Guía de desarrollo, Anexo 03.03

Guía de gestión de proyectos en ALM 12.55

Autor: Oficina de Pruebas

GERENCIA INFORMÁTICA
JOSEFA VALCÁRCEL, 44
28027-MADRID

Índice General

1 INTRODUCCIÓN 4

2 MÓDULOS DE LA APLICACIÓN 4

2.1 MODULO MANAGEMENT 4
2.1.1 Releases 4
2.1.1.1 Objetivo 4
2.1.1.2 Workflow de una Release 6
2.1.1.3 Estructura 6
2.1.1.4 Evolutivos 7
2.1.1.5 Subsistemas 8
2.1.1.6 Crear ciclos de otro ya existente 9
2.2 MÓDULO REQUIREMENTS 12
2.2.1 Objetivo 12
2.2.2 Estructura General 13
2.2.3 Información común 15
2.2.4 Requisitos 17
2.2.4.1 Requisitos funcionales 17
2.2.4.2 Requisitos no funcionales 19
2.2.4.2.1 Rendimiento 20
2.2.4.2.2 RNF Usuario de control 21
2.2.4.2.3 RNF Accesibilidad 21
2.2.4.2.4 RNF Auditoría 21
2.2.4.2.5 RNF Control de acceso 22
2.2.4.2.6 RNF Usabilidad 22
2.2.4.2.7 RNF Fiabilidad 22
2.2.4.2.8 RNF Disponibilidad 22
2.2.4.2.9 RNF Multi-idioma 22
2.2.4.2.10 RNF Seguridad 22
2.2.4.2.11 RNF Configuración de las aplicaciones 23
2.2.4.2.12 RNF Cohesión 23
2.2.4.2.13 RNF Almacén de datos 23
2.2.4.2.14 RNF Geolocalización 23
2.2.4.2.15 RNF Eficiencia 23
2.2.4.2.16 RNF Almacenamiento 23
2.2.4.2.17 RNF Portabilidad 24
2.2.4.2.18 RNF Compatibilidad 24
2.2.4.2.19 RNF Interoperabilidad 24
2.2.4.2.20 RNF Información sensible 24
2.2.4.3 Requisitos de interfaces 24
2.2.4.3.1 Interfaces de usuario 24
2.2.4.3.1.1 RNFI-YY: 24
2.2.4.3.2 Interfaces de/con otros sistemas: Ofrecidas 24
2.2.4.3.2.1 RNFI-XX: 25
2.2.4.3.3 Interfaces de/con otros sistemas: Consumidas 25
2.2.4.3.3.1 RNFI-XY: 25
2.2.4.3.4 Interfaces de comunicaciones 25
2.2.4.3.4.1 RNFI-ZZ: 25
2.2.5 Trazabilidad de Requisitos 25
2.3 MÓDULO TEST PLAN 29
2.3.1 Objetivo 29
2.3.2 Estructura 30
2.3.3 Ubicación ficheros de datos de entrada para auditorías 37
2.3.3.1 Fichero XLS de pruebas funcionales 37
2.3.3.2 Ficheros de pruebas de rendimiento 38
2.3.3.3 Otros ficheros 40
2.3.4 Casos de prueba 43

2.3.4.1 Normativa de ubicación de casos de prueba 43
2.3.4.2 Llamada a casos de prueba (Call to test) 43
2.3.4.3 Tipos de casos de prueba 45
2.3.4.3.1 Casos de prueba de rendimiento 45
2.3.4.3.1.1 Reglas de Nomenclatura 46
2.3.4.3.1.2 Ejemplo 46
2.3.4.3.1.3 Pasos a realizar 47
2.3.4.3.2 Casos de prueba funcionales (Manuales) 47
2.3.4.3.2.1 Reglas de Nomenclatura 47
2.3.4.3.2.2 Ejemplo 47
2.3.4.3.2.3 Pasos a realizar 48
2.3.4.3.3 Casos de prueba funcionales (Automáticas) 50
2.3.4.3.3.1 Reglas de Nomenclatura 50
2.3.4.3.3.2 Ejemplo 50
2.3.4.3.3.3 Pasos a realizar 51
2.3.4.3.4 Componentes comunes de casos de prueba funcionales manuales 52
2.3.4.3.4.1 Reglas de Nomenclatura 52
2.3.4.3.4.2 Ejemplo 53
2.3.4.3.4.3 Pasos a realizar 54
2.3.4.4 Documentación de casos de prueba 55
2.3.4.4.1 Detalles de un caso de prueba de rendimiento 55
2.3.4.4.2 Detalles de un caso de prueba funcional 56
2.3.4.4.3 Diseño de pasos 57
2.3.4.4.3.1 Anexar capturas a los pasos 59
2.3.4.4.4 Req. Coverage 60
2.3.4.5 Abrir Scripts Vugen desde ALM 61
2.4 MÓDULO TEST LAB 64
2.4.1 Objetivo 64
2.4.2 Creación de una batería de pruebas 65
2.4.3 Ubicación ficheros de datos de entrada para ejecuciones 67
2.4.3.1 Pruebas funcionales Manuales 67
2.4.3.2 Pruebas funcionales Automatizadas 67
2.4.3.3 Pruebas de carga 68
2.4.3.4 Pruebas de sobrecarga 68
2.4.3.5 Pruebas escalabilidad 69
2.4.3.1 Pruebas seguridad 69
2.4.3.2 Fichero .ZIP 69
2.4.3.3 Varias ejecuciones en una misma build 70
2.5 MÓDULO DEFECTS 70
2.5.1 Objetivo 70
2.5.2 Visualización de defectos en una release-ciclo 70
2.5.3 Modificación de defectos 74
2.5.4 Añadir comentarios 76

