

Modelado Enterprise Architect

Dirección General de Sistemas de Información y Comunicaciones

Áreas de Gobierno Tecnológico de SSII y del Dato



Servicio Andaluz de Salud
Consejería de Sanidad, Presidencia
y Emergencias

Contenidos

- Dirección General de Sistemas de Información y Comunicaciones
 - Áreas de Gobierno Tecnológico de SSII y del Dato
- Cumplimiento normativo
- Histórico de cambios
- Introducción
 - Testing temprano
- Normas de modelado
 - Estructura de carpetas
 - Información del aplicativo
 - Aplicativo Software
 - Definición de requisitos
 - Análisis del Sistema
 - Diseño del sistema
 - Construcción del sistema
 - Nomenclatura, contenido y relaciones
 - Matrices de trazabilidad
 - Versionado y entregas

Resumen



- **Versión:** v01r09
- **Fecha publicación:** 20 de septiembre de 2017
- **Entrada en vigor desde:** 20 de octubre de 2017

- Adaptación a la nueva versión de la plantilla de contenido ya existente
- Aplicativos sin histórico de documentación
- Plantillas y archivos para migración

Cumplimiento normativo



Las normas expuestas son de obligado cumplimiento. La STIC podrá estudiar los casos excepcionales los cuales serán gestionados a través de los responsables del proyecto correspondiente y autorizados por el Área de Gobernanza de la STIC. Asimismo cualquier aspecto no recogido en estas normas deberá regirse en primera instancia por las guías técnicas correspondientes al esquema nacional de seguridad y esquema nacional de interoperabilidad según correspondencia y en su defecto a los marcos normativos y de desarrollo software establecidos por la Junta de Andalucía, debiendo ser puesto de manifiesto ante la STIC.

La STIC se reserva el derecho a la modificación de la norma sin previo aviso, tras lo cual, notificará del cambio a los actores implicados para su adopción inmediata según la planificación de cada proyecto.

En el caso de que algún actor considere conveniente y/o necesario el incumplimiento de alguna de las normas y/o recomendaciones, deberá aportar previamente la correspondiente justificación fehaciente documentada de la solución alternativa propuesta, así como toda aquella documentación que le sea requerida por la STIC para proceder a su validación técnica.

Contacto Dpto: [Oficina de Calidad](#)

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución.

Versión	v01r09	Fecha publicación	20 de septiembre de 2017	Fecha entrada en vigor	20 de octubre de 2017
Alcance					
- Se añade una recomendación acerca de cómo nombrar a las carpetas creadas por el proveedor para evitar confusiones entre dichas carpetas y los subsistemas. - En la carpeta Aplicativo Software se usará también el código ALM del aplicativo para homogeneizar. - Se especifica la obligatoriedad de que los diferentes elementos incluidos en la plantilla aparezcan en algún diagrama. - Se incluyen recomendaciones a la hora del uso de la ingeniería inversa para incluir los modelos de clases y de datos. - Se especifica la obligatoriedad de una nueva entrega asociada a la fase de construcción con el resultado de la ejecución de los casos de prueba por parte del proveedor. - Se aclara como entregar las evidencias asociadas a los casos de prueba de integración. - A la hora de trabajar con el estado N/A se especifica ahora el motivo por el que no aplica.					

Versión	v01r08	Fecha publicación	5 de mayo de 2017	Fecha entrada en vigor	05 de junio de 2017
Alcance					
Versión BETA distribuida por la OCA de manera específica para los proyectos DBS (Módulo de Datos Básicos de Salud) y HSAP					
- Se sube la versión para evitar confusiones con versiones previas a la definitiva de la v01r07 distribuidas antes de su finalización. - Se aclara que puede haber requisitos de información sin tablas asociadas.					

Versión	v01r07	Fecha publicación		Fecha entrada en vigor	
Alcance					
Versión no liberada por la OCA (1 de noviembre de 2016)					
- Se incluyen las constantes para la generación de documentación. - Se especifica que todos los diagramas deben llevar su descripción. - Se modifica el nombre de los apartados relacionados con los modelos de clases y de datos y se aclaran las clases y tablas a incluir. - Se cambia el tipo de diagrama usado para los modelos de proceso de negocio. - Se incluye el catálogo de excepciones. - Se establece una nueva entrega asociada a la construcción. - Se incluyen aclaraciones en la nomenclatura de los diferentes elementos. - Se incluyen aclaraciones en la información relacionada con las relaciones entre los diferentes elementos. - Se simplifica la gestión de estados reduciendo el nº de ellos. - Se indica como especificar los puntos de entrada de las diferentes secuencias y excepciones de los escenarios de los casos de uso y de prueba. - Se amplía la información acerca de cómo incorporar la información acerca de los include y extend en los casos de uso. - Se define la información adicional a incorporar en los casos de uso relacionados con interfaces a través de servicios. - Se indica cómo trabajar con clases de los tipos interface y enumeration. - Se modifica la información relacionada con los casos de prueba de integración predefinidos y se añaden los relacionados con los servicios REST. - Los casos de prueba funcionales pasan a llamarse casos de prueba funcionales de interoperabilidad. - Se incluyen aclaraciones en las instrucciones para el versionado y la entrega.					

- Se incluye un último apartado con instrucciones para pasar la información existente de la anterior versión de la plantilla a la actual.

Versión	v01r06	Fecha publicación	5 de mayo de 2016	Fecha entrada en vigor	05 de mayo de 2016
Alcance					
Primera versión distribuida por la OCA y el área de desarrollo y proyectos y utilizada por la mayoría de proyectos					

Introducción

El objetivo de este texto es establecer un conjunto de normas y recomendaciones de uso de la plantilla de Enterprise Architect definida por el SAS, para realizar las entregas de la documentación de los diferentes aplicativos.

Se pretende describir en detalle la normativa relacionada con el uso de la plantilla de Enterprise Architect para los desarrollos de software de la STIC. Se describe la estructura de carpetas del aplicativo dentro de la plantilla de Enterprise Architect, la nomenclatura de los objetos, sus estereotipos, los campos a rellenar en cada uno de ellos y el versionado de los entregables.

Está dirigida a todos los interesados en los procesos de desarrollo de software de la STIC.

La plantilla de Enterprise Architect se utiliza para modelar toda la información relacionada con la toma de requisitos, el análisis (incluyendo las pruebas) y el diseño (incluyendo la arquitectura) de los aplicativos.

La información recogida en dicha plantilla para cada uno de ellos es revisada para evaluar la calidad del mismo y, al mismo tiempo, puesta a disposición de todos aquellos interesados que necesiten acceder a dicha documentación, centralizando en una misma ubicación toda la documentación funcional relacionada con un aplicativo.

El SAS se asegura así además de estar en posesión de un mínimo imprescindible de conocimientos que le permitan conocer de antemano el impacto y repercusiones de posibles cambios o evolutivos, pudiendo así realizar valoraciones y estimaciones de manera más ajustada y real.

Testing temprano

Una de las actividades que realiza la OCa es la revisión del fichero de modelado (EA) entregado por el proveedor con el objetivo de garantizar la aplicación de la norma a dicho fichero. Existe una relación entre la aplicación y la versión de la normativa que le aplica.

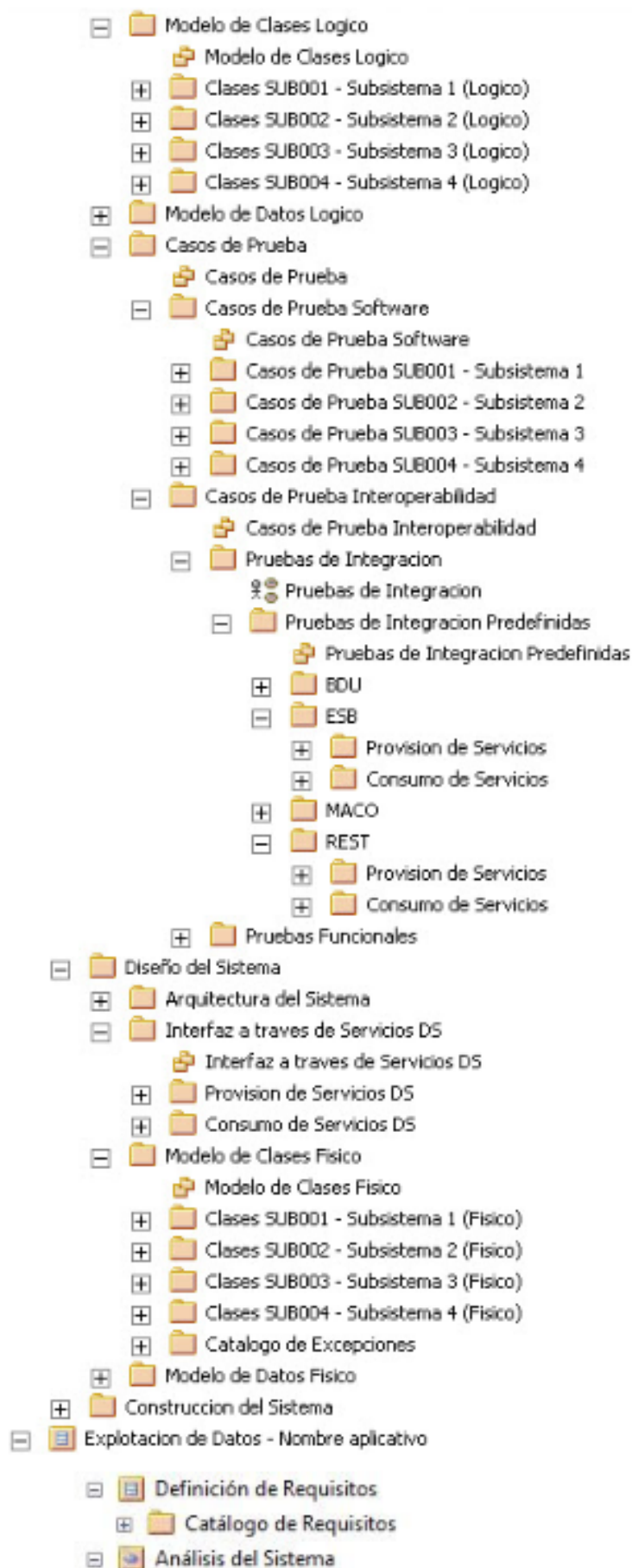
[Más información...](#)

