una Target External Entity que apunte a una Entity de cliente
Estándar	Cliente	Estándar	Estándar	En un modelo estándar, se puede relacionar una Source Entity de cliente con una Target External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una Source Entity de cliente en un modelo de estándar
Estándar	Estándar	Cliente	Cliente	En un modelo estándar, se puede relacionar una Source Entity de estándar con una Target External Entity de cliente que apunta a una Entity de cliente?	Si	Si
Estándar	Estándar	Cliente	Estándar	En un modelo estándar, se puede relacionar una Source Entity de estándar con una Target External Entity de cliente que apunta a una Entity de estándar?	No	No
Estándar	Estándar	Estándar	Cliente	En un modelo estándar, se puede relacionar una Source Entity de estándar con una Target External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una Target External Entity que apunte a una Entity de cliente
Estándar	Estándar	Estándar	Estándar	En un modelo estándar, se puede relacionar una Source Entity de estándar con una Target External Entity de estándar que apunta a una Entity de estándar?	No	No

Figura B.18: Segunda tabla de la relación *Association*

Relación Generalization							
Notas:							
Modelo	SubEntity	Entity	¿En este caso no existe un	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente		En un modelo de cliente, se pueden relacionar una SubEntity de cliente con una Entity de cliente?	Si	Si	
Cliente	Cliente	Estándar		En un modelo de cliente, se pueden relacionar una SubEntity de cliente con una Entity de estándar?	No existe	No puede haber una Entity de estándar en un modelo de cliente	
Cliente	Estándar	Cliente		En un modelo de cliente, se pueden relacionar una SubEntity de estándar con una Entity de cliente?	No existe	No puede haber una SubEntity de estándar en un modelo de cliente	
Cliente	Estándar	Estándar		En un modelo de cliente, se pueden relacionar una SubEntity de estándar con una Entity de estándar?	No existe	No puede haber una SubEntity de estándar en un modelo de cliente ni una Entity de estándar	
Estándar	Cliente	Cliente		En un modelo de estándar, se pueden relacionar una SubEntity de cliente con una Entity de cliente?	No existe	No puede haber una SubEntity de cliente en un modelo de estándar ni una Entity de cliente	
Estándar	Cliente	Estándar		En un modelo de estándar, se pueden relacionar una SubEntity de cliente con una Entity de estándar?	No existe	No puede haber una SubEntity de cliente en un modelo de estándar	
Estándar	Estándar	Cliente		En un modelo de estándar, se pueden relacionar una SubEntity de estándar con una Entity de cliente?	No existe	No puede haber una una Entity de cliente en un modelo de estándar	
Estándar	Estándar	Estándar		En un modelo de estándar, se pueden relacionar una SubEntity de estándar con una Entity de estándar?	No	No	

Figura B.19: Primera tabla de la relación *Generalization*

Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de cliente?	Si	Si
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de estándar?	No	No
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una External Entity de estándar en un modelo de cliente
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de cliente con una External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una External Entity de estándar en un modelo de cliente
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de cliente?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente ni una External Entity de estándar
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se puede relacionar una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de estándar en un modelo de cliente ni una External Entity de estándar
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de cliente?	No existe	No puede haber una ParentEntity de cliente en un modelo de estándar
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se puede relacionar una ParentEntity de cliente con una External Entity de cliente que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de cliente en un modelo de estándar ni una External Entity de estándar que apunte a una Entity de cliente
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se puede relacionar una ParentEntity de cliente con una External Entity de estándar que apunta a una Entity de estándar?	No existe	No puede haber una ParentEntity de cliente en un modelo de estándar
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se puede relacionar una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de	Si	Si
Estándar	Estándar	Cliente	Estándar	una ParentEntity de estándar con una External Entity de cliente que apunta a una Entity de estándar?	No	No
Estándar	Estándar	Estándar	Cliente	una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de cliente?	No existe	No puede haber una External Entity de estándar que apunte a una Entity de cliente
Estándar	Estándar	Estándar	Estándar	una ParentEntity de estándar con una External Entity de estándar que apunta a una Entity de estándar?	No	No

Figura B.20: Segunda tabla de la relación *Generalization*

Relación	Entity	References	Enumeration	Base			
<b>Notas:</b>							
	<b>Modelo</b>	<b>Entity</b>	<b>Enumerati on</b>	<b>(En este caso no existe un .)</b>	<b>Casuística</b>	<b>¿Permiti do?</b>	<b>Explicación</b>
	Cliente	Cliente	Cliente		En un modelo de cliente, se pueden relacionar una Entity de cliente con un Enumeration de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo de cliente, se pueden relacionar una Entity de cliente con un Enumeration de estándar?	No existe	No puede haber un Enumeration de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un modelo de cliente, se pueden relacionar una Entity de estándar con un Enumeration de cliente?	No existe	No puede haber una Entity de estándar en un modelo de cliente
	Cliente	Estándar	Estándar		En un modelo de cliente, se pueden relacionar una Entity de estándar con un Enumeration de estándar?	No existe	No puede haber una Entity de estándar en un modelo de cliente ni un Enumeration estándar
	Estándar	Cliente	Cliente		En un modelo de estándar, se pueden relacionar una Entity de cliente con un Enumeration de cliente?	No existe	No puede haber una Entity de cliente en un modelo de estándar ni un Enumeration de cliente
	Estándar	Cliente	Estándar		En un modelo de estándar, se pueden relacionar una Entity de cliente con un Enumeration de estándar?	No existe	No puede haber una Entity de cliente en un modelo de estándar
	Estándar	Estándar	Cliente		En un modelo de estándar, se pueden relacionar una Entity de estándar con un Enumeration de cliente?	No existe	No puede haber un Enumeration de cliente en un modelo de estándar
	Estándar	Estándar	Estándar		En un modelo de estándar, se pueden relacionar una Entity de estándar con un Enumeration de estándar?	No	No

Figura B.21: Primera tabla de la relación *EntityReferencesEnumerationBases*



Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se pueden relacionar una Entity de cliente con un External Enumeration de cliente que apunta a un Enumeration de cliente?	Si	Si
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se pueden relacionar una Entity de cliente con un External Enumeration de cliente que apunta a un Enumeration de estándar?	Si	Si
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se pueden relacionar una Entity de cliente con un External Enumeration de estándar que apunta a un Enumeration de cliente?	No existe	No puede haber un External Enumeration de estándar en un modelo de cliente
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se pueden relacionar una Entity de cliente con un External Enumeration de estándar que apunta a un Enumeration de estándar?	No existe	No puede haber un External Enumeration de estándar en un modelo de cliente
Cliente	Estándar	Cliente	Cliente	Entity de estándar con un External Enumeration de cliente que apunta a un Enumeration de cliente?	No existe	No puede haber una Entity de estándar en un modelo de cliente
Cliente	Estándar	Cliente	Estándar	Entity de estándar con un External Enumeration de cliente que apunta a un Enumeration de estándar?	No existe	No puede haber una Entity de estándar en un modelo de cliente
Cliente	Estándar	Estándar	Cliente	Entity de estándar con un External Enumeration de estándar que apunta a un Enumeration de cliente?	No existe	No puede haber una Entity de estándar en un modelo de cliente ni un External Enumeration de estándar
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se pueden relacionar una Entity de estándar con un External Enumeration de estándar que apunta a un Enumeration de estándar?	No existe	No puede haber una Entity de estándar en un modelo de cliente ni un External Enumeration de estándar
Estándar	Cliente	Cliente	Cliente	una Entity de cliente con un External Enumeration de cliente que apunta a un Enumeration de cliente?	No existe	No puede haber una Entity de cliente en un modelo de estándar
Estándar	Cliente	Cliente	Estándar	una Entity de cliente con un External Enumeration de cliente que apunta a un Enumeration de estándar?	No existe	No puede haber una Entity de cliente en un modelo de estándar
Estándar	Cliente	Estándar	Cliente	una Entity de cliente con un External Enumeration de estándar que apunta a un Enumeration de cliente?	No existe	No puede haber una Entity de cliente en un modelo de estándar ni un External Enumeration de estándar
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se pueden relacionar una Entity de cliente con un External Enumeration de estándar que apunta a un Enumeration de estándar?	No existe	No puede haber una Entity de cliente en un modelo de estándar
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se pueden relacionar una Entity de estándar con un External Enumeration de cliente que apunta a un Enumeration de cliente?	Si	Si
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se pueden relacionar una Entity de estándar con un External Enumeration de cliente que apunta a un Enumeration de estándar?	No	No
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se pueden relacionar una Entity de estándar con un External Enumeration de estándar que apunta a un Enumeration de cliente?	No existe	No puede haber un External Enumeration de estándar que apunte a un Enumeration de cliente
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se pueden relacionar una Entity de estándar con un External Enumeration de estándar que apunta a un Enumeration de estándar?	No	No

Figura B.22: Segunda tabla de la relación *EntityReferencesEnumerationBases*

## B.2 Aplicación

### B.2.1. Domain Classes

Clase: DefaultDTO				
Notas: Se elimina lo relacionado con la validación de merge en modelo estándar o cliente pues se hace				
	Entidad apuntada	Casuística	¿Se permite?	Explicación
	Cliente	Se puede añadir un DefaultDTO que apunta a una Entidad de cliente?	Si	Si
	Estándar	Se puede añadir un DefaultDTO que apunta a una Entidad de estándar?	No	No

Figura B.23: Tabla de la clase *DefaultDTO*

Clase: CalculatedDTO				
Notas: Se elimina lo relacionado con la validación de merge en modelo estándar o cliente pues se hace				
	Tiene al menos un campo de cliente	Casuística	¿Se permite?	Explicación
	Si	Se puede añadir un CalculatedDTO que tiene al menos un campo calculado de cliente?	Si	Si
	No	Se puede añadir un CalculatedDTO que no tiene ningún campo calculado de cliente?	No	No