3 GESTIÓN DE EVOLUTIVOS 78

3.1 DESCRIPCIÓN 78
3.2 EJEMPLO 79
3.2.1 Módulo Management 80
3.2.2 Módulo Requirements 81
3.2.3 Módulo Test Plan 84
3.3 COMPROBACIÓN 85

1 Introducción

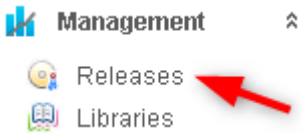
El presente Documento es la Guía de Gestión de Proyectos de la Herramienta **Application Lifecycle Management (ALM)** en la versión 12.55.

Esta Guía **no es un Manual** de **Application Lifecycle Management (ALM)**. Los Manuales Oficiales los podemos encontrar en el menú ayuda de los diferentes productos de HP/MicroFocus.

2 Módulos de la Aplicación

2.1 Modulo Management

2.1.1 Releases



2.1.1.1 Objetivo

Desde el módulo **Management** se realiza la gestión de las Releases del proyecto.

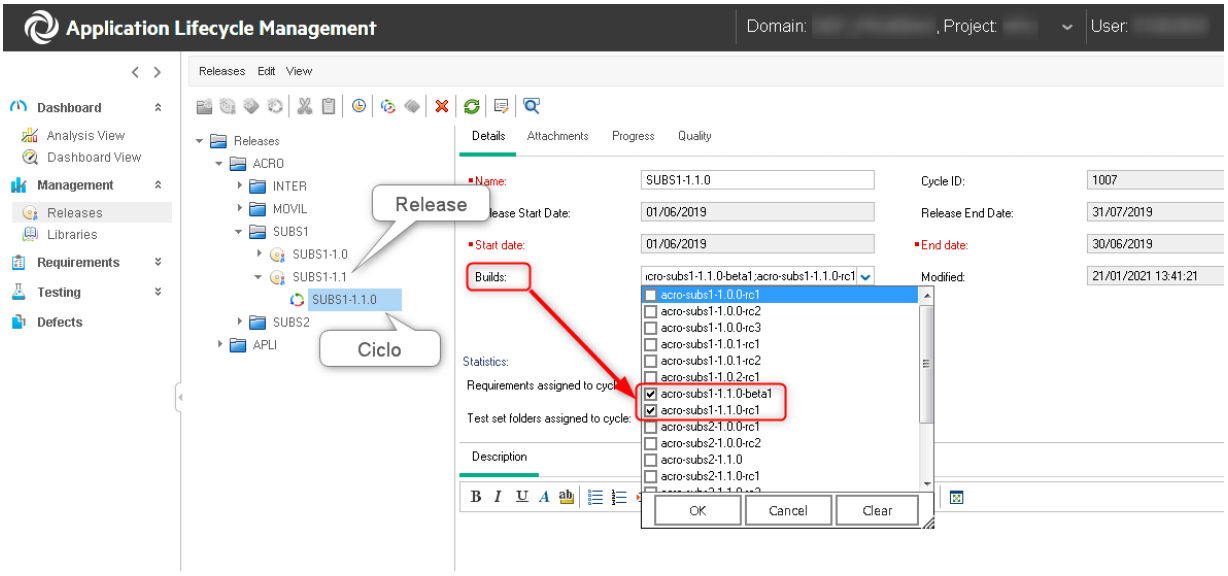
Una Release representa un conjunto de una o varias versiones de la aplicación informática que compartan funcionalidades. Siguiendo el Anexo 38 Normativa de Versionado, una Release estará compuesta por todas las versiones que compartan los dígitos x.y.

Dentro de las Releases están los Ciclos que se corresponden con cada una de las versiones definidas por x.y.z que se pretende distribuir. Por último tenemos las Builds, que son los productos de desarrollo (ejecutable) que serán objeto de las pruebas, que siguen la nomenclatura especificada en el “Anexo 10.0 Guia Tecnica de uso, estructura y nombrado en SVN”. Atendiendo al nombre de la Build

podremos saber si se trata de una versión de Preproducción (con el sufijo –rc, –beta, o sin sufijo), o de Desarrollo (sufijo –SNAPSHOT).