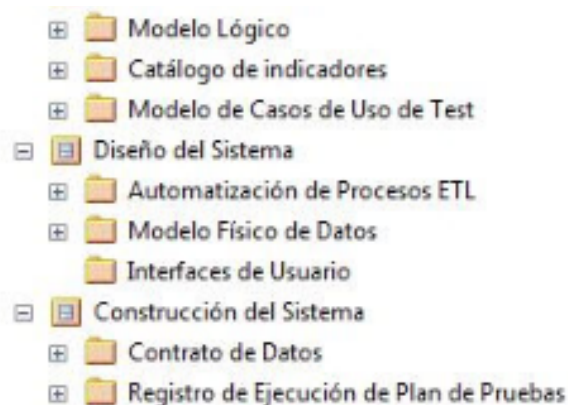
Normas de modelado

Estructura de carpetas

La estructura de carpetas de los aplicativos en Enterprise Architect debe ser:

- [-]  Programa - Nombre programa
 - [-]  Informacion Aplicativo
 - [+]  Participantes
 - [-]  Aplicativo Software - Nombre aplicativo
 - [-]  Definicion de Requisitos
 - [+]  Objetivos del Sistema
 - [-]  Catalogo de Requisitos
 -  Catalogo de Requisitos
 - [+]  Requisitos Funcionales
 - [+]  Requisitos de Integracion
 - [+]  Requisitos No Funcionales
 - [+]  Requisitos de Informacion
 - [-]  Analisis del Sistema
 - [-]  Modelo de Subsistemas
 -  Modelo de Subsistemas
 -  SUB001 - Subsistema 1
 -  SUB002 - Subsistema 2
 -  SUB003 - Subsistema 3
 -  SUB004 - Subsistema 4
 - [-]  Catalogo de Actores
 -  Catalogo de Actores
 - [+]  Actores
 - [+]  Sistemas
 - [-]  Casos de uso
 -  Casos de Uso
 - [+]  Casos de Uso SUB001 - Subsistema 1
 - [+]  Casos de Uso SUB002 - Subsistema 2
 - [+]  Casos de Uso SUB003 - Subsistema 3
 - [+]  Casos de Uso SUB004 - Subsistema 4
 - [-]  Modelo de Proceso de Negocio
 -  Modelo de Proceso de Negocio
 - [-]  Modelo de proceso de Negocio Software
 -  Modelo de proceso de Negocio Software
 - [+]  Modelo de proceso de Negocio SUB001
 - [+]  Modelo de proceso de Negocio SUB002
 - [+]  Modelo de proceso de Negocio SUB003
 - [+]  Modelo de proceso de Negocio SUB004
 - [+]  Modelo de proceso de Negocio Interoperabilidad
 - [-]  Interfaces de Usuario y Navegacion
 -  Diagrama de Navegabilidad
 - [+]  Pantallas
 - [+]  Informes
 - [-]  Interfaz a traves de Servicios
 -  Interfaz a traves de Servicios
 - [+]  Provision de Servicios
 - [+]  Consumo de Servicios





La raíz es el programa, y de ella cuelgan tres ramas, una con información general, una para el aplicativo software y otra para el aplicativo de explotación de datos. De momento cada uno de los aplicativos tendrá su propio entregable, pero de esta forma se posibilita que en el futuro ambos puedan compartir el mismo, incluso compartiendo elementos (por ejemplo de los apartados objetivos, requisitos, modelo de subsistemas, modelo de datos y arquitectura del aplicativo software).

Dentro de cada uno de estas ramas, no todas las carpetas (contenidos en realidad) serán obligatorias para todos los aplicativos, para cada uno de ellos, dependiendo de sus características (complejidad, madurez, tecnología, importancia, etc.) se establecerán aquellas carpetas que deberán cumplimentarse.

Es muy importante no modificar ni eliminar ninguna carpeta de la estructura que ya viene predefinida, pues el hacerlo puede provocar que posteriormente se produzcan fallos en la generación de la documentación asociada a la plantilla o durante la revisión de la misma.

Por otra parte, como se puede apreciar en la anterior figura, aparecen en diferentes apartados carpetas hijas para 4 subsistemas a modo de ejemplo. Para cada aplicativo se establecerán los subsistemas que sean necesarios, pudiéndose dar el caso de que dadas las características del aplicativo (reducido tamaño o complejidad) bastase con una sola carpeta, pudiendo incluso prescindir de dicha carpeta, ubicando los diferentes contenidos directamente en la carpeta raíz de los diferentes apartados. Del mismo modo, en aplicativos de gran tamaño o complejidad, se podrán crear (dentro de las carpetas predefinidas) todas aquellas subcarpetas que sean necesarias, incluso agrupándolas en diferentes niveles de anidamiento. Es conveniente no llamar a esas carpetas siguiendo la nomenclatura usada para los subsistemas, para evitar confusiones, es decir, en lugar de crear una carpeta, por ejemplo dentro de la carpeta "Casos de uso", que se llame "SUB001 - Subsistema 1", llamarla "Casos de uso SUB001 - Subsistema 1", para evitar confusiones con el subsistema en sí.

En las carpetas se han incluido diagramas a modo de ejemplo, donde se pueden apreciar de forma clara las relaciones entre los diferentes elementos contenidos en esa carpeta (y con elementos de otras carpetas). Además de estos diagramas, se podrán incluir todos aquellos que sean necesarios, para reflejar con claridad las relaciones entre los diferentes elementos, tanto en las carpetas presentes en la plantilla, como en las posibles subcarpetas creadas específicamente para el aplicativo. Todos los elementos deben aparecer en al menos uno de los diagramas, debiendo quedar reflejadas claramente para cada uno de los elementos existentes en la plantilla sus relaciones con el resto (ya sea en un solo diagrama o en varios). En todos los diagramas que se incluyan (tanto los que vienen en la plantilla como

los creados para cada aplicativo concreto) se debe especificar una descripción del mismo y de su contenido (en el campo "Notes").

En los siguientes apartados se describen brevemente las diferentes carpetas y el detalle concreto de los diferentes tipos de elementos a incluir en ellas se indica en apartados posteriores.

El nombre de la carpeta en la que están contenidos los siguientes apartados se construirá siguiendo la siguiente nomenclatura:

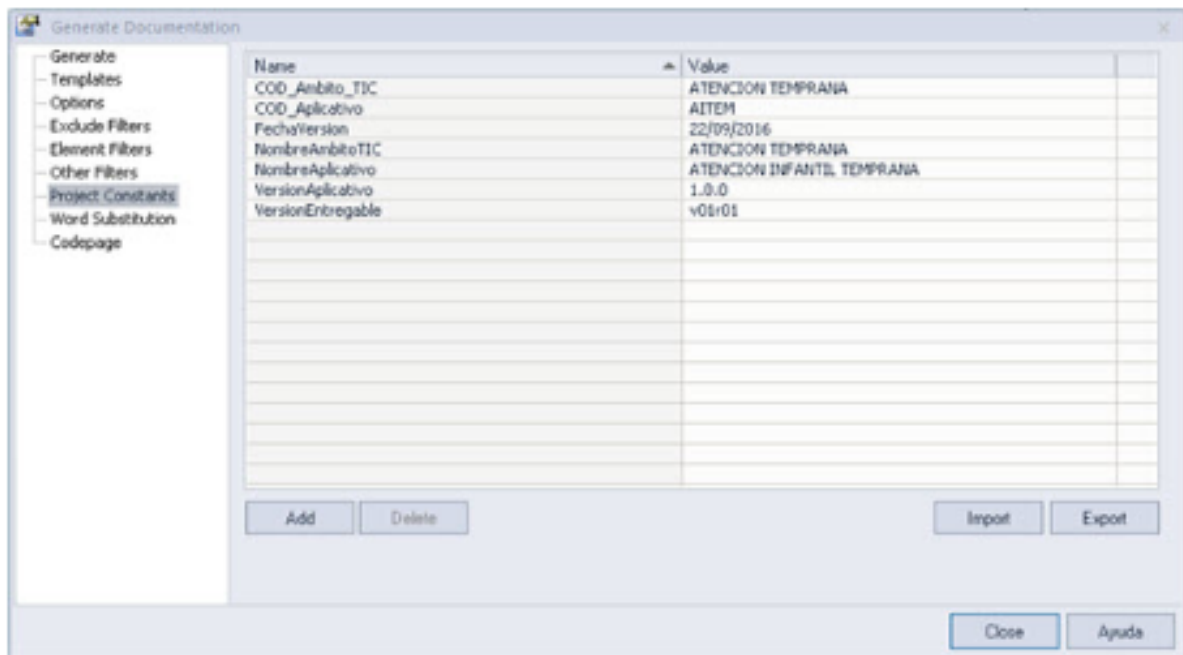
Programa - Nombre programa

Nombre programa = Nombre del programa en la CMS

Información del aplicativo

En esta carpeta irá recogida información genérica del aplicativo:

- **Glosario:** Aunque no pertenece a esta carpeta, ya que es un elemento propio de Enterprise Architect, se hace mención aquí al tratarse de un apartado que contiene información genérica del aplicativo. Cualquier término que se considere que debe aparecer recogido en el glosario del proyecto deberá incluirse en el glosario de Enterprise Architect (dentro del menú "PROJECT" la opción "Glossary"), para que después pueda incorporarse de forma automática a la documentación generada a partir de esta herramienta.
- **Constantes para la generación de la documentación:** Tampoco pertenece a esta carpeta, es otro elemento propio de Enterprise Architect, se hace mención aquí al tratarse de un apartado que contiene información genérica del aplicativo utilizada para la generación de la documentación. Debe indicarse el valor de las constantes incluidas en la plantilla para que después pueda incorporarse de forma automática a la documentación generada a partir de ella (dentro del menú "PACKAGE" la opción "Documentation" y dentro de esta última la opción "Document Generator(TRF/PDF/DOCX)¿" entrando en el apartado "Project Constants"). En la siguiente figura se muestran las constantes a definir (con valores de ejemplo):



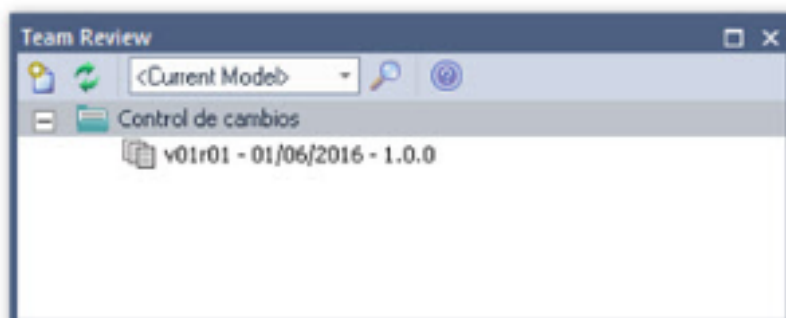
A la hora de introducir la información se deben usar los siguientes valores:

- **COD_Ambito_TIC:** Nombre del programa en la CMS.
- **COD_Aplicativo:** Código Aplicación FARO en la CMS.
- **FechaVersion:** Fecha de la versión del modelado en Enterprise Architect usado para generar la documentación (fecha en la que se crea la baseline correspondiente a la entrega del modelado).
- **NombreAmbitoTIC:** Nombre del programa en la CMS.
- **NombreAplicativo:** Código ALM del aplicativo en la CMS.
- **VersionAplicativo:** Versión del aplicativo.
- **VersionEntregable:** Versión del modelado en Enterprise Architect usado para generar la documentación (versión de la baseline creada para la entrega del modelado).

Si bien el glosario es un apartado opcional, es **obligatorio** especificar el valor de cada una de estas constantes. En el apartado "Versionado y Entregas" se puede encontrar más información acerca de las diferencias entre el versionado del modelado y el versionado de la aplicación.

- **Control de cambios:** Al igual que ocurre con el glosario y las constantes, no pertenece a esta carpeta (se utiliza otro elemento propio de Enterprise Architect), pero también se trata de un apartado que contiene información genérica del aplicativo. En este apartado cada versión del modelado tendrá una entrada en la tabla con el control de cambios. En la siguiente figura se muestra el ejemplo recogido en la plantilla (dentro del menú "PROJECT" la opción "Team Review"):

<u>Versión modelado</u>	<u>Fecha entrega</u>	<u>Versión aplicativo</u>	<u>Cambios</u>
v01r01	01/06/2016	1.0.0	Versión inicial



Para especificar un mayor detalle a la hora de llevar el control de cambios debe crearse una tabla específica para cada una de las versiones del modelado, indicando el detalle de los cambios concretos llevados a cabo. En la siguiente figura se muestra un ejemplo:

<u>Peticionario</u>	<u>Fecha petición</u>	<u>Descripción</u>
Responsable funcional	01/05/2016	Descripción cambio 1
Responsable funcional	02/06/2016	Descripción cambio 2
Responsable del modelado	03/06/2016	Descripción cambio 3



Al igual que ocurre con las para la generación de la documentación, es **obligatorio** completar el control de cambios. Debe además seguirse la nomenclatura empleada en el ejemplo a la hora de nombrar las entradas asociadas a cada una de las entregas. En el apartado "Versionado y Entregas" se puede encontrar más información acerca de las diferencias entre el versionado del modelado y el versionado de la aplicación.

- **Participantes:** Por participantes se entiende el conjunto de personas que de algún modo están relacionados con el aplicativo (descartando el conjunto de usuarios que se representan mediante los actores), básicamente existen tres tipos de participantes:
 - **Responsables del modelado**, entendiendo por modelado el contenido del Enterprise Architect, es decir, la información modelada en los diferentes apartados. Puede haber tantos responsables como sea necesario, tanto del modelado en sí como de la documentación generada a partir del mismo. En la plantilla se ha incluido un ejemplo en el que un mismo participante es responsable del modelado de las carpetas correspondientes a la toma de requisitos, al análisis y al diseño de un aplicativo software, así como de generar la documentación correspondiente a la toma de requisitos. Podría haber un responsable diferente para el modelado de cada una de ellas, o varios responsables para una misma, y lo mismo podría ocurrir en el caso de la documentación. Del mismo modo, se podría detallar aún más definiendo responsables del modelado para cada una de las carpetas o incluso para cada uno de los elementos.
 - **Responsables funcionales**, serían los responsables funcionales de la aplicación, aquellos de cuyos conocimientos y necesidades se obtienen los diferentes requisitos. Estos participantes estarán asociados a los diferentes requisitos a los que han dado origen. En la plantilla se puede ver un ejemplo en el diagrama "Requisitos Funcionales".
 - **Interesados**, que serían todas aquellas personas que deben tener acceso a la documentación asociada a la aplicación. Lógicamente, tanto los responsables de la documentación como los responsables funcionales pueden ser también interesados.

Es **obligatorio** completar el apartado de participantes para los aplicativos de **explotación de datos**, para los aplicativos software es un apartado opcional.

Aplicativo Software

El nombre de la carpeta en la que están contenidos los siguientes apartados se construirá siguiendo la siguiente nomenclatura:

Aplicativo Software - Nombre aplicativo

Nombre aplicativo = Código ALM corto del aplicativo en la CMS

Definición de requisitos

Dentro de este apartado, esencial para el testing temprano y punto de partida para el desarrollo, se cumplimentarán los siguientes apartados:

- **Objetivos del Sistema:** se incluirán aquí los objetivos principales que debe cumplir la aplicación.

- **Catálogo de Requisitos:** esta carpeta contendrá todos los requisitos, tanto funcionales como de cualquier otro tipo, que debe cumplir el sistema para satisfacer los objetivos. Se han dividido en cuatro tipos dependiendo de su naturaleza:
 - **Requisitos Funcionales:** en esta carpeta se ubicarán todos los requisitos funcionales que deba cumplir la aplicación.
 - **Requisitos de Integración:** dado el elevado nivel de interacción entre las diferentes aplicaciones existentes en el SAS, es necesario prestar especial atención a los requisitos de integración, creándose un apartado específico para ellos.
 - **Requisitos No Funcionales:** dentro de esta carpeta se incluirán todos los requisitos no funcionales que deba cumplir el sistema. En caso de ser necesario se podrán dividir o agrupar en todas aquellas carpetas que se estimen necesarias (normativas y estándares, arquitectura, instalación, etc.).
 - **Requisitos de Información:** teniendo en cuenta que en el SAS la ubicación de los diferentes repositorios de información, en la mayor parte de las ocasiones, no está ligada a una aplicación concreta, es necesario prestar especial atención a los requisitos de información. Por este motivo se ha creado un apartado específico para ellos.