Figura B.24: Tabla de la clase *SimpleDTO*

Clase: ExternalDtoBaseReference					
Notas:					
	Modelo	DTO apuntado	Casuística	¿Se permite?	Explicación
	Cliente	Cliente	En un modelo de cliente, se puede añadir un ExternalDtoBaseReference que apunta a un DtoBase de cliente?	Si	Si
	Cliente	Estándar	En un modelo de cliente, se puede añadir un ExternalDtoBaseReference que apunta a un DtoBase de estándar?	Si	Si
	Estándar	Cliente	En un modelo de estándar, se puede añadir un ExternalDtoBaseReference que apunta a un DtoBase de cliente?	Si	Si
	Estándar	Estándar	En un modelo de estándar, se puede añadir un ExternalDtoBaseReference que apunta a un DtoBase de estándar?	No	No

Figura B.25: Tabla de la clase *ExternalDtoBase*

Clase: <b>ParameterDTO</b>				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir un ParameterDTO?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir un ParameterDTO?	No	No	

**Figura B.26:** Tabla de la clase *ParameterDTO*

Clase: <b>ExternalParameterDtoReference</b>				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir un ExternalParameterDtoReference?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir un ExternalParameterDtoReference?	No	No	

**Figura B.27:** Tabla de la clase *ExternalParameterDtoReference*

Clase: <b>PrivateAdHocAction</b>				
Notas:				
PrivateAdHocActionContainer	Casuística	¿Se permite?	Explicación	
Cliente	En un PrivateAdHocActionContainer de cliente, se puede añadir una PrivateAdHocAction?	Si	Si	
Estándar	En un PrivateAdHocActionContainer de estándar, se puede añadir una PrivateAdHocAction?	No	No	

**Figura B.28:** Tabla de la clase *PrivateAdHocAction*

Clase: <b>PrivateAdHocActionContainer</b>				
Notas:				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un PrivateAdHocActionContainer ?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un PrivateAdHocActionContainer ?	No	No

**Figura B.29:** Tabla de la clase *PrivateAdHocActionContainer*

Clase: <b>PublicAdHocAction</b>				
Notas:				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir una PublicAdHocAction?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir una PublicAdHocAction?	No	No

**Figura B.30:** Tabla de la clase *PublicAdHocAction*

**B.2.2. Domain Relationships**

Relación: <b>AdHocActionReferencesInputParameterDTO / AdHocActionReferencesOutputParameterDTO / AdHocActionReferencesInputDtoBase / AdHocActionReferencesOutputDtoBase</b>						
Notas: Cuando se habla de DTO en este contexto, se trata de tanto DTOBase como ParameterDTO.						
Modelo	AdHocAction	DTO	(En este caso no existe un tercer)	Casística	¿Permitido?	Explicación
Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un DTO de cliente?	Si	Si
Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un DTO de estándar?	No existe	No puede haber un DTO de estándar en un modelo de cliente.
Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un DTO de cliente?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente.
Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un DTO de estándar?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente, ni un DTO de estándar.
Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un DTO de cliente?	No existe	No puede haber un DTO de cliente en un modelo estándar.
Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un DTO de estándar?	No existe	No puede haber una AdHocAction de cliente en un modelo estándar.
Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un DTO de cliente?	No existe	No puede haber un DTO de cliente en un modelo estándar.
Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un DTO de estándar?	No	No

**Figura B.31:** Tabla de las relaciones *AdHocActionReferencesInputParameterDTO*, *AdHocActionReferencesOutputParameterDTO*, *AdHocActionReferencesInputDtoBase* y *AdHocActionReferencesOutputDtoBase*

Relación	AdHocActionReferencesOutputExternalDtoBaseReference / AdHocActionReferencesInputExternalDtoBaseReference						
Modelo	AdHocAction	ExternalDTObase	DTObase apuntado	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de cliente que apunta a un DTO de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de cliente que apunta a un DTO de estándar?	Si	Si	
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de estándar que apunta a un DTO de cliente?	No existe	No puede haber un ExternalDTO de estándar en un modelo de cliente.	
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de estándar que apunta a un DTO de estándar?	No existe	No puede haber un ExternalDTO de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de cliente que apunta a un DTO de cliente?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de cliente que apunta a un DTO de estándar?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de estándar que apunta a un DTO de cliente?	No existe	No puede haber un ExternalDTO de estándar en un modelo de cliente, ni una AdHocAction de estándar.	
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de estándar que apunta a un DTO de estándar?	No existe	No puede haber un ExternalDTO de estándar en un modelo de cliente, ni una AdHocAction de estándar.	
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de cliente que apunta a un DTO de cliente?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar.	
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de cliente que apunta a un DTO de estándar?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar, ni un ExternalDTObase de cliente.	
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de estándar que apunta a un DTO de cliente?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar, ni un ExternalDTObase de cliente.	
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalDTObase de estándar que apunta a un DTO de estándar?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar.	
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de cliente que apunta a un DTO de cliente?	Si	Si	
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de cliente que apunta a un DTO de estándar?	No existe	No puede haber un ExternalDTObase de cliente que apunte a un DTObase de estándar en un modelo estándar.	
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de estándar que apunta a un DTO de cliente?	No existe	No puede haber un ExternalDTObase de estándar que apunte a un DTObase de cliente.	
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalDTObase de estándar que apunta a un DTO de estándar?	No	No	

Figura B.32: Tabla de las relaciones AdHocActionReferencesOutputExternalDtoBaseReference y AdHocActionReferencesInputExternalDtoBaseReference

Relación <i>AdHocActionReferencesOutputExternalParameterDTO / AdHocActionReferencesInputExternalParameterDTO</i>							
<b>Notas:</b> <i>Cuando se habla de DTO en este contexto, se habla de ParameterDTO.</i>							
Modelo	AdHocAction	ExternalParameterDTO	ParameterDTO asociado	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de cliente que apunta a un ParameterDTO de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de cliente que apunta a un ParameterDTO de estándar ?	Si	Si	
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de cliente?	No existe	No puede haber un ExternalParameterDTO de estándar en un modelo de cliente	
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un ExternalParameterDTO de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un ExternalParameterDTO de cliente que apunta a un ParameterDTO de cliente?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de cliente?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente, ni un ExternalParameterDTO de estándar	
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se puede relacionar una AdHocAction de estándar con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de estándar ?	No existe	No puede haber una AdHocAction de estándar en un modelo de cliente, ni un ExternalParameterDTO de estándar	
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de cliente que apunta a un ParameterDTO de cliente?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar, ni un ExternalParameterDTO de cliente	
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de cliente que apunta a un ParameterDTO de estándar ?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar, ni un ExternalParameterDTO de cliente	
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de cliente?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar, ni un ExternalParameterDTO de estándar que apunta a un ParameterDTO de cliente	
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de cliente con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de estándar ?	No existe	No puede haber una AdHocAction de cliente en un modelo de estándar	
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalParameterDTO de cliente que apunta a un ParameterDTO de cliente?	No existe	No puede haber un ExternalParameterDTO de cliente en un modelo de estándar	
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalParameterDTO de cliente que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un ExternalParameterDTO de cliente en un modelo de estándar	
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de cliente?	No existe	No puede haber un ExternalParameterDTO de estándar que apunta a un ParameterDTO de cliente	
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se puede relacionar una AdHocAction de estándar con un ExternalParameterDTO de estándar que apunta a un ParameterDTO de estándar ?	No	No	

Figura B.33: Tabla de las relaciones *AdHocActionReferencesOutputExternalParameterDTO* y *AdHocActionReferencesInputExternalParameterDTO*

Relación <i>HierarchicalDTOTReferencesDTO</i>							
<b>Notas:</b> <i>Cuando se habla de DTO en este contexto, se trata de DTOBase.</i>							
Modelo	HierarchicalDTOT	DTOT	(EN ESTE CASO NO existe un tercer DTO)	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar un HierarchicalDTOT de cliente con un DTOT de cliente?	Si	Si	
Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar un HierarchicalDTOT de cliente con un DTOT de estándar ?	No existe	No puede haber un DTOT de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar un HierarchicalDTOT de estándar con un DTOT de cliente?	No existe	No puede haber un HierarchicalDTOT de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar un HierarchicalDTOT de estándar con un DTOT de estándar ?	No existe	No puede haber un HierarchicalDTOT de estándar en un modelo de cliente, ni un DTOT de estándar.	
Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar un HierarchicalDTOT de cliente con un DTOT de cliente?	No existe	No puede haber un HierarchicalDTOT de estándar en un modelo de cliente, ni un DTOT de estándar.	
Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar un HierarchicalDTOT de cliente con un DTOT de estándar ?	No existe	No puede haber un HierarchicalDTOT de cliente en un modelo de estándar.	
Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar un HierarchicalDTOT de estándar con un DTOT de cliente?	No existe	No puede haber un DTOT de cliente en un modelo de estándar.	
Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar un HierarchicalDTOT de estándar con un DTOT de estándar ?	No	No	

Figura B.34: Tabla de la relación *HierarchicalDTOTReferencesDTOT*

Relación <i>MessageProcessorProcessesParameterDTO</i>							
Notas: <i>Cuando se habla de DTO en este contexto, se trata de ParameterDTO.</i>							
	Modelo	MessageProcessor	ParameterDTO	¿En este caso no existe un tercer elemento?	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar un MessageProcessor de cliente con un ParameterDTO de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar un MessageProcessor de cliente con un ParameterDTO de estándar?	No existe	No puede haber un ParameterDTO de estándar en un modelo de cliente.
	Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar un MessageProcessor de estándar con un ParameterDTO de cliente?	No existe	No puede haber un MessageProcessor de estándar en un modelo de cliente.
	Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar un MessageProcessor de estándar con un ParameterDTO de estándar?	No existe	No puede haber un MessageProcessor de estándar en un modelo de cliente, ni un ParameterDTO de estándar.
	Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar un MessageProcessor de cliente con un ParameterDTO de cliente?	No existe	No puede haber un MessageProcessor de cliente en un modelo de estándar, ni un ParameterDTO de cliente.
	Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar un MessageProcessor de cliente con un ParameterDTO de estándar?	No existe	No puede haber un MessageProcessor de cliente en un modelo de estándar.
	Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar un MessageProcessor de estándar con un ParameterDTO de cliente?	No existe	No puede haber un ParameterDTO de cliente en un modelo de estándar.
	Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar un MessageProcessor de estándar con un ParameterDTO de estándar?	No	No

Figura B.35: Tabla de la relación *MessageProcessorProcessesParameterDTO*

Relación <i>ReportHasCustomLabelGroupBases</i>							
Notas:							
	Modelo	Report	CustomLabelGroup	¿En este caso no existe un tercer elemento?	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo de cliente, se permite relacionar un Report de cliente con un CustomLabelGroup de cliente	Si	Se puede
	Cliente	Cliente	Estándar		En un modelo de cliente, se permite relacionar un Report de cliente con un CustomLabelGroup de estándar	No existe	
	Cliente	Estándar	Cliente		En un modelo de cliente, se permite relacionar un Report de estándar con un CustomLabelGroup de cliente	No existe	
	Cliente	Estándar	Estándar		En un modelo de cliente, se permite relacionar un Report de estándar con un CustomLabelGroup de estándar	No existe	
	Estándar	Cliente	Cliente		En un modelo de estándar, se permite relacionar un Report de cliente con un CustomLabelGroup de cliente	No existe	
	Estándar	Cliente	Estándar		En un modelo de estándar, se permite relacionar un Report de cliente con un CustomLabelGroup de estándar	No existe	
	Estándar	Estándar	Cliente		En un modelo de estándar, se permite relacionar un Report de estándar con un CustomLabelGroup de cliente	No existe	
	Estándar	Estándar	Estándar		En un modelo de estándar, se permite relacionar un Report de estándar con un CustomLabelGroup de estándar	No se permite	Si esto se permitiera, se añadiría a un report de estándar, y cuando se actualizara la plantilla del Report se perderían los cambios del cliente.