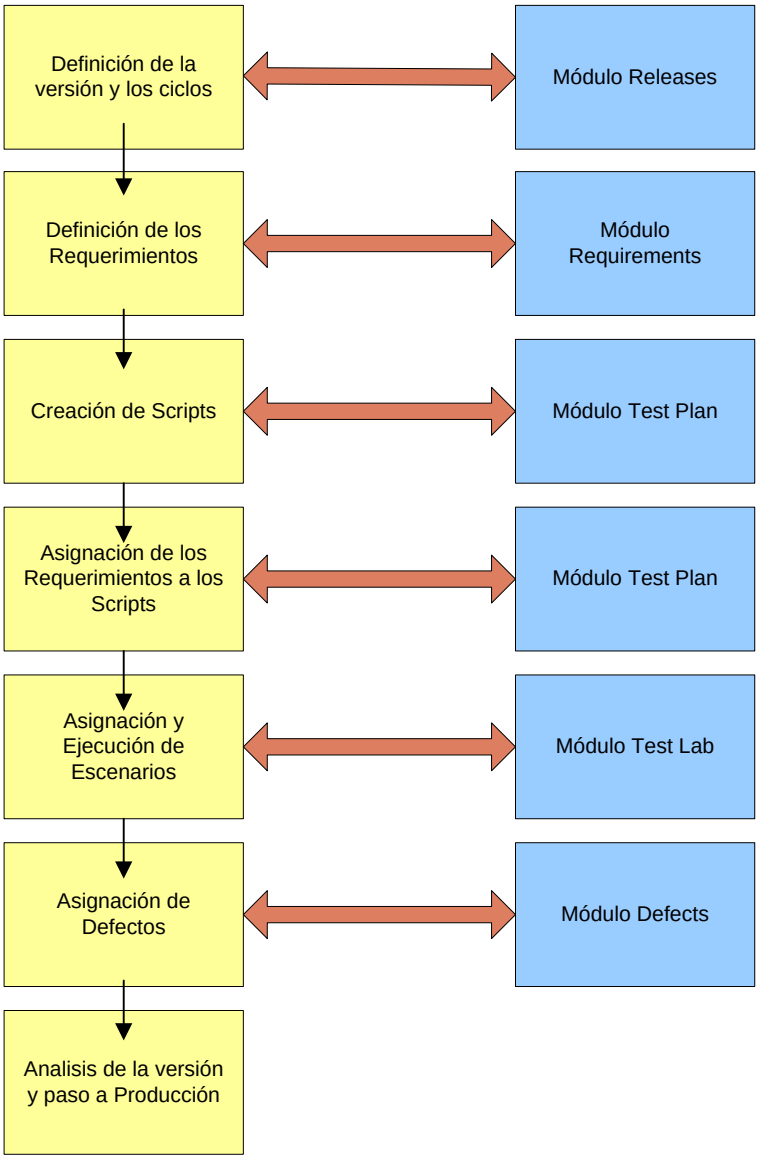
Durante la creación de la aplicación el equipo de desarrollo irá elaborando de manera paralela la documentación asociada (Requerimientos, Planes de Pruebas, Scripts de Rendimiento) que será revisada por la Oficina de Pruebas previamente a la ejecución de las pruebas solicitadas. Por último los Ciclos no contendrán cambios en la funcionalidad de la aplicación, sino correcciones de errores técnicos o de funcionalidades detectados en la versión, y tendrán siempre una fecha de inicio y de fin.
Para poder poner la fecha de inicio y fin, será necesario pulsar botón derecho sobre el ciclo y en el menú contextual, seleccionar la opción "Reschedule". Tanto la Release como el Ciclo se deben crear en ALM a través del módulo "Management – Releases". Las Builds las creará la Oficina de Pruebas a petición de desarrollo cuando se vayan a ejecutar pruebas.

Así por ejemplo, si Desarrollo está trabajando en código en la versión objetivo **acro-subs1-1.1.0**, para lo cual libera varias builds (acro-subs1-1.1.0-rc1 y acro-subs1-1.1.0-beta1), en ALM debería tener las siguientes entidades:
Release: SUBS1-1.1
Cycle: SUBS1-1.1.0
Builds: acro-subs1-1.1.0-rc1, acro-subs1-1.1.0-beta1



2.1.1.2 Workflow de una Release

Pasos de trabajo a realizar en la liberación de una Release y su correspondencia con los módulos de ALM.