Análisis del Sistema

Esta carpeta contendrá todos los elementos referentes al análisis del sistema que se va a desarrollar. A continuación se detallan las carpetas que albergarán a estos elementos:

- **Modelo de Subsistemas:** se incluirán aquí los diferentes subsistemas y las relaciones entre los mismos. Para cada proyecto se establecerán los subsistemas que sean necesarios.
- **Catálogo de Actores:** esta carpeta albergará todos los actores que intervendrán en la aplicación. Se han dividido en dos tipos, cada uno de ellos asociado a una carpeta. En la carpeta **Actores** se incluirán los diferentes tipos de usuarios de la aplicación, así como la relación entre ellos. En la carpeta **Sistemas** se incluirán todos aquellos actores externos que interactúan con la aplicación (otras aplicaciones del SAS, aplicaciones externas, bus, web services proporcionados por otros organismos, etc.)
- **Modelo de Casos de Uso:** se incluirán en este apartado todos los casos de uso necesarios para cubrir los requisitos de la aplicación, creando las carpetas, niveles y diagramas necesarios para una correcta interpretación de los mismos.
- **Modelo de Procesos de Negocio:** en este apartado se incorporará la información acerca de los diferentes procesos de negocio del sistema, representándola mediante diagramas del tipo "Business Process Diagram". Se usarán los elementos de Enterprise Architect destinados a representar esta información, los contenidos en "BPMN 2.0 ¿ Business Process" y en " BPMN 2.0 ¿ Business Process Connectors" (Toolbox). Se crearán las carpetas, niveles y diagramas necesarios para una correcta interpretación de los mismos. La información se dividirá en dos apartados dependiendo del área responsable de cada uno de los procesos:
 - **Modelo de proceso de Negocio Software:** En esta carpeta se incluirá la información relacionada con los procesos asociados con el aplicativo en sí, dejando a un lado los procesos de negocio asociados al

área de Interoperabilidad (lógicamente se hará referencia a ellos, pero el detalle de los mismos irá en la carpeta asociada al área de Interoperabilidad). El contenido de esta carpeta no será revisado por la OCA, ni es obligatorio incorporarlo, pero es conveniente que se añada la información asociada a los diferentes procesos de negocio del sistema para poder disponer de ella, tanto en caso de duda durante la revisión del resto de elementos, como en caso de que algún otro interesado del proyecto necesite consultarla.

Una alternativa a lo anterior es prescindir de este apartado e incluir en los casos de uso la información específica de los procesos de negocio asociados a cada uno de ellos, representándola como en el caso anterior, pero esta vez creando para cada caso de uso su propio diagrama. La elección de una u otra alternativa dependerá de las características del aplicativo, pudiendo incluso utilizarse ambas (ninguna de las dos es obligatoria).

- **Modelo de proceso de Negocio Interoperabilidad:** En esta carpeta se incluirá la información relacionada con los procesos de negocio asociados al área de Interoperabilidad. En este caso sí que es **obligatorio** incluir la información.

Se puede ampliar la información de este apartado incluyendo en los interfaces la información específica de los procesos de negocio asociados a cada uno de ellos, también creando para cada interface su propio diagrama. En cualquier caso deberá cumplimentarse de forma obligatoria el apartado "Modelo de proceso de Negocio Interoperabilidad".

- **Interfaces de Usuario y Navegación:** en este apartado, además de las diferentes pantallas (carpeta **Pantallas**), aparecen recogidos los diferentes informes y listados (documentación en general) generados por la aplicación (carpeta **Informes**). También se debe incluir aquí el diagrama de navegabilidad de la aplicación.

En el caso de que para un aplicativo se haya elaborado una maqueta o prototipo gráfico, con el objetivo por ejemplo de ser validado funcionalmente, dicho prototipo se entregará al SAS junto con el resto de la documentación asociada al proyecto, pero eso no exime de la obligación de incluir la información correspondiente a este apartado.

- **Interfaz a través de servicios:** dada la importancia de la comunicación a través de servicios (ya sea entre aplicaciones o a través del bus) entre las diferentes aplicaciones del SAS este apartado es imprescindible.
 - **Provisión de Servicios:** en esta carpeta se incluirán los interfaces que proporciona la propia aplicación al resto de aplicaciones.
 - **Consumo de Servicios:** en esta carpeta se incluirán los interfaces proporcionados por otras aplicaciones a los que accede la aplicación.
- **Modelo de Clases Lógico:** en el modelo de clases lógico deberían aparecer además de las clases y las relaciones existentes entre ellas, los principales atributos y métodos de las mismas. No es necesario llegar al máximo nivel de detalle, que se incluiría durante el diseño del sistema, pero sí al menos indicar aquellos métodos y atributos relevantes desde el punto de vista funcional.

Tanto en el análisis como en el diseño, no es necesario incluir la totalidad de las clases, lo que se pretende recoger en este apartado es un modelo de clases básico donde se incluyan las principales clases de la aplicación.

Si se opta por incluir el modelo de clases (ya sea durante la fase de análisis o durante la de diseño) partiendo de la información ya existente mediante ingeniería inversa, se recomienda no incluir la totalidad de las clases, sólo las relevantes. Un modelo de clases en el que aparezcan todas, incluso las menos relevantes, puede llegar a dificultar la consulta del mismo. Se debe tener en cuenta que la información recogida en la plantilla no incluye el código en sí del aplicativo, que estará en el repositorio específico del SAS destinado para ello, por lo que no es necesario incluir en la plantilla clases que no aporten información relevante desde el punto de vista funcional.

- **Modelo de Datos Lógico:** si bien en el SAS la ubicación de los datos, los datos mismos y su propio modelado en la mayoría de los casos no están vinculados directamente al aplicativo o no son de su uso exclusivo, es imprescindible conocer el modelado de los mismos desde un punto de vista funcional. Deberían aparecer además de las tablas y las relaciones existentes entre ellas, los principales campos de las mismas. No es necesario llegar al máximo nivel de detalle, que se incluiría durante el diseño del sistema, pero sí al menos indicar aquellos campos relevantes desde el punto de vista funcional.

Tanto en el análisis como en el diseño, no es necesario incluir la totalidad de las tablas, lo que se pretende recoger en este apartado es un modelo de datos básico donde se incluyan las principales tablas de la aplicación.

Deben tenerse en cuenta las mismas consideraciones con respecto a la ingeniería inversa que aparecen en el punto anterior.

El resto de elementos del tipo vista, procedimiento, etc., si son necesarios para la comprensión del modelo de datos básico de la aplicación deberán incluirse, pero no es necesario seguir ningún tipo de normativa. Se usarán los elementos de Enterprise Architect destinados a representar esta información. Esta información no será revisada por la OCA, pero es necesario que se incorpore para poder disponer de ella, tanto en caso de duda durante la revisión del resto de elementos, como en caso de que algún otro interesado del proyecto necesite consultarla.

- **Casos de Prueba:** en este apartado se incorporará la información acerca de los diferentes casos de prueba del sistema. Se crearán las carpetas, niveles y diagramas necesarios para una correcta interpretación de los mismos. La información se dividirá en dos apartados dependiendo del área responsable de cada uno de los procesos:
 - **Casos de Prueba Software:** En esta carpeta se incluirá la información relacionada con los casos de prueba asociados al aplicativo en sí, dejando a un lado los casos de prueba asociados al área de Interoperabilidad (lógicamente se hará referencia a ellos, pero el detalle de los mismos irá en la carpeta asociada al área de Interoperabilidad).
 - **Casos de Prueba Interoperabilidad:** En esta carpeta se incluirá la información relacionada con los casos de prueba asociados al área de Interoperabilidad. Se dividirán en dos grupos, cada uno de ellos con su carpeta correspondiente, por un lado los casos de prueba asociados a las **pruebas funcionales** y por otro los asociados a las **pruebas de integración**.

Existen (además de los normales) dos tipos de casos de prueba de especial relevancia (tanto para Software como para Interoperabilidad):

- **Prueba de autorización:** Son aquellos casos de prueba relacionados con el control del acceso a las opciones disponibles del aplicativo en función del perfil del usuario. Conforman el conjunto de casos de prueba de autorización. En estos casos de prueba debe indicarse que forman parte de dicho conjunto.
- **Smoke test:** Los casos de prueba del tipo smoke test prueban un conjunto mínimo de funcionalidades que deben estar disponibles para continuar con la ejecución del resto de casos de prueba. Serán los primeros casos de prueba en ejecutarse, y si un solo caso de prueba del tipo smoke test no se supera se darán por finalizadas las pruebas marcando su resultado como no favorable (se probarán a continuación tan solo los casos de prueba del tipo smoke test restantes, la ejecución de los demás casos de prueba se omitirá). Al igual que en el caso anterior, en estos casos de prueba debe indicarse que forman parte de dicho conjunto.

Un mismo caso de prueba puede ser al mismo tiempo del tipo prueba de autorización y del tipo smoke test.

En la mayoría de los casos, durante la etapa de análisis es imposible definir los casos de prueba con el nivel de detalle suficiente (botones, validaciones, opciones, etc. que no se terminan de definir hasta la etapa de construcción). Para ir afinando los casos de prueba a medida que se va avanzando con el desarrollo de la aplicación hasta conseguir que no haya ninguna diferencia entre lo descrito en los casos de prueba y la versión definitiva de la aplicación, se podrán modificar durante la etapa de diseño, incluyendo los cambios que se produzcan o incorporando detalles desconocidos durante la etapa de análisis. **Además, se deberá realizar otra entrega una vez finalizada la etapa de construcción.** En esa última entrega será posible detallar la información contenida en los casos de prueba de manera que se corresponda con la versión definitiva de la aplicación, evitando así disconformidades o ambigüedades. En esta entrega asociada a la etapa de construcción será cuando se incluya en el modelado el **resultado de las pruebas que ha ejecutado el proveedor** junto con las evidencias asociadas (consultas realizadas, respuestas obtenidas, etc.).