Figura B.36: Primera tabla de la relación *ReportHasCustomLabelGroupBases*



Relación ReportHasCustomLabelGroupBases							
Notas:							
Modelo	Report	ExternalCustomLabelGroup	CustomLabelGroup Apatado	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de cliente	Si	Se permite	
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de estándar	Si	Se permite para reaprovechar CustomLabelGroups de estándar en sus Reports	
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de cliente	No existe	No puede haber un ExternalCustomLabelGroup de estándar en un modelo de cliente	
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de estándar	No existe	No puede haber un ExternalCustomLabelGroup de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de cliente	No existe	No puede haber un Report de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de estándar	No existe	No puede haber un ExternalCustomLabelGroup de cliente en un modelo estándar	
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de cliente	No existe	No puede haber un Report de estándar en un modelo de cliente ni un ExternalCustomLabelGroup de estándar	
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de estándar	No existe	No puede haber un Report de estándar en un modelo de cliente ni un ExternalCustomLabelGroup de estándar	
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de cliente	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalCustomLabelGroup	
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de estándar	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalCustomLabelGroup	
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de cliente	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalCustomLabelGroup de estándar	
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar se permite relacionar un Report de cliente con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de estándar	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalCustomLabelGroup	
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar se permite relacionar un Report de estándar con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de cliente	No existe	No puede haber un ExternalCustomLabelGroup de cliente en un modelo estándar	
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar se permite relacionar un Report de estándar con un ExternalCustomLabelGroup de cliente que apunta a un CustomLabelGroup de estándar	No existe	No puede haber un ExternalCustomLabelGroup de cliente en un modelo estándar	
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar se permite relacionar un Report de estándar con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de cliente	No existe	No puede haber un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup	
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar se permite relacionar un Report de estándar con un ExternalCustomLabelGroup de estándar que apunta a un CustomLabelGroup de estándar	No se permite	Si esto se permitiera, se añadiría a un report de estándar, y cuando se actualizara la plantilla del Report se perderían los cambios del cliente.	

Figura B.37: Segunda tabla de la relación ReportHasCustomLabelGroupBases2

Relación	ReportHasInputExternalDataDtoBaseReference						
Notas:	<i>Cuando se habla de DTO en este contexto, se habla de DTOBase.</i>						
	Modelo	Report	ExternalDTO	DTO apuntado	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalDTO de cliente que apunta a un DTO de cliente?	Si	Si.
	Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalDTO de estándar que apunta a un DTO de estándar?	Si	Si.
	Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalDTO de estándar que apunta a un DTO de cliente?	No existe	No puede haber un ExternalDTO de estándar en un modelo de cliente.
	Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalDTO de estándar que apunta a un DTO de estándar?	No existe	No puede haber un ExternalDTO de estándar en un modelo de cliente.
	Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalDTO de cliente que apunta a un DTO de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente.
	Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalDTO de cliente que apunta a un DTO de estándar?	No existe	No puede haber un Report de estándar en un modelo de cliente.
	Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalDTO de estándar que apunta a un DTO de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente ni un ExternalDTO de estándar.
	Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalDTO de estándar que apunta a un DTO de estándar?	No existe	No puede haber un Report de estándar en un modelo de cliente ni un ExternalDTO de estándar.
	Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un Report de cliente con un ExternalDTO de cliente que apunta a un DTO de cliente?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalDTO de cliente.
	Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un Report de cliente con un ExternalDTO de cliente que apunta a un DTO de estándar?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalDTO de cliente que apunta a un DTO de estándar.
	Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un Report de cliente con un ExternalDTO de estándar que apunta a un DTO de cliente?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalDTO de estándar que apunta a un DTO de cliente.
	Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un Report de cliente con un ExternalDTO de estándar que apunta a un DTO de estándar?	No existe	No puede haber un Report de cliente en un modelo de estándar.
	Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalDTO de cliente que apunta a un DTO de cliente?	No	No
	Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalDTO de cliente que apunta a un DTO de estándar?	No existe	No puede haber un ExternalDTO de cliente que apunta a un DTO de estándar en un modelo de estándar.
	Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalDTO de estándar que apunta a un DTO de cliente?	No existe	No puede haber un ExternalDTO de estándar que apunta a un DTO de cliente.
	Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalDTO de estándar que apunta a un DTO de estándar?	No	No

Figura B.38: Tabla de la relación *ReportHasInputExternalDataDtoBaseReference*

**Relación ReportHasInputExternalParametersDTO**

**Notas:** *Cuando se habla de DTO en este contexto, se habla de ParameterDTO.*

Modelo	Report	ExternalParameterDTO	ParameterDTO	Casística	¿Permitido?	Explicación
Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de cliente?	Si	Si.
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de estándar ?	Si	Si.
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalParametersDTO de estándar que apunta a un ParameterDTO de cliente?	No existe	No puede haber un ExternalParametersDTO de estándar en un modelo de cliente.
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un Report de cliente con un ExternalParametersDTO de estándar que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un ExternalParametersDTO de estándar en un modelo de cliente.
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente.
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un Report de estándar en un modelo de cliente.
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalParametersDTO de estándar que apunta a un ParameterDTO de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente ni un ExternalParametersDTO de cliente.
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un Report de estándar con un ExternalParametersDTO de estándar que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un Report de estándar en un modelo de cliente ni un ExternalParametersDTO de cliente.
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un Report de cliente con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de cliente?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalParametersDTO de cliente.
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un Report de cliente con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ExternalParametersDTO de cliente.
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de cliente?	No existe	No puede haber un Report de estándar en un modelo de estándar ni un ExternalParametersDTO de cliente.
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un Report de estándar en un modelo de estándar ni un ExternalParametersDTO de cliente.
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de cliente?	No existe	No puede haber un ExternalParametersDTO de cliente en un modelo de estándar.
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalParametersDTO de cliente que apunta a un ParameterDTO de estándar ?	No existe	No puede haber un ExternalParametersDTO de cliente en un modelo de estándar.
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalParametersDTO de estándar que apunta a un ParameterDTO de cliente?	No existe	No se puede modificar el apuntado de un ExternalParametersDTO de estándar por parte del cliente.
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un Report de estándar con un ExternalParametersDTO de estándar que apunta a un ParameterDTO de estándar ?	No	No

Figura B.39: Tabla de la relación ReportHasInputExternalParameterDTO

**Relación ReportHasInputParametersDTO**

**Notas:** *Cuando hablamos de DTO en este contexto, hablamos de ParameterDTO.*

Modelo	Report	ParameterDTO	¿En este caso no existe un tercer elemento?	Casística	¿Permitido?	Explicación
Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar un Report de cliente con un ParameterDTO de cliente?	Si	Si.
Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar un Report de cliente con un ParameterDTO de estándar ?	No existe	No puede haber un ParameterDTO de estándar en un modelo de cliente.
Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar un Report de estándar con un ParameterDTO de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente.
Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar un Report de estándar con un ParameterDTO de estándar ?	No existe	No puede haber un Report de estándar en un modelo de cliente ni un ParameterDTO de estándar.
Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar un Report de cliente con un ParameterDTO de cliente?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un ParameterDTO de cliente.
Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar un Report de cliente con un ParameterDTO de estándar ?	No existe	No puede haber un Report de cliente en un modelo de estándar.
Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar un Report de estándar con un ParameterDTO de cliente?	No existe	No puede haber un ParameterDTO de cliente en un modelo de estándar.
Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar un Report de estándar con un ParameterDTO de estándar ?	No	No

Figura B.40: Tabla de la relación ReportHasInputParameterDTO

Relación		SubReport					
<b>Notas:</b> En este caso, como el Target de la relación puede ser tanto un ExternalReport como un Report, se tienen en cuenta tres casos, Report de cliente/estándar, ExternalReport de cliente que apunta a un Report de cliente/estándar.							
Modelo	Report	SubReport (Report)	(En este caso no existe un tercer componente)	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar un Report de cliente con un SubReport de cliente?	Si	Si	
Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar un Report de cliente con un SubReport de estándar?	No existe	No puede haber un SubReport de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar un Report de estándar con un SubReport de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar un Report de estándar con un SubReport de estándar?	No existe	No puede haber un Report de estándar ni un SubReport de estándar en un modelo de cliente.	
Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar un Report de cliente con un SubReport de cliente?	No existe	No puede haber un Report de cliente en un modelo de estándar, ni un SubReport de cliente.	
Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar un Report de cliente con un SubReport de estándar?	No existe	No puede haber un Report de cliente en un modelo estándar.	
Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar un Report de estándar con un SubReport de cliente?	No existe	No puede haber un SubReport de cliente en un modelo estándar.	
Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar un Report de estándar con un SubReport de estándar?	No	No	