2.1.1.3 Estructura

La estructura general a utilizar en el módulo de Management será la siguiente:

+ Requirements
    + Nombre Aplicación (se corresponde con un contrato)
        + Acrónimo aplicación
            + Acrónimo del subsistema (en caso de que tenga subsistemas)
                + Release
                    + Ciclo: Builds asociadas al ciclo

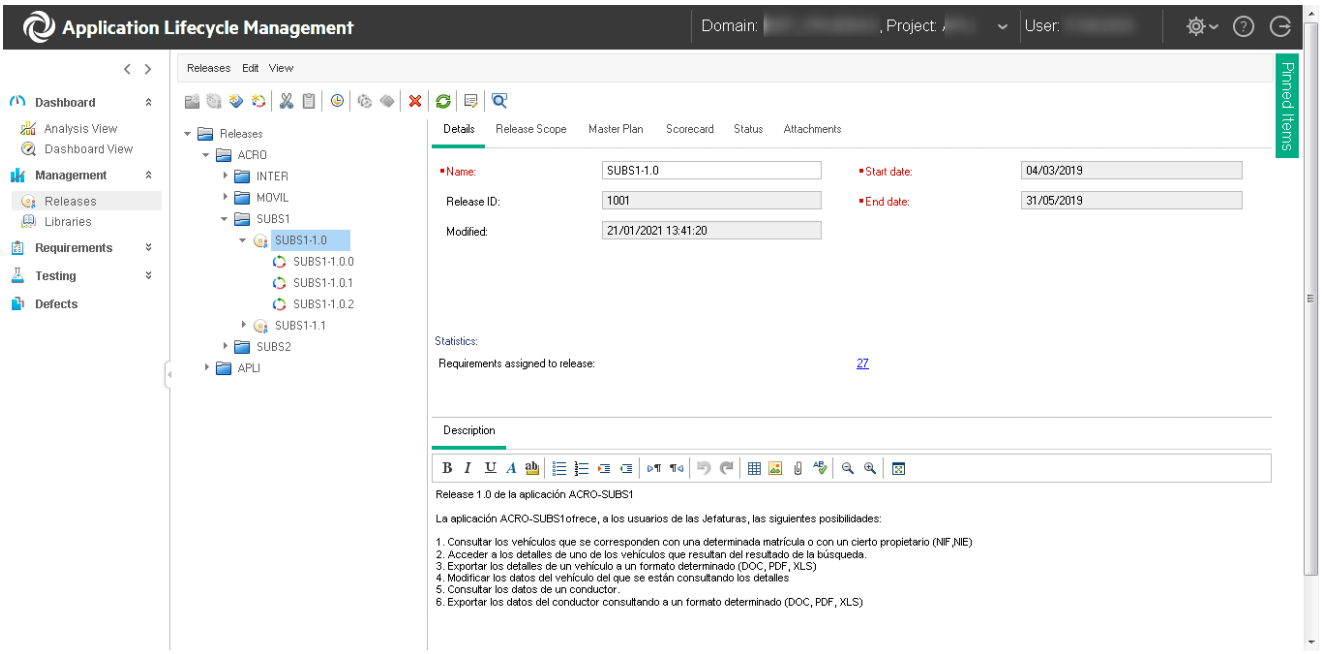


Ilustración 1. Estructura de Releases de proyecto en el módulo Management

2.1.1.4 Evolutivos

Ver sección “3 Gestión de evolutivos”.

2.1.1.5 Subsistemas

En el caso de que se trate de un proyecto con subsistemas se debe crear una carpeta por cada subsistema existente (esto aplica para todos los módulos de ALM). Las Releases, Ciclos y Builds deben incluir el nombre del subsistema en el nombre.

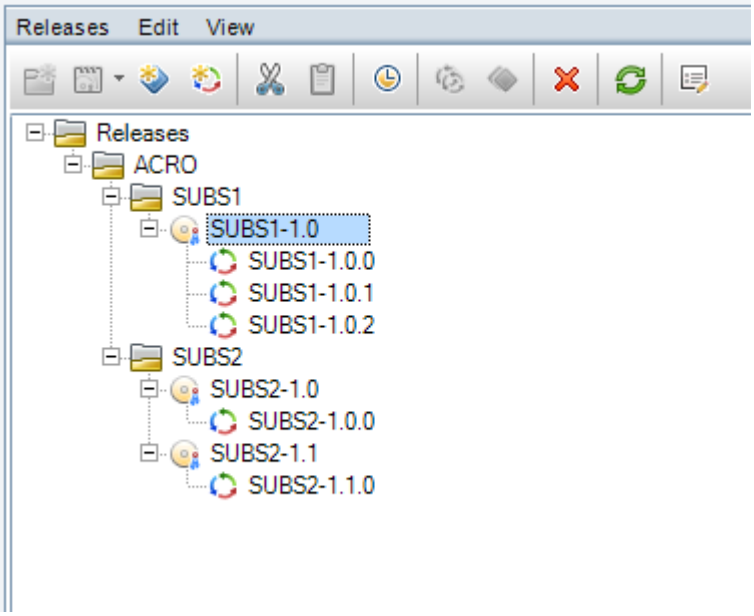


Ilustración 2. Estructura de subsistemas en el módulo Management, con releases menores

En lo que respecta a los requisitos, si existieran requisitos comunes a varios subsistemas deberán duplicarse en ALM y crear un requisito por cada subsistema.