En los casos de prueba de integración (ya sean predefinidos o no) las evidencias, debido a su formato, no irán en la descripción del caso de prueba, se recogerán en un único documento las evidencias de todos los casos prueba. Dicho documento se incluirá en la entrega ubicándolo en el repositorio del SAS (SharePoint) y se indicará su ruta en la carpeta "Pruebas de Integración". En el apartado Files se seleccionará el tipo "Web Address" y en el campo "File Path" se indicará la URL del documento en el repositorio. Se recomienda incluir una referencia en la descripción de los casos de prueba a dicho documento.

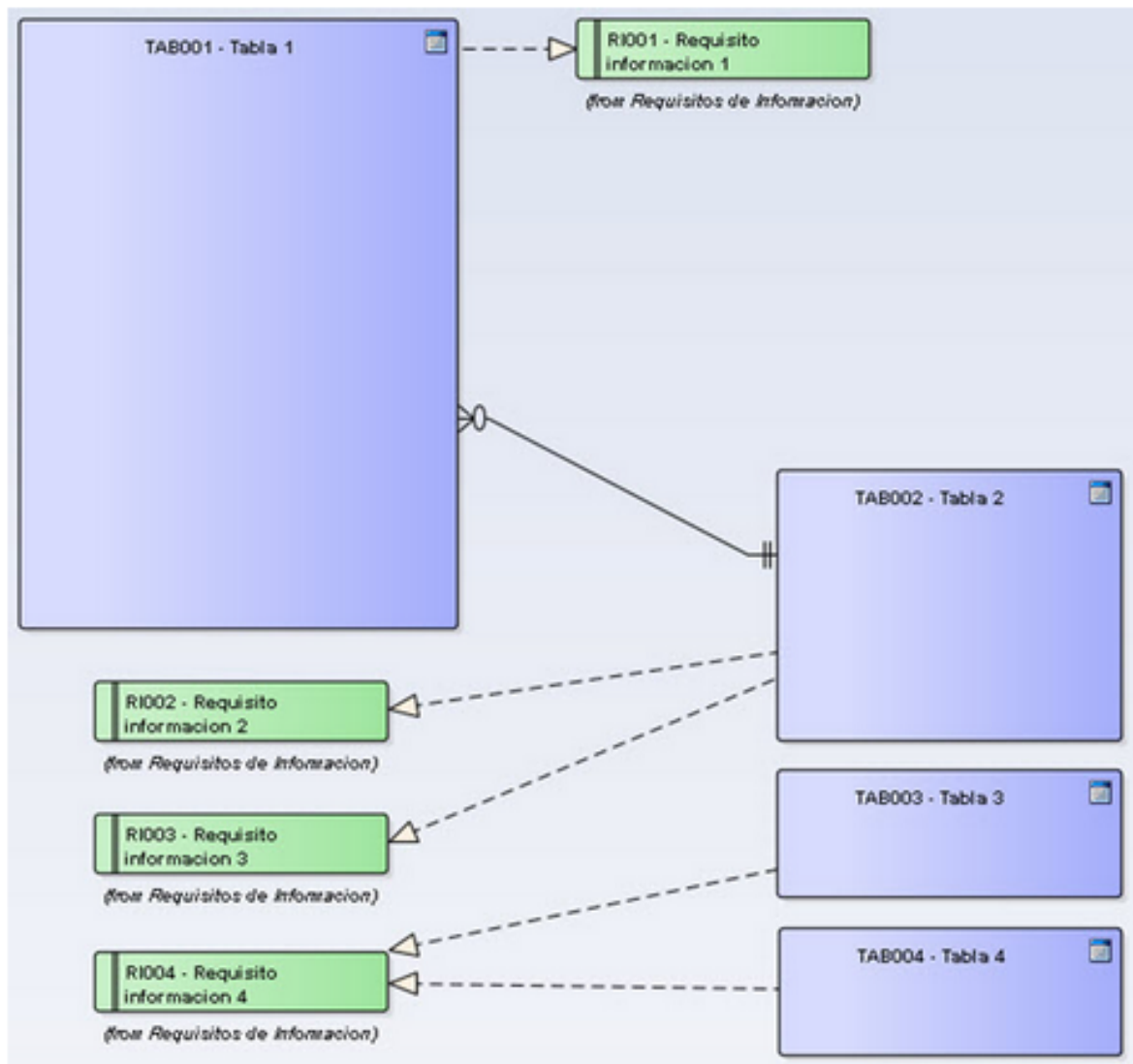
Al igual que ocurre con los casos de prueba, en esta entrega se podrán refinar el resto de elementos, incorporando al modelado los posibles cambios que hayan aparecido durante la construcción.

Diseño del sistema

Esta carpeta contendrá todos los elementos referentes al diseño del sistema.

Aquellos elementos creados durante la etapa de diseño estarán ubicados en las carpetas pertenecientes a este apartado, pero aquellos elementos ya creados en la etapa de análisis, que vuelven a usarse en la etapa de diseño, no serán duplicados, se referenciará al elemento creado durante la etapa de análisis (que seguirá estando ubicado en la carpeta correspondiente del apartado de Análisis), que será modificado en caso de ser necesario.

Para aclarar lo anterior se usará un ejemplo incluido en la plantilla:

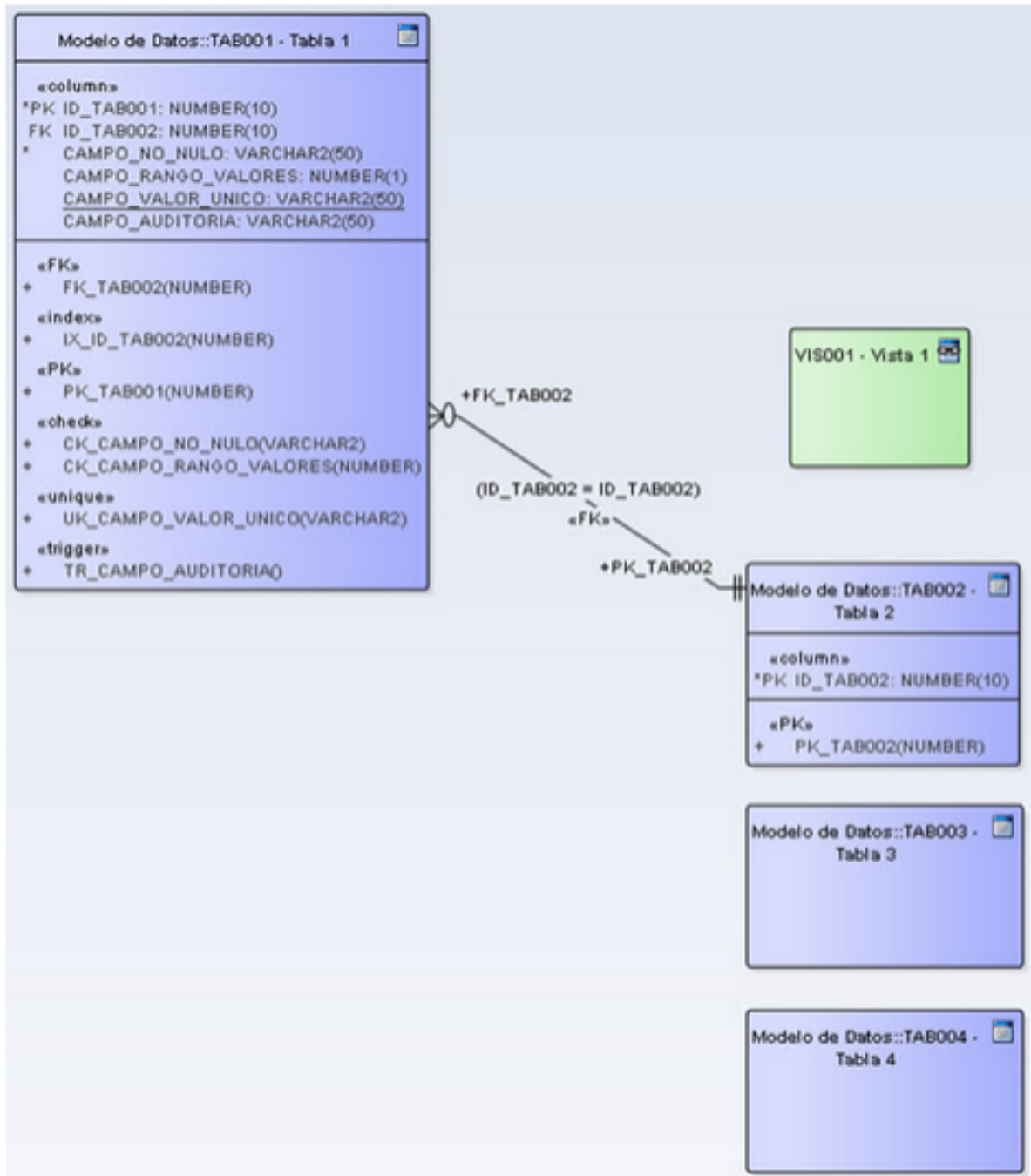


En la anterior imagen aparece el diagrama presente en la carpeta "Modelo de Datos Logico" perteneciente al análisis. Como se puede apreciar, para cada tabla aparece sólo su nombre y las relaciones que mantiene con el resto de tablas, así como los requisitos de información relacionados con cada una de las tablas. Esta es la información que debe incluirse durante la etapa de análisis.