Figura B.41: Primera tabla de la relación SubReport

Modelo	Report	Subreport (ExternalReport de cliente que apunta a un Report de)	(En este caso no existe un tercer componente)	Casística	¿Permitido?	Explicación
Cliente	Cliente	Cliente		En un modelo de cliente, se puede relacionar un Report de cliente con un SubReport de cliente externo que apunta a un Report de cliente?	Si	Si
Cliente	Cliente	Estándar		En un modelo de cliente, se puede relacionar un Report de cliente con un SubReport de cliente externo que apunta a un Report de estándar?	No existe	Si
Cliente	Estándar	Cliente		En un modelo de cliente, se puede relacionar un Report de estándar con un SubReport de cliente externo que apunta a un Report de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente.
Cliente	Estándar	Estándar		En un modelo de cliente, se puede relacionar un Report de estándar con un SubReport de cliente externo que apunta a un Report de estándar?	No existe	No puede haber un Report de estándar en un modelo de cliente.
Estándar	Cliente	Cliente		En un modelo de estándar, se puede relacionar un Report de cliente con un SubReport de cliente externo que apunta a un Report de cliente?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un SubReport de cliente externo que apunte a un Report de cliente.
Estándar	Cliente	Estándar		En un modelo de estándar, se puede relacionar un Report de cliente con un SubReport de cliente externo que apunta a un Report de estándar?	No existe	No puede haber un Report de cliente en un modelo de estándar, ni un SubReport de cliente externo que apunte a un Report de estándar.
Estándar	Estándar	Cliente		En un modelo de estándar, se puede relacionar un Report de estándar con un SubReport de cliente externo que apunta a un Report de cliente?	No existe	No puede haber un SubReport de cliente externo que apunte a un Report de cliente en un modelo de estándar.
Estándar	Estándar	Estándar		En un modelo de estándar, se puede relacionar un Report de estándar con un SubReport de cliente externo que apunta a un Report de estándar?	No existe	No puede haber un SubReport de cliente externo que apunte a un Report de estándar en un modelo de estándar.

Figura B.42: Segunda tabla de la relación SubReport2

Relación ReportHasInputDataDtoBase							
Notas: Cuando hablamos de DTO en este contexto, hablamos de DtoBase							
Modelo	Report	DTO	¿En este caso no existe un tercer elemento?	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente		En un modelo de cliente, se pueden relacionar un Report de cliente y un DTO de cliente?	Si	Si.	
Cliente	Cliente	Estándar		En un modelo de cliente, se pueden relacionar un Report de cliente y un DTO de estándar?	No existe	No puede haber un DTO de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente		En un modelo de cliente, se pueden relacionar un Report de estándar y un DTO de cliente?	No existe	No puede haber un Report de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar		En un modelo de cliente, se pueden relacionar un Report de estándar y un DTO de estándar?	No existe	No puede haber un Report de estándar en un modelo de cliente ni un DTO de estándar.	
Estándar	Cliente	Cliente		En un modelo de estándar, se pueden relacionar un Report de cliente y un DTO de cliente?	No existe	No puede haber un Report de cliente en un modelo de estándar ni un DTO de cliente.	
Estándar	Cliente	Estándar		En un modelo de estándar, se pueden relacionar un Report de cliente y un DTO de estándar?	No existe	No puede haber un Report de cliente en un modelo de estándar.	
Estándar	Estándar	Cliente		En un modelo de estándar, se pueden relacionar un Report de estándar y un DTO de cliente?	No existe	No puede haber un DTO de cliente en un modelo de estándar.	
Estándar	Estándar	Estándar		En un modelo de estándar, se pueden relacionar un Report de estándar y un DTO de estándar?	No	No	

Figura B.43: Tercera tabla de la relación *SubReport3*

## B.3 UI

### B.3.1. Domain Classes

Clase:	Command			
Notas:				
Modelo	Casística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir un Command?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir un Command?	Si	Si	

Figura B.44: Tabla de la clase *Command*

Clase: <b>CommandGroup</b>				
Notas:				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un CommandGroup?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un CommandGroup?	Si	Si

Figura B.45: Tabla de la clase *CommandGroup*

Clase: <b>CrudCommandGroup</b>				
Notas:				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un CrudCommandGroup?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un CrudCommandGroup?	No	No

Figura B.46: Tabla de la clase *CrudCommandGroup*

Clase: <b>CustomControl</b>				
Notas:				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un CustomControl?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un CustomControl?	No	No se puede, ya que no se puede tener el código parcial del cliente en un ensamblado distinto al estándar

Figura B.47: Tabla de la clase *CustomControl*

Clase: Dashboard				
Notas:				
	DashboardContainer	Casuística	¿Se permite?	Explicación
	Cliente	En un modelo de cliente, se puede añadir un Dashboard?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	En un modelo de estándar, se puede añadir un Dashboard?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente

Figura B.48: Tabla de la clase *Dashboard*

Clase: DashboardContainer				
Notas:				
	Workspace	Casuística	¿Se permite?	Explicación
	Cliente	En un modelo de cliente, se puede añadir un DashboardContainer?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	En un modelo de estándar, se puede añadir un DashboardContainer?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente

Figura B.49: Tabla de la clase *DashboardContainer*

Clase: ExternalCommandReference				
Notas:				
	Modelo	Casuística	¿Se permite?	Explicación
	Cliente	En un modelo de cliente, se puede añadir un ExternalCommandReference de cliente?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un ExternalCommandReference de cliente?	Si	Si

Figura B.50: Tabla de la clase *ExternalCommandReference*

Clase: ExternalFormContainerReference					
Notas:					
Modelo	Form apuntado	Casuística	¿Se permite?	Explicación	
Cliente	Cliente	puede añadir un ExternalFormReference que apunta a un Form de cliente?	Si	Si	
Cliente	Estándar	En un modelo de cliente, se puede añadir un ExternalFormReference que apunta a un Form de estándar?	Si	Si	
Estándar	Cliente	se puede añadir un ExternalFormReference que apunta a un Form de cliente?	Si	Si	
Estándar	Estándar	En un modelo de estándar, se puede añadir un ExternalFormReference que apunta a un Form de estándar?	Si	Si	

Figura B.51: Tabla de la clase *ExternalFormContainerReference*

Clase: ExternalFormReference					
Notas:					
Modelo	Form apuntado	Casuística	¿Se permite?	Explicación	
Cliente	Cliente	En un modelo de cliente, se puede añadir un ExternalFormReference que apunta a un Form de cliente?	Si	Si	
Cliente	Estándar	En un modelo de cliente, se puede añadir un ExternalFormReference que apunta a un Form de estándar?	Si	Si	
Estándar	Cliente	En un modelo de estándar, se puede añadir un ExternalFormReference que apunta a un Form de cliente?	Si	Si	
Estándar	Estándar	En un modelo de estándar, se puede añadir un ExternalFormReference que apunta a un Form de estándar?	Si	Si	

Figura B.52: Tabla de la clase *ExternalFormReference*



<b>Clase:</b> ExternalIndicator				
<b>Notas:</b>				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un ExternalIndicator de cliente?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un ExternalIndicator de cliente?	Si	Si

**Figura B.53:** Tabla de la clase *ExternalIndicator*

<b>Clase:</b> Field				
<b>Notas:</b>				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un Field?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un Field?	Si	Si

**Figura B.54:** Tabla de la clase *Field*

<b>Clase:</b> Form				
<b>Notas:</b>				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un Form?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un Form?	No	No

**Figura B.55:** Tabla de la clase *Form*

Clase: FormContainer				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir un FormContainer?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir un FormContainer?	No	No se puede, está ligado al Navigation y un Navigation no se puede añadir	

**Figura B.56:** Tabla de la clase *FormContainer*

Clase: Layout				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir un Layout?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir un Layout?	Si	Si	

**Figura B.57:** Tabla de la clase *Layout*

Clase: Navigation				
Notas:				
Modelo	Casuística	¿Se permite?	Explicación	
Cliente	En un modelo de cliente, se puede añadir un Navigation?	Si	Si	
Estándar	En un modelo de estándar, se puede añadir un Navigation?	No	Podría darse el caso en el que en el estándar se añadiera a la vez un Navigation, y tener dos Navigation no tendría sentido.	

**Figura B.58:** Tabla de la clase *Navigation*

Clase: <b>NavigationOption</b>				
Notas:				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un NavigationOption?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un NavigationOption?	Si	Si

Figura B.59: Tabla de la clase *NavigationOption*

Clase: <b>Selector</b>				
Notas:				
	<b>Modelo</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	En un modelo de cliente, se puede añadir un Selector?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un Selector?	Si	Si

Figura B.60: Tabla de la clase *Selector*

Clase: <b>Table</b>					
Notas:					
	<b>Modelo</b>	<b>CustomDataSource apuntado</b>	<b>Casuística</b>	<b>¿Se permite?</b>	<b>Explicación</b>
	Cliente	Cliente	En un modelo de cliente, se puede añadir un Table con un CustomDataSource de cliente?	Si	Si
	Cliente	Estándar	En un modelo de cliente, se puede añadir un Table con un CustomDataSource de estándar?	Si	Si
	Estándar	Cliente	En un modelo de estándar, se puede añadir un Table con un CustomDataSource de cliente?	Si	Si
	Estándar	Estándar	En un modelo de estándar se puede añadir un Table con un CustomDataSource de estándar?	No	No

Figura B.61: Tabla de la clase *Table*

Clase: <b>Template</b>				
Notas:				
	Modelo	Casuística	¿Se permite?	Explicación
	Cliente	En un modelo de cliente, se puede añadir un Template?	Si	Si
	Estándar	En un modelo de estándar, se puede añadir un Template?	Si	Si

**Figura B.62:** Tabla de la clase *Template*

Clase: <b>Workspace</b>				
Notas:				
	WorkspaceArea	Casuística	¿Se permite?	Explicación
	Cliente	En un modelo de cliente, se puede añadir un Workspace?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	En un modelo de estándar, se puede añadir un Workspace?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente

**Figura B.63:** Tabla de la clase *Workspace*

Clase: <b>WorkspaceArea</b>				
Notas:				
	Workspace	Casuística	¿Se permite?	Explicación
	Cliente	En un Workspace de cliente, se puede añadir un WorkspaceArea?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	En un Workspace de estándar, se puede añadir un WorkspaceArea?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente

**Figura B.64:** Tabla de la clase *WorkspaceArea*

B.3.2. Domain Relationships

Relación/ CommandReferencesExternalFormContainerReference							
Notas:							
Modelo	Command	ExternalFormContainerReference	FormContainer apuntado	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo cliente, se puede relacionar un Command de cliente con un ExternalFormContainerReference de cliente que apunta a un FormContainer de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo cliente, se puede relacionar un Command de cliente con un ExternalFormContainerReference de cliente que apunta a un FormContainer de estándar?	Si	Si	
Cliente	Cliente	Estándar	Cliente	En un modelo cliente, se puede relacionar un Command de cliente con un ExternalFormContainerReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un ExternalFormContainerReference de estándar en un modelo de cliente.	
Cliente	Cliente	Estándar	Estándar	En un modelo cliente, se puede relacionar un Command de cliente con un ExternalFormContainerReference de estándar que apunta a un FormContainer de estándar?	No existe	No puede haber un ExternalFormContainerReference de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Cliente	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormContainerReference de cliente que apunta a un FormContainer de cliente?	No existe	No puede haber un Command de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Estándar	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormContainerReference de cliente que apunta a un Form de estándar?	No existe	No puede haber un Command de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar	Cliente	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormContainerReference de estándar que apunta a un FormContainer de cliente?	No existe	No puede haber un Command de estándar en un modelo de cliente, ni un ExternalFormContainer de estándar.	
Cliente	Estándar	Estándar	Estándar	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormContainerReference de estándar que apunta a un FormContainer de estándar?	No existe	No puede haber un Command de estándar en un modelo de cliente, ni un ExternalFormContainer de estándar.	
Estándar	Cliente	Cliente	Cliente	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormContainerReference de cliente que apunta a un FormContainer de cliente?	Si	Si	
Estándar	Cliente	Cliente	Estándar	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormContainerReference de cliente que apunta a un FormContainer de estándar?	Si	Si	
Estándar	Cliente	Estándar	Cliente	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormContainerReference de estándar que apunta a un FormContainer de cliente?	No existe	No se puede modificar el apuntado de un ExternalFormContainerReference de estándar por parte del cliente.	
Estándar	Cliente	Estándar	Estándar	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormContainerReference de estándar que apunta a un FormContainer de estándar?	Si	Si	
Estándar	Estándar	Cliente	Cliente	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormContainerReference de cliente que apunta a un FormContainer de cliente?	No existe	No se puede modificar el apuntado de un Command de estándar por parte del cliente.	
Estándar	Estándar	Cliente	Estándar	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormContainerReference de cliente que apunta a un FormContainer de estándar?	No existe	No se puede modificar el apuntado de un Command de estándar por parte del cliente.	
Estándar	Estándar	Estándar	Cliente	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormContainerReference de estándar que apunta a un FormContainer de cliente?	No existe	No se puede modificar el apuntado de un Command de estándar por parte del cliente, ni el de un	
Estándar	Estándar	Estándar	Estándar	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormContainerReference de estándar que apunta a un FormContainer de estándar?	No existe	No se puede modificar el apuntado de un Command de estándar por parte del cliente, ni de un	

Figura B.65: Tabla de la relación CommandReferencesExternalFormContainerReference

Relación: <b>CommandReferencesExternalFormReference</b>							
Notas: <i>Anotaciones extra</i>							
Modelo	Command	ExternalFormReference	Form apuntado	Casuística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo cliente, se puede relacionar un Command de cliente con un ExternalFormReference de cliente que apunta a un Form de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo cliente, se puede relacionar un Command de cliente con un ExternalFormReference de cliente que apunta a un Form de estándar?	Si	Si	
Cliente	Cliente	Estándar	Cliente	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un ExternalFormReference de estándar en un modelo de cliente	
Cliente	Cliente	Estándar	Estándar	En un modelo cliente, se puede relacionar un Command de cliente con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No puede haber un ExternalFormReference de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Cliente	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormReference de cliente que apunta a un Form de cliente?	No existe	No puede haber un Command de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Estándar	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormReference de cliente que apunta a un Form de estándar?	No existe	No puede haber un Command de estándar en un modelo de cliente	
Cliente	Estándar	Estándar	Cliente	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un Command de estándar en un modelo de cliente ni un ExternalFormReference	
Cliente	Estándar	Estándar	Estándar	En un modelo cliente, se puede relacionar un Command de estándar con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No puede haber un Command de estándar en un modelo de cliente ni un ExternalFormReference	
Estándar	Cliente	Cliente	Cliente	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormReference de cliente que apunta a un Form de cliente?	Si	Si	
Estándar	Cliente	Cliente	Estándar	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormReference de cliente que apunta a un Form de estándar?	Si	Si	
Estándar	Cliente	Estándar	Cliente	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No se puede modificar el apuntado de un ExternalFormReference estándar por parte del cliente.	
Estándar	Cliente	Estándar	Estándar	En un modelo estándar, se puede relacionar un Command de cliente con un ExternalFormReference de estándar que apunta a un Form de estándar?	Si	Si	
Estándar	Estándar	Cliente	Cliente	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormReference de cliente que apunta a un Form de cliente?	No existe	No se puede modificar el apuntado de un Command estándar por parte del cliente	
Estándar	Estándar	Cliente	Estándar	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormReference de cliente que apunta a un Form de estándar?	No existe	No se puede modificar el apuntado de un Command estándar por parte del cliente	
Estándar	Estándar	Estándar	Cliente	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No se puede modificar el apuntado de un Command estándar por parte del cliente ni de un ExternalFormReference	
Estándar	Estándar	Estándar	Estándar	En un modelo estándar, se puede relacionar un Command de estándar con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No se puede modificar el apuntado de un Command estándar por parte del cliente ni de un ExternalFormReference	

Figura B.66: Tabla de la relación *CommandReferencesExternalFormReference*

Relación:	CommandReferencesTargetForm						
Notas:	.Anotaciones extra						
	Modelo	Command	Form	¿En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se puede relacionar un Command de cliente con un Form de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo cliente, se puede relacionar un Command de cliente con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se puede relacionar un Command de estándar con un Form de cliente?	No existe	No puede haber un Command de estándar en un modelo de cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se puede relacionar un Command de estándar con un Form de estándar?	No existe	No puede haber un Command de estándar en un modelo de cliente ni un Form de estándar
	Estándar	Cliente	Cliente		En un modelo estándar, se puede relacionar un Command de cliente con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo de cliente
	Estándar	Cliente	Estándar		En un modelo estándar, se puede relacionar un Command de cliente con un Form de estándar?	Si	Si
	Estándar	Estándar	Cliente		En un modelo estándar, se puede relacionar un Command de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo de cliente ni se puede cambiar el apuntado de un Command por parte del cliente
	Estándar	Estándar	Estándar		En un modelo estándar, se puede relacionar un Command de estándar con un Form de estándar?	No existe	No se puede cambiar el apuntado de un Command de estándar por parte del cliente

Figura B.67: Tabla de la relación *CommandReferencesTargetForm*

Relación:	CommandReferencesTargetFormContainer						
Notas:	.Anotaciones extra						
	Modelo	Command	FormContainer	¿En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se puede relacionar un Command de cliente con un FormContainer de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo cliente, se puede relacionar un Command de cliente con un FormContainer de estándar?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se puede relacionar un Command de estándar con un FormContainer de cliente?	No existe	No puede haber un Command de estándar en un modelo de cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se puede relacionar un Command de estándar con un FormContainer de estándar?	No existe	No puede haber un Command de estándar en un modelo de cliente ni un FormContainer de estándar
	Estándar	Cliente	Cliente		En un modelo estándar, se puede relacionar un Command de cliente con un FormContainer de cliente?	No existe	No puede haber un FormContainer de cliente en un modelo de estándar
	Estándar	Cliente	Estándar		En un modelo estándar, se puede relacionar un Command de cliente con un FormContainer de estándar?	Si	Si
	Estándar	Estándar	Cliente		En un modelo estándar, se puede relacionar un Command de estándar con un FormContainer de cliente?	No existe	No se puede cambiar el apuntado de un Command de estándar por parte del cliente
	Estándar	Estándar	Estándar		En un modelo estándar, se puede relacionar un Command de estándar con un FormContainer de estándar?	No existe	No se puede cambiar el apuntado de un Command de estándar por parte del cliente