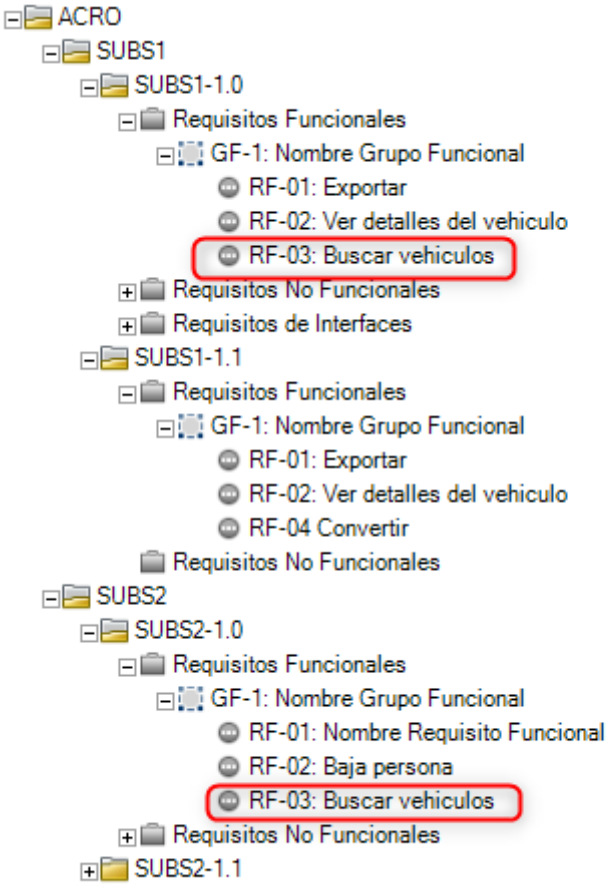


Ilustración 3. Requisitos comunes a subsistemas distintos. Módulo Requirements

2.1.1.6 Crear ciclos de otro ya existente

Desde el módulo **Management** se realiza la gestión de las releases y ciclos de los proyectos. Cuando ya tenemos un ciclo creado ([SUBS-]1.0.0) y queremos crear un nuevo ciclo en el que aparezcan asociados todos los requisitos la mejor opción es:

1. Seleccionar el ciclo último que haya sido validado
2. Pulsar botón derecho y seleccionar la opción "*Duplicate Cycle CTRL+D*"