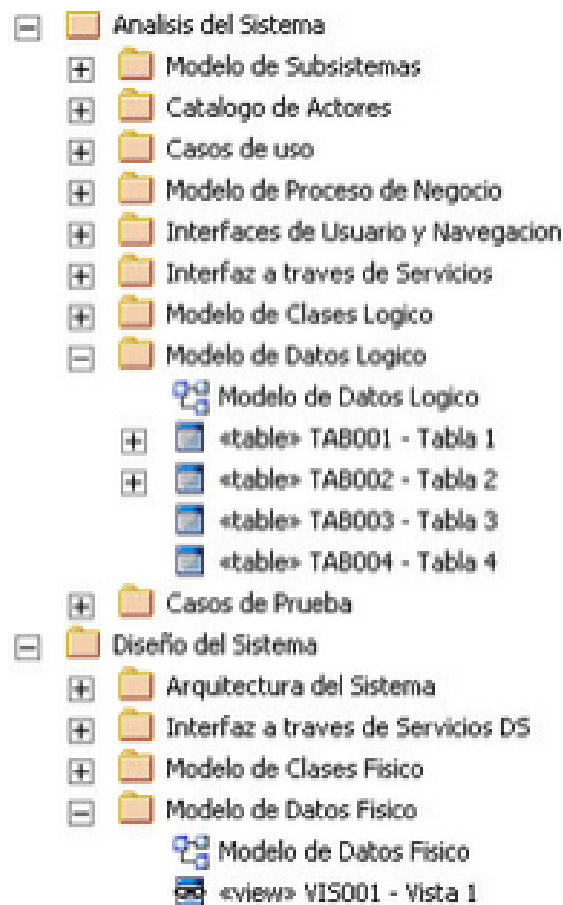
Una vez finalizado el análisis y comenzada la etapa de diseño, para cada tabla es necesario indicar todos sus campos, relaciones y restricciones, así como incluir en caso de ser necesario nuevos elementos propios de esta etapa no presentes durante el análisis (vistas, procedimientos, etc.). En lugar de duplicar los elementos creados durante el análisis creando unos nuevos equivalentes para la etapa de diseño, se ha optado por usar los mismos ampliando la información ya incluida durante el análisis para contemplar la propia del diseño. De esta manera se evitarán duplicidades innecesarias (así como ambigüedades y contradicciones), teniendo recogida en un mismo elemento toda la información referente al mismo. Desde las carpetas correspondientes al diseño, a la hora de completar los diferentes diagramas, simplemente se referenciarán estos elementos presentes en las carpetas equivalentes del apartado de análisis.

En la siguiente imagen se muestra el diagrama presente en la carpeta "Modelo de Datos Fisico", donde se pueden apreciar las mismas tablas que ya aparecían en el diagrama perteneciente a la etapa de análisis, pero con la

información propia del diseño (para mostrar más o menos información en un diagrama basta con acceder a las propiedades del mismo y en el apartado "Elements" seleccionar la que se desea visualizar), así como los nuevos elementos creados durante la etapa de diseño:

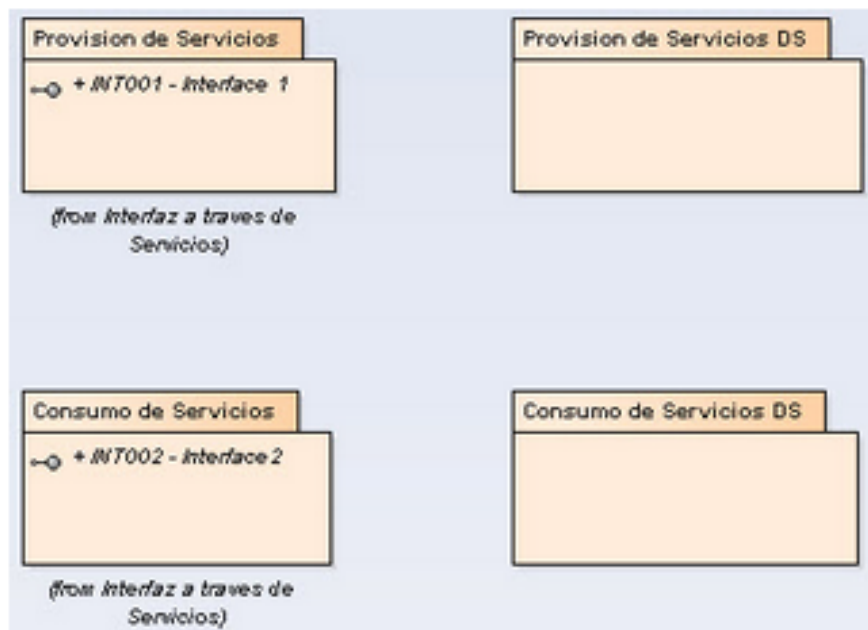


Aparece la información correspondiente a la etapa de diseño de los elementos ya creados durante la etapa de análisis, así como los nuevos elementos creados durante esta última etapa (vista). La diferencia es que los elementos creados durante la etapa de análisis siguen estando en la carpeta correspondiente a esta etapa y los de nueva creación están ubicados en la carpeta correspondiente a la etapa de diseño, como se puede ver en la siguiente figura:



A continuación se detallan las carpetas que albergarán a los nuevos elementos creados durante la etapa de diseño:

- **Arquitectura del sistema:** en esta carpeta se debe incorporar la información acerca de la arquitectura del sistema representándola mediante diagramas del tipo "Deployment Diagram". Se usarán los elementos de Enterprise Architect destinados a representar esta información, los contenidos en "Deployment" y en "Deployment Relationships" (Toolbox). El contenido de esta carpeta no será revisado por la OCA, pero es necesario que se incorpore la información asociada a la arquitectura del sistema para poder disponer de ella, tanto en caso de duda durante la revisión del resto de elementos, como en caso de que algún otro interesado del proyecto necesite consultarla.
- **Interfaz a través de servicios DS:** la estructura de esta carpeta es la misma que la de la carpeta análoga perteneciente al análisis. Se tienen en cuenta tanto los elementos creados en la etapa de análisis como los creados durante el diseño:



- **Provisión de Servicios DS:** en esta carpeta se incluirán los nuevos interfaces creados durante la etapa de diseño (si fuera necesario añadir alguno más no contemplado durante la etapa de análisis), si bien en su diagrama aparecerán todos (los creados durante la etapa de análisis y los creados durante la etapa de diseño). Será en esta etapa cuando se amplie la información ya incluida durante el análisis, incorporando información más detallada en cada uno de los interfaces.
- **Consumo de Servicios DS:** aplica lo comentado para la de "Provisión de Servicios".
- **Modelo de Clases Físico:** la estructura de esta carpeta es la misma que la de la carpeta análoga perteneciente al análisis. Al igual que ocurre con la carpeta "Interfaz a través de servicios DS" se tienen en cuenta tanto los elementos creados en la etapa de análisis como los creados durante el diseño. En esta carpeta se incluirán las nuevas clases creadas durante la etapa de diseño (si fuera necesario añadir alguna más no contemplada durante la etapa de análisis), si bien en su diagrama aparecerán todas (las creadas durante la etapa de análisis y las creadas durante la etapa de diseño). Será durante esta etapa cuando se amplie la información ya incluida durante el análisis, detallándola en cada una de las clases.
 - En la etapa de diseño se incorporará en este apartado el **catálogo de excepciones**.
- **Modelo de Datos Físico:** aplica lo comentado para la de "Modelo de Clases Físico" (salvo lo tocante al catálogo de excepciones).

Construcción del sistema

Esta carpeta contendrá todos los elementos referentes a la construcción del sistema que se va a desarrollar. Se usarán los elementos de Enterprise Architect destinados a representar esta información. El contenido de esta carpeta no será revisado por la OCA, pero es necesario que se incorpore la información asociada a la construcción del sistema para poder disponer de ella, tanto en caso de duda durante la revisión del resto de elementos, como en caso de que algún otro interesado del proyecto necesite consultarla.

Nomenclatura, contenido y relaciones

Se necesita una normativa lo suficientemente detallada como para evitar cualquier tipo de confusión o interpretación equivocada. Esto último es imprescindible para automatizar cualquier proceso de testing o de generación de documentación (e incluso de código o de scripts) a partir del Enterprise Architect.