Figura B.68: Tabla de la relación *CommandReferencesTargetFormContainer*

Relación <i>CrudCommandGroupReferencesExternalFormReference</i>							
Modelo	CrudCommandGroup	ExternalFormReference	Form apuntado	Casuística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de cliente que apunta a un Form de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de estándar que apunta a un Form de estándar?	Si	Si	
Cliente	Cliente	Estándar	Cliente	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de estándar que apunta a un Form de cliente?	No existe	No puede haber un <i>ExternalFormReference</i> de estándar en un modelo de cliente	
Cliente	Cliente	Estándar	Estándar	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de estándar que apunta a un Form de estándar?	No existe	No puede haber un <i>ExternalFormReference</i> de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Cliente	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de cliente que apunta a un Form de cliente?	No existe	No puede haber un <i>CrudCommandGroup</i> de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Estándar	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de cliente que apunta a un Form de estándar?	No existe	No puede haber un <i>CrudCommandGroup</i> de estándar en un modelo de cliente	
Cliente	Estándar	Estándar	Cliente	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de estándar que apunta a un Form de cliente?	No existe	No puede haber un <i>CrudCommandGroup</i> de estándar en un modelo de cliente ni un <i>ExternalFormReference</i> de estándar	
Cliente	Estándar	Estándar	Estándar	En un modelo cliente, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de estándar que apunta a un Form de estándar?	No existe	No puede haber un <i>CrudCommandGroup</i> de estándar en un modelo de cliente ni un <i>ExternalFormReference</i> de estándar	
Estándar	Cliente	Cliente	Cliente	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de cliente que apunta a un Form de cliente?	No existe	No puede haber un <i>CrudCommandGroup</i> de cliente en un modelo de estándar	
Estándar	Cliente	Cliente	Estándar	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de cliente que apunta a un Form de estándar?	No existe	No puede haber un <i>CrudCommandGroup</i> de cliente en un modelo de estándar	
Estándar	Cliente	Estándar	Cliente	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de estándar que apunta a un Form de cliente?	No existe	No puede haber un <i>CrudCommandGroup</i> de cliente en un modelo de estándar ni un <i>ExternalFormReference</i> de estándar que apunta a un Form de cliente	
Estándar	Cliente	Estándar	Estándar	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de cliente con un <i>ExternalFormReference</i> de estándar que apunta a un Form de estándar?	No existe	No puede haber un <i>CrudCommandGroup</i> de cliente en un modelo de estándar	
Estándar	Estándar	Cliente	Cliente	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de cliente que apunta a un Form de cliente?	No existe	No se puede cambiar el apuntado de un <i>CrudCommandGroup</i> de estándar por parte del cliente	
Estándar	Estándar	Cliente	Estándar	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de cliente que apunta a un Form de estándar?	No existe	No se puede cambiar el apuntado de un <i>CrudCommandGroup</i> de estándar por parte del cliente	
Estándar	Estándar	Estándar	Cliente	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de estándar que apunta a un Form de cliente?	No existe	No se puede cambiar el apuntado de un <i>CrudCommandGroup</i> de estándar por parte del cliente ni se puede cambiar el apuntado de un <i>ExternalFormReference</i> de estándar por parte del cliente	
Estándar	Estándar	Estándar	Estándar	En un modelo estándar, se puede relacionar un <i>CrudCommandGroup</i> de estándar con un <i>ExternalFormReference</i> de estándar que apunta a un Form de estándar?	No existe	No se puede cambiar el apuntado de un <i>CrudCommandGroup</i> de estándar por parte del cliente	

Figura B.69: Tabla de la relación *CrudCommandGroupReferencesExternalFormReference*



Relación: <i>CrudCommandGroupReferencesTargetForm</i>							
Notas: <i>Anotaciones extra</i>							
	Modelo	CrudCommandGroup	Form	¿En este caso no existe un tercer?	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se puede relacionar un CrudCommandGroup de cliente con un Form de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo cliente, se puede relacionar un CrudCommandGroup de cliente con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se puede relacionar un CrudCommandGroup de estándar con un Form de cliente?	No existe	No puede haber un CrudCommandGroup de estándar en un modelo de cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se puede relacionar un CrudCommandGroup de estándar con un Form de estándar?	No existe	No puede haber un CrudCommandGroup de estándar en un modelo de cliente ni un Form de estándar
	Estándar	Cliente	Cliente		En un modelo estándar, se puede relacionar un CrudCommandGroup de cliente con un Form de cliente?	No existe	No puede haber un CrudCommandGroup de cliente en un modelo de estándar ni un Form de cliente
	Estándar	Cliente	Estándar		En un modelo estándar, se puede relacionar un CrudCommandGroup de cliente con un Form de estándar?	No existe	No puede haber un CrudCommandGroup de cliente en un modelo de estándar
	Estándar	Estándar	Cliente		En un modelo estándar, se puede relacionar un CrudCommandGroup de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo de estándar
	Estándar	Estándar	Estándar		En un modelo estándar, se puede relacionar un CrudCommandGroup de estándar con un Form de estándar?	No existe	No se puede cambiar el apuntado de un CrudCommandGroup de estándar por parte del cliente

Figura B.70: Tabla de la relación *CrudCommandGroupReferencesTargetForm*

Relación: <i>DashboardContainerIncludesDashboards</i>							
Notas:							
	Modelo	Dashboard Container	Dashboard	¿En este caso no existe un tercer?	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo de cliente, se pueden relacionar un DashboardContainer de cliente con un Dashboard de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Cliente	Cliente	Estándar		En un modelo de cliente, se pueden relacionar un DashboardContainer de cliente con un Dashboard de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Cliente	Estándar	Cliente		En un modelo de cliente, se pueden relacionar un DashboardContainer de estándar con un Dashboard de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Cliente	Estándar	Estándar		En un modelo de cliente, se pueden relacionar un DashboardContainer de estándar con un Dashboard de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Cliente	Cliente		En un modelo de estándar, se pueden relacionar un DashboardContainer de cliente con un Dashboard de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Cliente	Estándar		En un modelo de estándar, se pueden relacionar un DashboardContainer de cliente con un Dashboard de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Estándar	Cliente		En un modelo de estándar, se pueden relacionar un DashboardContainer de estándar con un Dashboard de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Estándar	Estándar		En un modelo de estándar, se pueden relacionar un DashboardContainer de estándar con un Dashboard de estándar?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente

Figura B.71: Tabla de la relación *DashboardContainerIncludesDashboards*

Relación: DashboardIncludesExternalIndicator							
Notas:							
Modelo	Dashboard	ExternalIndicator	Indicator apuntado	Casuística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de cliente que apunta a un Indicator de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de cliente que apunta a un Indicator de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de estándar que apunta a un Indicator de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de estándar que apunta a un Indicator de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de cliente que apunta a un Indicator de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de cliente que apunta a un Indicator de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de estándar que apunta a un Indicator de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de estándar que apunta a un Indicator de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de cliente que apunta a un Indicator de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de cliente que apunta a un Indicator de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de estándar que apunta a un Indicator de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un Dashboard de cliente con un ExternalIndicator de estándar que apunta a un Indicator de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de cliente que apunta a un Indicator de cliente?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de cliente que apunta a un Indicator de estándar?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de estándar que apunta a un Indicator de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un Dashboard de estándar con un ExternalIndicator de estándar que apunta a un Indicator de estándar?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente	

Figura B.72: Tabla de la relación *DashboardIncludesExternalIndicator*

Relación: ExternalCommandReferenceReferencesForm							
Notas:							
Modelo	ExternalCommandReference	Command apuntado	Form	Casística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente	
Cliente	Cliente	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de cliente?	No.	No	
Cliente	Cliente	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de cliente?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente	
Cliente	Estándar	Cliente	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de estándar?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente ni un Form de estándar	
Cliente	Estándar	Estándar	Cliente	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de cliente?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente	
Cliente	Estándar	Estándar	Estándar	En un modelo de cliente, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de estándar?	No existe	No puede haber un ExternalCommandReference de estándar en un modelo de cliente ni un Form de estándar	
Estándar	Cliente	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Cliente	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de cliente con un Form de estándar?	Si	Si	
Estándar	Cliente	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Cliente	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de cliente que apunta a un Command de estándar con un Form de estándar?	No	No	
Estándar	Estándar	Cliente	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Estándar	Cliente	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de cliente con un Form de estándar?	No existe	No puede haber un ExternalFormReference de estándar que apunte a un Command de cliente	
Estándar	Estándar	Estándar	Cliente	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar	
Estándar	Estándar	Estándar	Estándar	En un modelo de estándar, se pueden relacionar un ExternalCommandReference de estándar que apunta a un Command de estándar con un Form de estándar?	No	No	

Figura B.73: Tabla de la relación *ExternalCommandReferenceReferencesForm*

Relación: FormContainerReferencesExternalFormReference							
Notas:							
Modelo	FormContainer	ExternalFormReference	Form apuntado	Casuística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo cliente, se puede relacionar un FormContainer de cliente con un ExternalFormReference de cliente que apunta a un Form de cliente?	Si	Si	
Cliente	Cliente	Cliente	Estándar	En un modelo cliente, se puede relacionar un FormContainer de cliente con un ExternalFormReference de cliente que apunta a un Form de estándar?	Si	Si	
Cliente	Cliente	Estándar	Cliente	En un modelo cliente, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un ExternalFormReference de estándar en un modelo de cliente.	
Cliente	Cliente	Estándar	Estándar	En un modelo cliente, se puede relacionar un FormContainer de cliente con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No puede haber un ExternalFormReference de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Cliente	En un modelo cliente, se puede relacionar un FormContainer de estándar con un ExternalFormReference de cliente que apunta a un Form de cliente?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Estándar	En un modelo cliente, se puede relacionar un FormContainer de estándar con un ExternalFormReference de cliente que apunta a un Form de estándar?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar	Cliente	En un modelo cliente, se puede relacionar un FormContainer de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente ni un ExternalFormReference de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar	Estándar	En un modelo cliente, se puede relacionar un FormContainer de estándar con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente ni un ExternalFormReference de estándar en un modelo de cliente.	
Estándar	Cliente	Cliente	Cliente	En un modelo estándar, se puede relacionar un FormContainer de cliente con un ExternalFormReference de cliente que apunta a un Form de cliente?	No existe	No se puede añadir un FormContainer de cliente en un modelo de estándar.	
Estándar	Cliente	Cliente	Estándar	En un modelo estándar, se puede relacionar un FormContainer de cliente con un ExternalFormReference de cliente que apunta a un Form de estándar?	No existe	No se puede añadir un FormContainer de cliente en un modelo de estándar.	
Estándar	Cliente	Estándar	Cliente	En un modelo estándar, se puede relacionar un FormContainer de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No se puede añadir un FormContainer de estándar en un modelo de estándar, ni un ExternalFormReference de cliente en un modelo de estándar.	
Estándar	Cliente	Estándar	Estándar	En un modelo estándar, se puede relacionar un FormContainer de estándar con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No se puede añadir un FormContainer de estándar en un modelo de estándar, ni un ExternalFormReference de estándar en un modelo de estándar.	
Estándar	Estándar	Cliente	Cliente	En un modelo estándar, se puede relacionar un FormContainer de estándar con un ExternalFormReference de cliente que apunta a un Form de cliente?	No existe	No se puede modificar el apuntado de un FormContainer de estándar por parte del cliente.	
Estándar	Estándar	Cliente	Estándar	En un modelo estándar, se puede relacionar un FormContainer de estándar con un ExternalFormReference de cliente que apunta a un Form de estándar?	No existe	No se puede modificar el apuntado de un FormContainer de estándar por parte del cliente.	
Estándar	Estándar	Estándar	Cliente	En un modelo estándar, se puede relacionar un FormContainer de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No se puede modificar el apuntado de un FormContainer de estándar por parte del cliente, ni un ExternalFormReference de cliente en un modelo de estándar.	
Estándar	Estándar	Estándar	Estándar	En un modelo estándar, se puede relacionar un FormContainer de estándar con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No se puede modificar el apuntado de un FormContainer de estándar por parte del cliente.	