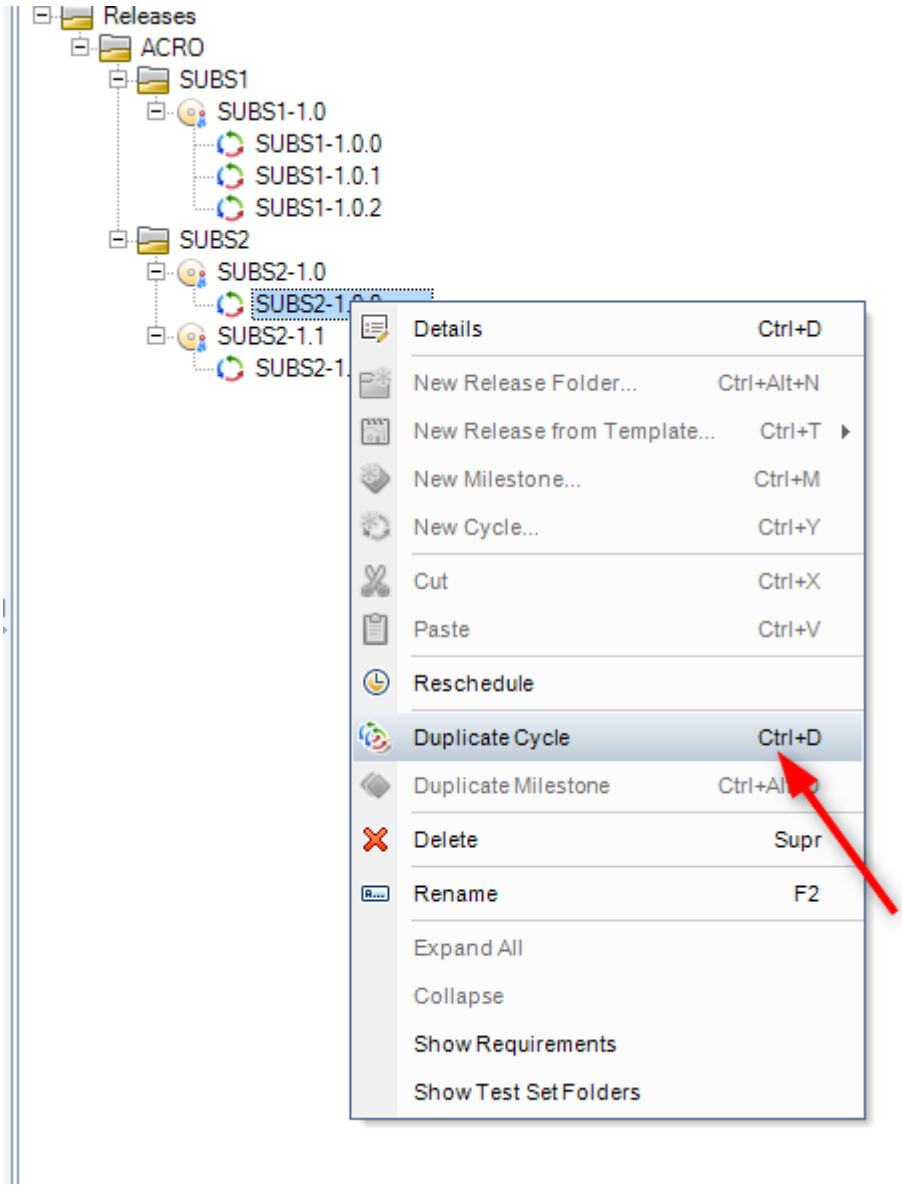


Ilustración 4. Duplicar Ciclo

Se creara un nuevo ciclo con el mismo nombre y finalizado en "_Copy_1". Ahora se deberá proceder a renombrarlo con el nombre del nuevo ciclo.

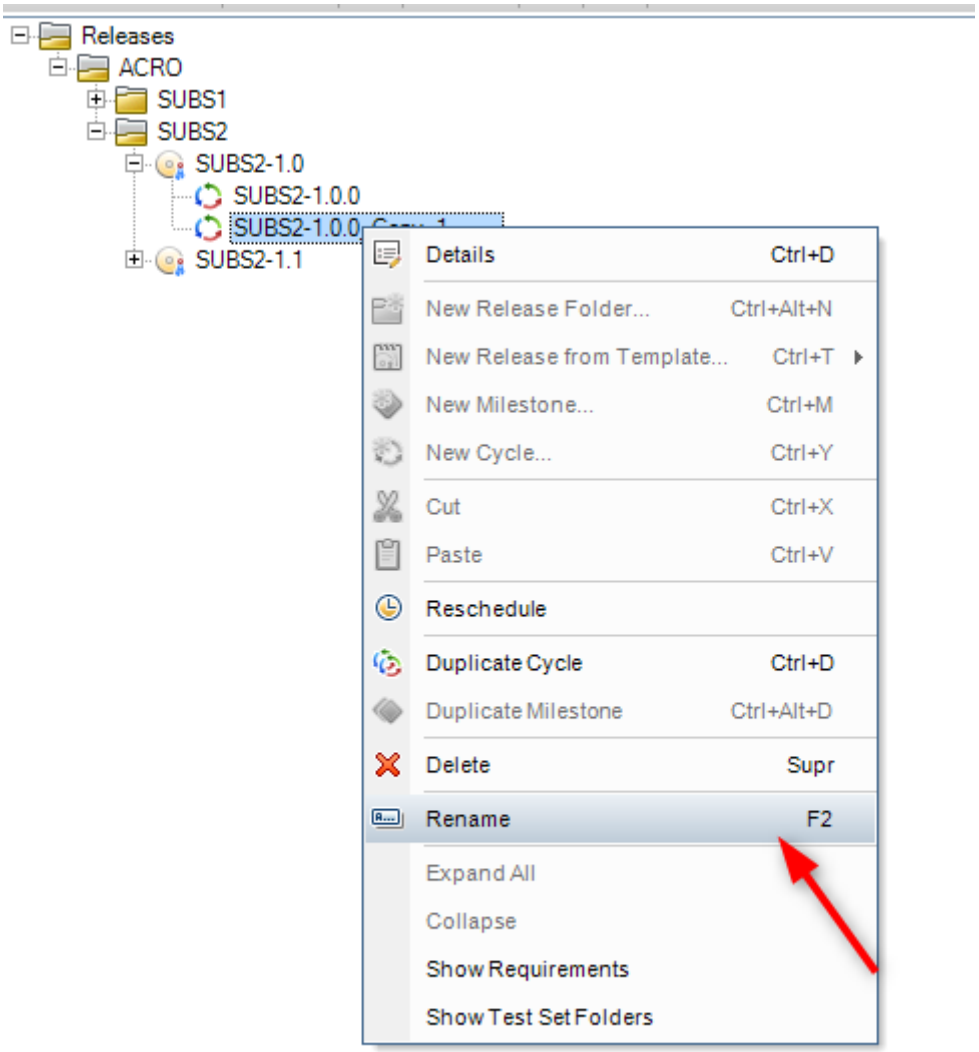
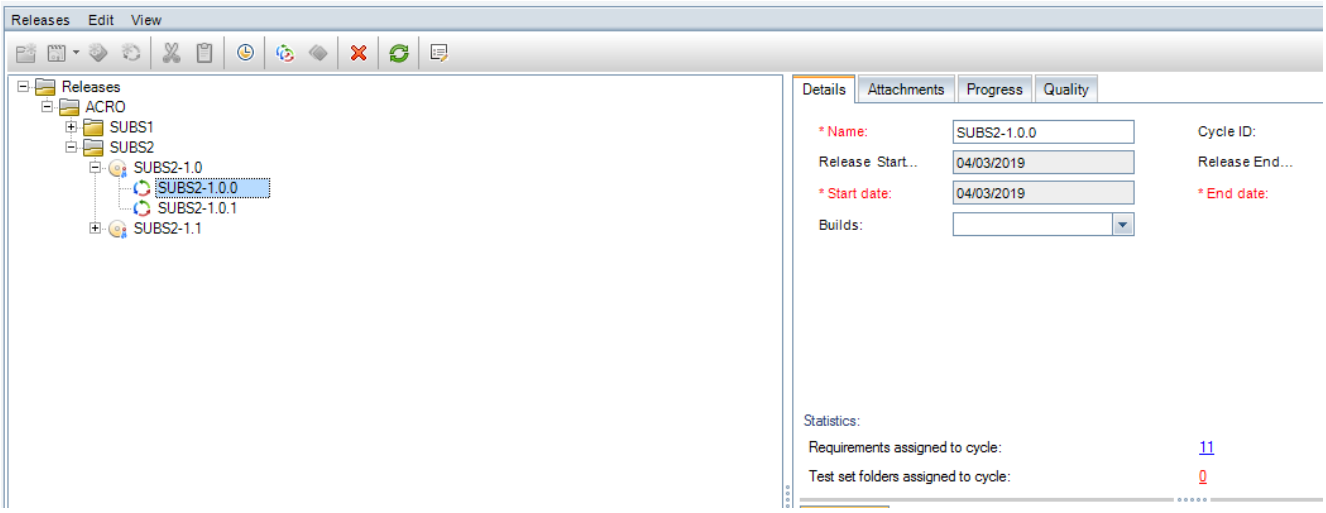