Se hace hincapié en la necesidad de especificar, además de todos los elementos existentes, las relaciones existentes entre los mismos. Esto es imprescindible para futuras modificaciones o correcciones en el aplicativo, permitiendo detectar a priori efectos colaterales no deseados o incluso repercusiones que hagan replantearse la conveniencia o no de ejecutar cualquier cambio tal y como estaba previsto en principio. Asimismo es imprescindible para poder tener la trazabilidad necesaria en cualquier tipo de revisión.

En el siguiente [enlace](#) puede consultarse para cada uno de los elementos se describe la nomenclatura a usar, el contenido (campos de Enterprise Architect) que debe cumplimentarse y las relaciones con el resto de elementos que deben incluirse (en caso de duda puede recurrirse a la plantilla y consultar los elementos de ejemplo).

Matrices de trazabilidad

En la siguiente tabla aparecen las relaciones que deben modelarse entre los diferentes elementos de la plantilla de Enterprise Architect en el caso de **aplicativos software**:

	OBJ	RF	RINT	RNF	RI	SUB	ACT	SIST	CU	PANT	INF	INT	CL	TAB	CP	CPI	CPFI	EXC
OBJ		X	X	X														
RF	X				X				X									
RINT	X				X				X									
RNF	X				X				X									
RI		X	X	X	X									X				
SUB						X												
ACT							X		X									
SIST									X			X						
CU		X	X	X			X	X	X	X	X	X	X		X		X	
PANT									X	X	X							
INF									X	X								
INT								X	X				X		X	X	X	
CL									X			X	X					X
TAB					X									X				

CP									X			X						
CPI												X						
CPFI									X			X						
EXC													X					X

Para comprobar que las relaciones modeladas son las correctas se ha incluido en la plantilla de Enterprise Architect un conjunto de matrices de trazabilidad. Estas matrices permiten detectar fácilmente la ausencia de relaciones necesarias (porque hay elementos a los que no se les da solución) o aquellas relaciones innecesarias (porque hay elementos innecesarios). Estas matrices están ubicadas en "Matrix Profiles" dentro de "Resources".

En la siguiente tabla aparecen recogidas las matrices existentes en la plantilla junto con las fases en las que aplican y las comprobaciones a realizar en cada una de ellas en el caso de **aplicativos software**:

Matriz	Fase	Comprobaciones
OBJ x Catálogo de Requisitos	DR, AS, DS	Todo objetivo debe tener al menos un requisito asociado (sin tener en cuenta los requisitos de información), y no pueden existir requisitos sin objetivos asociados (sin tener en cuenta los requisitos de información).
Catálogo de Requisitos x RI	DR, AS, DS	No pueden existir requisitos de información sin requisitos asociados. Podrán existir requisitos de información que sólo tengan relación con otros requisitos de información, siempre que estos últimos tengan relación con otros requisitos funcionales, no funcionales y/o de integración.
Catálogo de Requisitos x CU	AS, DS	Todo requisito debe tener al menos un caso de uso asociado (sin tener en cuenta los requisitos de información), y no pueden existir casos de uso sin requisitos asociados (sin tener en cuenta los requisitos de información).
Catálogo de Actores x CU	AS, DS	Todo actor o sistema debe tener al menos un caso de uso asociado, y no pueden existir casos de uso sin actores o sistemas asociados.
CU x PANT	AS, DS	No pueden existir pantallas sin casos de uso asociados.
CU x INF	AS, DS	No pueden existir informes sin casos de uso asociados.
SIS x Consumo de Servicios	AS, DS	No pueden existir interfaces a través de servicios del tipo consumo de servicios sin sistemas asociados.
SIS x Consumo de Servicios DS	DS	No pueden existir interfaces a través de servicios del tipo consumo de servicios sin sistemas asociados.
CU x INT	AS, DS	No pueden existir interfaces a través de servicios sin casos de uso asociados.
CU x INT DS	DS	No pueden existir interfaces a través de servicios sin casos de uso asociados.
CU x CL	AS, DS	No pueden existir clases sin casos de uso asociados. Podrán existir clases que sólo tengan relación con otras clases, siempre que estas últimas tengan relación con casos de uso.

CU x CL DS	DS	No pueden existir clases sin casos de uso asociados. Podrán existir clases que sólo tengan relación con otras clases, siempre que estas últimas tengan relación con casos de uso.
INT x CL	AS, DS	No pueden existir interfaces a través de servicios sin clases asociadas.
INT x CL DS	AS, DS	No pueden existir interfaces a través de servicios sin clases asociadas (sin tener en cuenta las excepciones).
INT DS x CL	DS	No pueden existir interfaces a través de servicios sin clases asociadas.
INT DS x CL DS	DS	No pueden existir interfaces a través de servicios sin clases asociadas..
RI x TAB	AS, DS	No pueden existir tablas sin requisitos de información asociados.
RI x TAB DS	DS	No pueden existir tablas sin requisitos de información asociados.
CU x CP	AS, DS	Todo caso de uso debe tener al menos un caso de prueba asociado (o en su defecto un caso de prueba funcional de interoperabilidad), y no pueden existir casos de prueba sin casos de uso asociados.
CU x CPFI	AS, DS	Todo caso de uso debe tener al menos un caso de prueba funcional de interoperabilidad asociado (o en su defecto un caso de prueba).
INT x CP	AS, DS	No pueden existir interfaces a través de servicios sin casos de prueba asociados (o en su defecto un caso de prueba funcional de interoperabilidad).
INT DS x CP	AS, DS	No pueden existir interfaces a través de servicios sin casos de prueba asociados (o en su defecto un caso de prueba funcional de interoperabilidad).
INT x CPI	AS, DS	No pueden existir interfaces a través de servicios sin casos de prueba de integración asociados, y no pueden existir casos de prueba de integración sin interfaces a través de servicios asociadas.
INT DS x CPI	AS, DS	No pueden existir interfaces a través de servicios sin casos de prueba de integración asociados, y no pueden existir casos de prueba de integración sin interfaces a través de servicios asociadas.
INT x CPFI	AS, DS	No pueden existir interfaces a través de servicios sin casos de prueba funcionales de interoperabilidad asociados (o en su defecto un caso de prueba), y no pueden existir casos de prueba funcionales de interoperabilidad sin interfaces a través de servicios asociadas.
INT DS x CPFI	AS, DS	No pueden existir interfaces a través de servicios sin casos de prueba funcionales de interoperabilidad asociados (o en su defecto un caso de prueba), y no pueden existir casos de prueba funcionales de interoperabilidad sin interfaces a través de servicios asociadas.
CL x EXC	DS	No pueden existir excepciones sin clases asociadas. Podrán existir excepciones que sólo tengan relación con otras excepciones, siempre que estas últimas tengan relación con clases.
CL DS x EXC	DS	No pueden existir excepciones sin clases asociadas. Podrán existir excepciones que sólo tengan relación con otras excepciones, siempre que estas últimas tengan relación con clases.

Versionado y entregas

Para el versionado de los diferentes elementos se utilizará el campo "Version" para incluir la versión y la release. Se utilizarán dos dígitos tanto para la versión como para la release, siendo la primera versión (la correspondiente a la primera entrega) la v01r01. A partir de esta primera entrega, todos los elementos de nueva creación, así como los elementos ya existentes que sufran modificaciones, deberán especificar en el campo "Version" la información correspondiente a la próxima entrega, es decir, la correspondiente a la versión en la que se esté trabajando. Se utilizará un versionado común durante todo el proyecto, es decir, no se empezará un versionado nuevo para cada fase, se continuará con el ya existente.

Se entenderá también como modificación el crear una relación entre dos elementos, teniendo por tanto que versionar ambos elementos asociándoles la versión correspondiente a la próxima entrega.

Cada vez que se vaya a realizar una entrega deberá realizarse el versionado previo de la plantilla estableciendo una línea base (baseline) partiendo de la carpeta raíz, cuya versión será la correspondiente a la entrega que se va a realizar (vXXrXX). Una vez creada la línea base se exportará a un fichero xml que será el que se entregue.

Para llevar a cabo lo anterior se procederá del siguiente modo, sobre "Programa ¿ Nombre programa" pulsando el botón derecho seleccionamos "Package Control", y a continuación seleccionamos la opción "Package Baselines". Para crear la línea base se usará la opción "New Baseline". Una vez creada la línea base se exportará dicha línea base a un fichero xml que será el que se entregue, con la opción "Export File" seleccionando previamente la línea base a exportar.

Se recomienda el uso de la versión 12 de Enterprise Architect a la hora de trabajar con la plantilla, en cualquier caso, independientemente de la versión de Enterprise Architect usada, el fichero xml generado debe generarse para la especificación 1.1 de xmi.