Figura B.74: Tabla de la relación FormContainerReferencesExternalFormReference

Relación: <b>FormContainerReferencesForm</b>							
Notas:							
	Modelo	FormContainer	Form	¿En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se puede relacionar un FormContainer de cliente con un Form de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo cliente, se puede relacionar un FormContainer de cliente con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente.
	Cliente	Estándar	Cliente		En un modelo cliente, se puede relacionar un FormContainer de estándar con un Form de cliente?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente.
	Cliente	Estándar	Estándar		En un modelo cliente, se puede relacionar un FormContainer de estándar con un Form de estándar?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente ni un Form de estándar.
	Estándar	Cliente	Cliente		En un modelo estándar, se puede relacionar un FormContainer de cliente con un Form de cliente?	No existe	No puede haber un FormContainer de cliente en un modelo de estándar ni un Form de cliente.
	Estándar	Cliente	Estándar		En un modelo estándar, se puede relacionar un FormContainer de cliente con un Form de estándar?	No existe	No puede haber un FormContainer de cliente en un modelo estándar.
	Estándar	Estándar	Cliente		En un modelo estándar, se puede relacionar un FormContainer de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo estándar.
	Estándar	Estándar	Estándar		En un modelo estándar, se puede relacionar un FormContainer de estándar con un Form de estándar?	No existe	No se puede modificar el apuntado de un FormContainer de estándar por parte del cliente.

Figura B.75: Tabla de la relación *FormContainerReferencesForm*

Relación: <i>NavigationOptionReferencesExternalFormReference</i>							
Notas:							
Modelo	NavigationOption	ExternalFormReference	Form apuntado	Casuística	¿Permitido?	Explicación	
Cliente	Cliente	Cliente	Cliente	En un modelo cliente, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de cliente que apunta a un Form de cliente?	Si	Si.	
Cliente	Cliente	Cliente	Estándar	En un modelo cliente, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de cliente que apunta a un Form de estándar?	Si	Si.	
Cliente	Cliente	Estándar	Cliente	En un modelo cliente, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un ExternalFormReference de estándar en un modelo de cliente.	
Cliente	Cliente	Estándar	Estándar	En un modelo cliente, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No puede haber un ExternalFormReference de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Cliente	En un modelo cliente, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de cliente que apunta a un Form de cliente?	No existe	No puede haber un NavigationOption de estándar en un modelo de cliente.	
Cliente	Estándar	Cliente	Estándar	En un modelo cliente, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de cliente que apunta a un Form de estándar?	No existe	No puede haber un NavigationOption de estándar en un modelo de cliente.	
Cliente	Estándar	Estándar	Cliente	En un modelo cliente, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un NavigationOption de estándar en un modelo de cliente ni un ExternalFormReference de estándar.	
Cliente	Estándar	Estándar	Estándar	En un modelo cliente, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No puede haber un NavigationOption de estándar en un modelo de cliente ni un ExternalFormReference de estándar.	
Estándar	Cliente	Cliente	Cliente	En un modelo estándar, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de cliente que apunta a un Form de cliente?	Si	Si.	
Estándar	Cliente	Cliente	Estándar	En un modelo estándar, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de cliente que apunta a un Form de estándar?	Si	Si.	
Estándar	Cliente	Estándar	Cliente	En un modelo estándar, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un ExternalFormReference de estándar que apunte a un Form de cliente.	
Estándar	Cliente	Estándar	Estándar	En un modelo estándar, se puede relacionar un NavigationOption de cliente con un ExternalFormReference de estándar que apunta a un Form de estándar?	Si	Si (PLUVERT ENLIZA: A asegurar con el futuro cambio del algoritmo relativo al eliminado de clientes).	
Estándar	Estándar	Cliente	Cliente	En un modelo estándar, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de cliente que apunta a un Form de cliente?	No existe	No puede haber un NavigationOption de estándar que apunte a un ExternalFormReference de cliente.	
Estándar	Estándar	Cliente	Estándar	En un modelo estándar, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de cliente que apunta a un Form de estándar?	No existe	No puede haber un NavigationOption de estándar que apunte a un ExternalFormReference de cliente.	
Estándar	Estándar	Estándar	Cliente	En un modelo estándar, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de estándar que apunta a un Form de cliente?	No existe	No puede haber un ExternalFormReference de estándar que apunte a un Form de cliente ni el cliente puede modificarlo.	
Estándar	Estándar	Estándar	Estándar	En un modelo estándar, se puede relacionar un NavigationOption de estándar con un ExternalFormReference de estándar que apunta a un Form de estándar?	No existe	No se puede modificar el apuntado de un NavigationOption de estándar por parte del cliente.	

Figura B.76: Tabla de la relación *NavigationOptionReferencesExternalFormReference*

Relación: <b>NavigationOptionReferencesTargetForm</b>							
Notas:							
	Modelo	Navigation Option	Form	¿En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un navigation de cliente dentro de un form de cliente en un modelo cliente, se puede relacionar un NavigationOption de cliente con un Form de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un navigation de cliente dentro de un form de cliente en un modelo cliente se puede relacionar un NavigationOption de cliente con un Form de estándar?	No existe	No puede haber un Form estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un navigation de cliente dentro de un form de cliente en un modelo cliente se puede relacionar un NavigationOption de estándar con un Form de cliente?	No existe	No puede haber un NavigationOption de estándar en un navigation de cliente
	Cliente	Estándar	Estándar		En un navigation de cliente dentro de un form de cliente en un modelo cliente se puede relacionar un NavigationOption de estándar con un Form de estándar?	No existe	No puede haber un NavigationOption de estándar en un navigation de cliente ni un Form de estándar en un modelo de cliente
	Estándar	Cliente	Cliente		En un navigation estándar dentro de un form estándar en un modelo estándar, se puede relacionar un NavigationOption de cliente con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo de estándar
	Estándar	Cliente	Estándar		En un navigation estándar dentro de un form estándar en un modelo estándar, se puede relacionar un NavigationOption de cliente con un Form de estándar?	Si	Si (ADVERTENCIA: A cubrir con futuro añadido del algoritmo)
	Estándar	Estándar	Cliente		En un navigation estándar dentro de un form estándar en un modelo estándar, se puede relacionar un NavigationOption de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo de estándar
	Estándar	Estándar	Estándar		En un navigation estándar dentro de un form estándar en un modelo estándar, se puede relacionar un NavigationOption de estándar con un Form de estándar?	No existe	No se puede ya que el NavigationOption de estándar ya señalará a un Form estándar

Figura B.77: Tabla de la relación *NavigationOptionReferencesTargetForm*

Relación:		NavigationOptionReferencesTemplate					
Notas:							
	Modelo	Navigation Option	Template	¿En este caso no existe un tercer	Casuística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un navigation de cliente dentro de un form de cliente en un modelo cliente, se puede relacionar un NavigationOption de cliente con un Template de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un navigation de cliente dentro de un form de cliente en un modelo cliente, se puede relacionar un NavigationOption de cliente con un Template de estándar?	Si	No puede haber un Template de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un navigation de cliente dentro de un form de cliente en un modelo cliente, se puede relacionar un NavigationOption de estándar con un Template de cliente?	No existe	No puede haber un NavigationOption de estándar en un Form de cliente en un modelo de cliente
	Cliente	Estándar	Estándar		En un navigation de cliente dentro de un form de cliente en un modelo cliente, se puede relacionar un NavigationOption de estándar con un Template de estándar?	No existe	No puede haber un NavigationOption de estándar en un Navigation de cliente en un Form de cliente en un modelo de cliente ni un Template de estándar
	Estándar	Cliente	Cliente		En un navigation de estándar dentro de un form de estándar en un modelo estándar, se puede relacionar un NavigationOption de cliente con un Template de cliente?	Si	Si
	Estándar	Cliente	Estándar		En un navigation de estándar dentro de un form de estándar en un modelo estándar, se puede relacionar un NavigationOption de cliente con un Template de estándar?	Si	Si
	Estándar	Estándar	Cliente		En un navigation de estándar dentro de un form de estándar en un modelo estándar, se puede relacionar un NavigationOption de estándar con un Template de cliente?	No existe	No se puede modificar el apuntado de un NavigationOption de estándar
	Estándar	Estándar	Estándar		En un navigation de estándar dentro de un form de estándar en un modelo estándar, se puede relacionar un NavigationOption de estándar con un Template de estándar?	No existe	No se puede modificar el apuntado de un NavigationOption de estándar

Figura B.78: Tabla de la relación *NavigationOptionReferencesTemplate*



Relación <i>NavigationOptionReferencesWorkspace</i>							
Notas:							
	Modelo	Navigation Option	Workspace	¿En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se pueden relacionar un <i>NavigationOption</i> de cliente con un <i>Workspace</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Cliente	Cliente	Estándar		En un modelo cliente, se pueden relacionar un <i>NavigationOption</i> de cliente con un <i>Workspace</i> de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se pueden relacionar un <i>NavigationOption</i> de estándar con un <i>Workspace</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se pueden relacionar un <i>NavigationOption</i> de estándar con un <i>Workspace</i> de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Cliente	Cliente		En un modelo estándar se pueden relacionar un <i>NavigationOption</i> de cliente con un <i>Workspace</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Cliente	Estándar		En un modelo estándar, se pueden relacionar un <i>NavigationOption</i> de cliente con un <i>Workspace</i> de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Estándar	Cliente		En un modelo estándar, se pueden relacionar un <i>NavigationOption</i> de estándar con un <i>Workspace</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Estándar	Estándar		En un modelo estándar, se pueden relacionar un <i>NavigationOption</i> de estándar con un <i>Workspace</i> de estándar?	No	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente