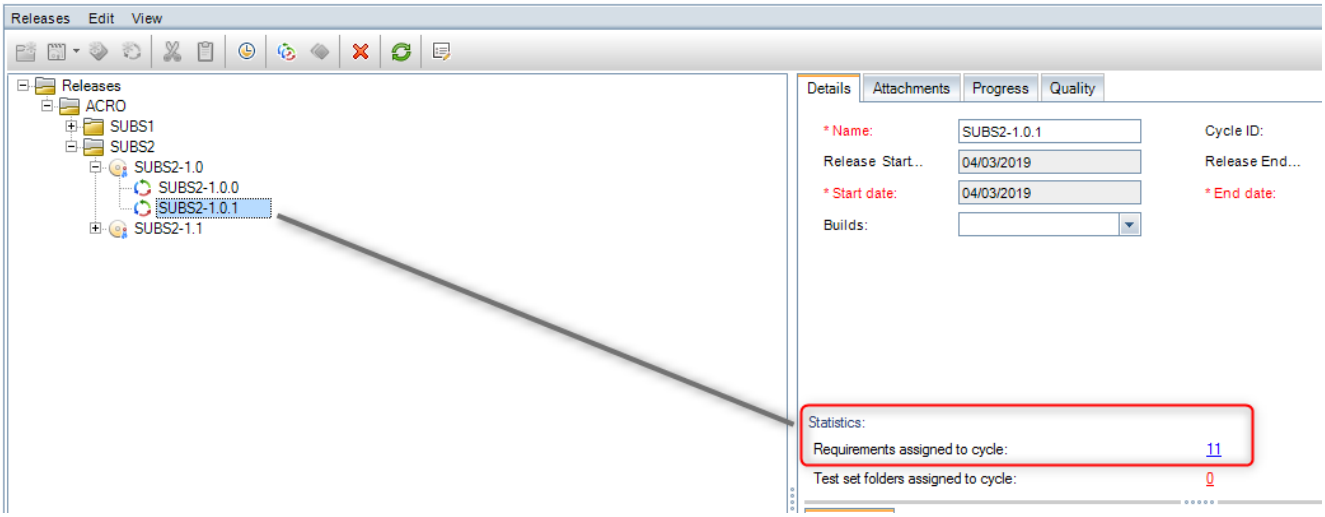
Ilustración 5. Renombrar Ciclo

4. Seleccionar el nuevo ciclo creado y finalizado en "_Copy_1".

5. Pulsar botón derecho y seleccionar la opción "*Rename F2*"

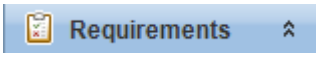
Verificar que siguen asociados el mismo número de requisitos.





2.2 Módulo Requirements

2.2.1 Objetivo



Permite la gestión de Requisitos (Funcionales, de Rendimiento y de Interfaz) y su relación con las Pruebas, es decir, **el grado de cobertura de los Requisitos frente a las Pruebas**.

Los requisitos en árbol es una representación gráfica de su especificación, mostrando la relación jerárquica entre las diferentes necesidades. De cada requisito se puede determinar su cobertura y su riesgo.