El nombre del fichero xml se construirá siguiendo la siguiente nomenclatura:

Nombre_FF_EA_vVvRR.xml

Nombre = Código ALM del aplicativo en la CMS

FF = Fase (DR = Definición de Requisitos, AS = Análisis del Sistema, DS = Diseño del Sistema, CO = Construcción del Sistema)

EA = Hace referencia al tipo de entregable (Enterprise Architect)

VV = Versión

RR = Release

Aclarar que si bien la versión (versión y reléase en realidad) irá aumentando a lo largo de la vida del proyecto con cada entrega, la fase puede volver a repetirse, ya que una vez finalizado un desarrollo puede surgir la necesidad de un evolutivo que implique de nuevo entregas asociadas a fases ya finalizadas, repitiéndose así estas fases. Puede además que una fase necesite más de una entrega, repitiéndose así también una misma fase para diferentes versiones. A continuación se muestra un ejemplo de lo anterior:

Nombre_DR_EA_v01r01.xml

Nombre_AS_EA_v02r01.xml

Nombre_AS_EA_v02r02.xml - 2ª entrega correspondiente a la fase de análisis

Nombre_DS_EA_v03r01.xml

Nombre_CO_EA_v04r01.xml

Nombre_DR_EA_v05r01.xml - Se inicia un evolutivo que requiere a su vez de todas las fases

Nombre_AS_EA_v06r01.xml

Nombre_DS_EA_v07r01.xml

Nombre_DS_EA_v07r02.xml - 2ª entrega correspondiente a la fase de diseño del evolutivo

Es importante no confundir la versión del entregable (xml), que coincide con la versión de la baseline creada asociada a la entrega, y que además aparece en todos los elementos existentes en el modelado en el campo correspondiente ("Version"), con la versión del aplicativo. Todo lo comentado anteriormente se refiere a la versión del entregable asociado al modelado (y de los elementos que contiene), que debe tratarse como un entregable más del proyecto del tipo documentación, como por ejemplo el manual de usuario, que lleva su propio versionado como documento independientemente del versionado de la aplicación (versionado asociado a la entrega de software). Se debe incorporar en el campo de notas de la baseline la versión del aplicativo (y la fecha en la que se crea la baseline), para mantener así la correspondencia entre ambos versionados de una manera sencilla.

En el caso de que una entrega incluya varias fases, para la nomenclatura se debe usar siempre el valor asociado a la última fase de las incluidas en esa entrega. Por ejemplo, si una entrega incluye análisis y diseño, debe usarse para la nomenclatura el valor asociado al diseño (DS).

Se deberá entregar además un fichero con la información del **glosario** y del **control de cambios**. Para ello se debe exportar la anterior información mediante la opción "Export Reference Data" (dentro del menú "PROJECT" en la opción "Data Management"), seleccionando "Project Glossary" (dentro de "Project") y Team Review. El nombre del fichero se construirá siguiendo la siguiente nomenclatura:

Nombre_FF_EAERD_vVvRR.xml

Nombre = Código ALM del aplicativo en la CMS

FF = Fase (DR = Definición de Requisitos, AS = Análisis del Sistema, DS = Diseño del Sistema, CO = Construcción del Sistema)

EAERD = Hace referencia al tipo de entregable (Export Reference Data del Enterprise Architect)

VV = Versión

RR = Release

Si bien no es obligatorio, puesto que la documentación generada a partir del Enterprise Architect será un entregable más, es conveniente entregar también un fichero con la información de las **constantes para la generación de la documentación**. Para ello se debe exportar la anterior información mediante la opción "Export" (dentro del menú "PACKAGE" la opción "Documentation" y dentro de esta última la opción "Document Generator(TRF/PDF/DOCX)¿" entrando en el apartado "Project Constants"). El nombre del fichero se construirá siguiendo la siguiente nomenclatura:

Nombre_FF_EAPC_vVvRR.xml

Nombre = Código ALM del aplicativo en la CMS

FF = Fase (DR = Definición de Requisitos, AS = Análisis del Sistema, DS = Diseño del Sistema, CO = Construcción del Sistema)

EAPC = Hace referencia al tipo de entregable (Project Constants del Enterprise Architect)

VV = Versión

RR = Release

Adaptación a la nueva versión de la plantilla de contenido ya existente

Por último se incluyen una serie de instrucciones que permitirán adaptar a la versión actual de la plantilla el contenido ya existente creado con versiones anteriores:

- Siguiendo las instrucciones del apartado anterior, desde la anterior versión de la plantilla, generar los ficheros xml con el contenido ya existente y con el glosario y el control de cambios correspondientes, como si de una entrega se tratase (la nomenclatura y el versionado no importan puesto que este proceso es interno para el proveedor y por lo tanto para la OCA es transparente).
- En el caso de que se estuviese trabajado con versiones previas de la plantilla que ya incluyen el uso de las constantes para la generación de la documentación, debe generarse también el fichero xml correspondiente.
- Ya en la nueva plantilla, restaurar el xml con el contenido. Para llevar a cabo lo anterior se procederá del siguiente modo, sobre "Programa ¿ Nombre programa" pulsando el botón derecho seleccionamos "Package Control", y a continuación seleccionamos la opción "Package Baselines". Para importar el xml con el contenido se usará la opción "Import File", creando así una nueva línea base. Una vez creada la línea base se seleccionará y a continuación se usará la opción "Restore to Baseline", incorporando así el contenido ya existente a la nueva plantilla.
- Para incorporar en la nueva plantilla el glosario y el control de cambios usaremos la opción "Import Reference Data" (dentro del menú "PROJECT" en la opción "Data Management"), usando el xml anteriormente generado y seleccionando "Project Glossary" y "Team Review" en "Select Databases to Import".
- Si se estaba trabajando ya con las constantes de proyecto (con las definidas en la normativa presentes ya en la nueva plantilla lógicamente, no con otras) se puede incorporar la información de manera automática en la nueva plantilla para no tener que volverla a incorporar a mano. Para ello se debe importar el xml anteriormente generado mediante la opción "Import" (dentro del menú "PACKAGE" la opción "Documentation" y dentro de esta última la opción "Document Generator(TRF/PDF/DOCX)¿" entrando en el apartado "Project Constants").
- Una vez hecho lo anterior se le debe cambiar el nombre a las siguientes carpetas (los acentos se han omitido a propósito):
 - "Modelo de Clases" pasará a llamarse "Modelo de Clases Logico".
 - "Modelo de Datos" pasará a llamarse "Modelo de Datos Logico".
 - "Modelo de Clases DS" pasará a llamarse "Modelo de Clases Fisico".
 - "Modelo de Datos DS" pasará a llamarse "Modelo de Datos Fisico".
- Se debe incluir el apartado correspondiente a las excepciones. Para ello, sobre la carpeta "Modelo de Clases Fisico", pulsando el botón derecho seleccionamos "Import/Export", y a continuación seleccionamos la opción "Import package from XMI file". Seleccionaremos el fichero "EXC.xml" (facilitado junto con la plantilla y la normativa) y elegiremos la opción "Import".
- Se debe modificar el apartado correspondiente a las pruebas predefinidas de interoperabilidad, que ha sufrido cambios con respecto a versiones anteriores. Para ello se procederá de una u otra forma en función de si se ha modificado o no su contenido mientras se trabajaba con la anterior plantilla:

- Si no se ha modificado el contenido de este apartado presente en la anterior plantilla, ni siquiera para incorporar relaciones a los casos de prueba predefinidos, basta con sustituir el contenido de esta carpeta por el correspondiente a la nueva versión. Para ello, en primer lugar, se eliminará todo el contenido de la carpeta "Pruebas de Integración Predefinidas" (la carpeta en sí no). A continuación, sobre la carpeta "Pruebas de Integración Predefinidas", pulsando el botón derecho seleccionamos "Import/Export", y a continuación seleccionamos la opción "Import package from XMI file". Seleccionaremos el fichero "PIP.xml" (facilitado junto con la plantilla y la normativa) y elegiremos la opción "Import".
 - Si se ha modificado se debe proceder del siguiente modo:
 - Los casos de prueba asociados a BDU y a MACO ya existentes permanecen igual.
 - Se importan los casos de prueba para los servicios REST. Para ello, sobre la carpeta "Pruebas de Integración Predefinidas", pulsando el botón derecho seleccionamos "Import/Export", y a continuación seleccionamos la opción "Import package from XMI file". Seleccionaremos el fichero "REST.xml" (facilitado junto con la plantilla y la normativa) y elegiremos la opción "Import".
 - En cuanto a los casos de pruebas para el ESB, se eliminarán los casos de prueba predefinidos existentes.
 - Si existiesen las carpetas "Provision de Servicios" y "Consumo de Servicios" colgando de la carpeta "ESB" y tuviesen cualquier tipo de contenido, se creará una carpeta colgando de "ESB" llamada "OLD" y se moverá dicho contenido a la carpeta "OLD". A continuación se eliminarán ambas carpetas (tuviesen o no contenido).
 - Una vez hecho lo anterior, sobre la carpeta "ESB", pulsando el botón derecho seleccionamos "Import/Export", y a continuación seleccionamos la opción "Import package from XMI file". Seleccionaremos el fichero "ESB_PS.xml" (facilitado junto con la plantilla y la normativa) y elegiremos la opción "Import". Este paso se repetirá con el fichero "ESB_CS.xml".
-

Aplicativos sin histórico de documentación

Con respecto a las aplicaciones histórica que no disponen de una documentación mínima en un archivo de EA, se llega a un acuerdo de mínimos para su entrega.

[Más información...](#)

Plantillas y archivos para migración

Es este apartado se disponen las plantillas de EA en las distintas versiones de la normativa que se han distribuido y los archivos de ayuda para la migración de versiones de la plantilla EA.

[Más información...](#)