Figura B.79: Tabla de la relación *NavigationOptionReferencesWorkspace*

Relación/ <i>NavigationReferencesFormContainer</i>							
Notas:							
	Modelo	Navigation	FormContainer	¿En este caso no existe un tercer	Casuística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un form de cliente en un modelo cliente, se puede relacionar un Navigation de cliente con un FormContainer de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un form de cliente en un modelo cliente, se puede relacionar un Navigation de cliente con un FormContainer de estándar?	No existe	No puede haber un Form Container de estándar en un form de cliente
	Cliente	Estándar	Cliente		En un form de cliente en un modelo cliente, se puede relacionar un Navigation de estándar con un FormContainer de cliente?	No existe	No puede haber un Navigation de estándar en un form de cliente
	Cliente	Estándar	Estándar		En un form de cliente en un modelo cliente, se puede relacionar un Navigation de estándar con un FormContainer de estándar?	No existe	No puede haber un Navigation de estándar ni un Form Container de estándar en un form de cliente
	Estándar	Cliente	Cliente		En un form estándar dentro de un modelo estándar, se puede relacionar un Navigation de cliente con un FormContainer de cliente?	No existe	En un modelo de estándar no se puede añadir un Navigation de cliente
	Estándar	Cliente	Estándar		En un form estándar dentro de un modelo estándar, se puede relacionar un Navigation de estándar con un FormContainer de cliente?	No existe	En un modelo de estándar no se puede añadir un Navigation de cliente
	Estándar	Estándar	Cliente		En un form estándar dentro de un modelo estándar, se puede relacionar un Navigation de estándar con un FormContainer de cliente?	No existe	En un form estándar no puede haber un form container de cliente.
	Estándar	Estándar	Estándar		En un form estándar dentro de un modelo estándar, se puede relacionar un Navigation de estándar con un FormContainer de estándar?	No	No se permite hacer por parte del cliente, ya lo hacemos nosotros.

Figura B.80: Tabla de la relación *NavigationReferencesFormContainer*

Relación/ <i>TableReferencesFormContainer</i>							
Notas:							
	Modelo	Table	FormContainer	¿En este caso no existe un tercer	Casuística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se puede relacionar un Table de cliente con un FormContainer de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo cliente, se puede relacionar un Table de cliente con un FormContainer de estándar?	No existe	No puede haber un FormContainer de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se puede relacionar un Table de estándar con un FormContainer de cliente?	No existe	No puede haber un Table de estándar en un modelo de cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se puede relacionar un Table de estándar con un FormContainer de estándar?	No existe	No puede haber un Table de estándar en un modelo de cliente ni un FormContainer de estándar
	Estándar	Cliente	Cliente		En un modelo estándar, se puede relacionar un Table de cliente con un FormContainer de cliente?	No existe	No puede haber un FormContainer de cliente en un modelo estándar
	Estándar	Cliente	Estándar		En un modelo estándar, se puede relacionar un Table de cliente con un FormContainer de estándar?	Si	Si
	Estándar	Estándar	Cliente		En un modelo estándar, se puede relacionar un Table de estándar con un FormContainer de cliente?	No existe	No puede haber un FormContainer de cliente en un modelo estándar
	Estándar	Estándar	Estándar		En un modelo estándar, se puede relacionar un Table de estándar con un FormContainer de estándar?	No existe	No se puede cambiar el apuntado de un Table de estándar por parte del cliente

Figura B.81: Tabla de la relación *TableReferencesFormContainer*

Relación/ <i>TableReferencesTargetForm</i>							
Notas:							
	Modelo	Table	Form	¿En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se puede relacionar un Table de cliente con un Form de cliente?	Si	Si
	Cliente	Cliente	Estándar		En un modelo cliente, se puede relacionar un Table de cliente con un Form de estándar?	No existe	No puede haber un Form de estándar en un modelo de cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se puede relacionar un Table de estándar con un Form de cliente?	No existe	No puede haber un Table de estándar en un modelo de cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se puede relacionar un Table de estándar con un Form de estándar?	No existe	No puede haber un Table de estándar en un modelo de cliente ni un Form de estándar
	Estándar	Cliente	Cliente		En un modelo estándar, se puede relacionar un Table de cliente con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo de estándar
	Estándar	Cliente	Estándar		En un modelo estándar, se puede relacionar un Table de cliente con un Form de estándar?	Si	Si
	Estándar	Estándar	Cliente		En un modelo estándar, se puede relacionar un Table de estándar con un Form de cliente?	No existe	No puede haber un Form de cliente en un modelo de estándar
	Estándar	Estándar	Estándar		En un modelo estándar, se puede relacionar un Table de estándar con un Form de estándar?	No existe	No se puede cambiar el apuntado de un Table estándar por parte del cliente

Figura B.82: Tabla de la relación *TableReferencesTargetForm*

Relación/ <i>WorkspaceAreaIncludesDashboardContainers</i>							
Notas:							
	Modelo	Workspace Area	Dashboard Containers	¿En este caso no existe un tercer	Casística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se pueden relacionar un WorkspaceArea de cliente y un DashboardContainer de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Cliente	Cliente	Estándar		En un modelo cliente, se pueden relacionar un WorkspaceArea de cliente y un DashboardContainer de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se pueden relacionar un WorkspaceArea de estándar y un DashboardContainer de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se pueden relacionar un WorkspaceArea de estándar y un DashboardContainer de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Cliente	Cliente		En un modelo estándar, se pueden relacionar un WorkspaceArea de cliente y un DashboardContainer de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Cliente	Estándar		En un modelo estándar, se pueden relacionar un WorkspaceArea de cliente y un DashboardContainer de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Estándar	Cliente		En un modelo estándar, se pueden relacionar un WorkspaceArea de estándar y un DashboardContainer de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente
	Estándar	Estándar	Estándar		En un modelo estándar, se pueden relacionar un WorkspaceArea de estándar y un DashboardContainer de estándar?	No	Se bloquean todas las funcionalidades relacionadas con Workspaces en modo cliente

Figura B.83: Tabla de la relación *WorkspaceAreaIncludesDashboardContainers*

Relación: <i>WorkspaceAreaIncludesUIContainerElements</i>							
Notas:							
	Modelo	Workspace Area	UIContainerElement	¿En este caso no existe un tercer	Casuística	¿Permitido?	Explicación
	Cliente	Cliente	Cliente		En un modelo cliente, se puede relacionar un <i>WorkspaceArea</i> de cliente con un <i>UIContainerElement</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Cliente	Cliente	Estándar		En un modelo cliente, se puede relacionar un <i>WorkspaceArea</i> de cliente con un <i>UIContainerElement</i> de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Cliente	Estándar	Cliente		En un modelo cliente, se puede relacionar un <i>WorkspaceArea</i> de estándar con un <i>UIContainerElement</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Cliente	Estándar	Estándar		En un modelo cliente, se puede relacionar un <i>WorkspaceArea</i> de estándar con un <i>UIContainerElement</i> de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Cliente	Cliente		En un modelo estándar, se puede relacionar un <i>WorkspaceArea</i> de cliente con un <i>UIContainerElement</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Cliente	Estándar		En un modelo estándar, se puede relacionar un <i>WorkspaceArea</i> de cliente con un <i>UIContainerElement</i> de estándar?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Estándar	Cliente		En un modelo estándar, se puede relacionar un <i>WorkspaceArea</i> de estándar con un <i>UIContainerElement</i> de cliente?	No existe	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente
	Estándar	Estándar	Estándar		En un modelo estándar, se puede relacionar un <i>WorkspaceArea</i> de estándar con un <i>UIContainerElement</i> de estándar?	No	Se bloquean todas las funcionalidades relacionadas con <i>Workspaces</i> en modo cliente

Figura B.84: Tabla de la relación *WorkspaceAreaIncludesUIContainerElements*

---

## APÉNDICE C

# Objetivos de desarrollo sostenible

---

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de desarrollo sostenibles	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos				X

### Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

El trabajo realizado ofrece una importante repercusión en lo que a nivel social respecta. Recordemos que el contexto en el cual es realizado se trata de un ERP de residencias de ancianos y centros de día.

El primer y más destacable punto es el relacionado con **salud y bienestar**, el cual se puede ver fácilmente reflejado, ya que el proyecto forma parte de un ERP que afecta a centros sociales de grupos vulnerables.

El objetivo principal del producto relacionado con el trabajo realizado es facilitar en la mayor medida posible las tareas de los trabajadores de las mencionadas residencias y centros. Por lo cual, este trabajo contribuye a que se le pueda brindar el mejor servicio posible a los asistentes, promoviendo así su **salud y bienestar**.

La empresa en la que el alumno realiza el trabajo lleva años demostrando un compromiso con los ancianos de este país, y tratan de mostrarse siempre lo más cercano posibles, aprendiendo más día a día, acudiendo a eventos y convenciones promovidos por las propias residencias.

Esto se puede destacar aún más teniendo en cuenta que el proyecto abre las puertas a la personalización del software por parte del cliente. Gracias a este proyecto, el personal de los centros podrá tener la mejor y más adecuada versión del producto.

El segundo y último ODS para mencionar es **industria, innovación e infraestructuras**, esta relación se puede atribuir principalmente a que el trabajo se realiza en un departamento de I+D+i.

Este ambiente promueve el avance de la industria de la informática, siendo que el departamento en el cual es realizado el trabajo utiliza tecnologías punteras como son el desarrollo guiado por modelos o la generación automática de código.

El mencionado departamento tiene un especial compromiso también con el avance de las tecnologías software, demostrándolo con los desafíos que afronta al usar las tecnologías mencionadas, sobre todo teniendo en cuenta que cuando el desarrollo del producto comenzó, estos métodos aún estaban surgiendo.

El trabajo realizado, además, ofrece una pionera solución al problema de la personalización del software, pues emplea la mezcla de modelos a través de ficheros XML.

En el estudio realizado en uno de los capítulos del trabajo se puede apreciar que ninguna de las soluciones promovidas para la resolución de este problema utiliza métodos similares.

Pese a ser un trabajo, como se ha mencionado, con un importante impacto social, resulta difícil relacionarlo con más puntos de los diecisiete presentados por los objetivos de desarrollo sostenible. Esto es así, pues no se relacionan en especial objetivos relacionados con el medio ambiente y el consumo o algunos puntos más sociales.

Se podría llegar a relacionar algún punto relacionado con ámbitos sociales como la igualdad, analizando el ambiente en la empresa donde el trabajo es realizado, sin embargo, esto estaría vagamente relacionado con el trabajo.