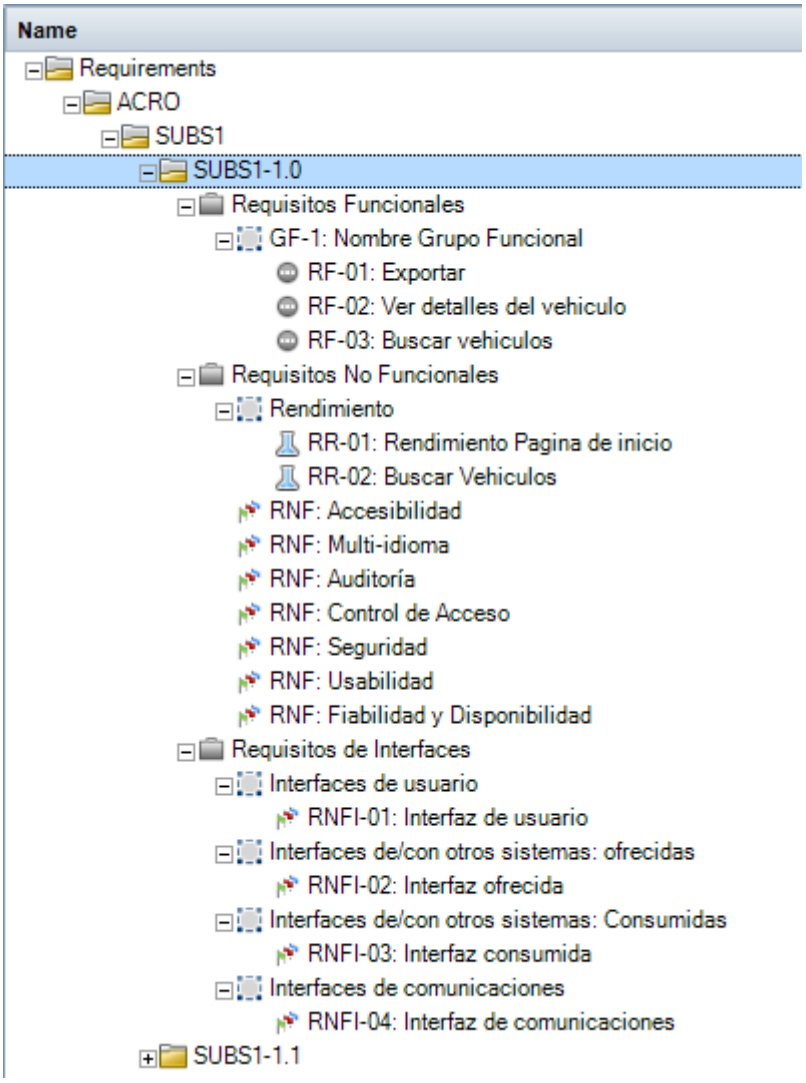
ALM va a ser el repositorio de todos los requisitos de la aplicación y desde aquí se podrá exportar en un documento Word o PDF con todos los requisitos aplicables a la release, como se explica en el Anexo 03.02. En los siguientes puntos se explica cómo se ha de estructurar la información de requisitos en ALM.

2.2.2 Estructura General

La estructura general a utilizar en el módulo de Requerimientos será la siguiente:

```
+ Requirements
    + Nombre Aplicación
        + [Subsistema] (en caso de que tenga subsistemas)
            + Release
                + Requisitos Funcionales
                    + Grupo Funcional (1 a N)
                        + Requisito Funcional (1 a N)
                + Requisitos NO Funcionales
                    + Rendimiento
                        + Requisito Rendimiento (1 a 1)
                    + Otros requisitos no funcionales (1 a N)
                + Requisitos de Interfaces
                    + Interfaces de usuario
                    + Interfaces de/con otros sistemas
                        + Interfaces ofrecidas
                        + Interfaces consumidas
                    + Interfaces de comunicaciones
```

Debe quedar una estructura como la siguiente.



En ALM tenemos el botón “New Requirement” con el que podremos ir creando la estructura anterior. Los tipos de requisitos permitidos son:

- *Functional*
- *Performance*
- *No Functional*
- *Folder (Requisito de Carpeta)*

- *Business (similar a Carpeta para agrupar los requisitos Funcionales, no Funcionales y de Interfaces)*
- *Group (Requisito de Grupo)*

2.2.3 Información común

Situándonos en la estructura anterior al nivel de la "Release" (ej: SUBS1-1.0) (requisito tipo "Folder") introduciremos toda la información común aplicable a la aplicación (o subsistema) en la reléase en curso dentro del campo "Description" siguiendo la siguiente estructura (sustituir el texto en **azul** por la información adecuada de nuestra aplicación):

1. Objeto

El propósito de este documento es presentar una descripción detallada de los requisitos que debe satisfacer el producto x que realiza la funcionalidad explicar qué funcionalidad va a englobar con la finalidad de saber qué se requiere del sistema y establecer el alcance.

2. Audiencia
La audiencia a la que va dirigido es especificar la audiencia a la que va dirigido.

3. Alcance
**Identificación del producto(s) incluyendo su nombre
Consistencia con definiciones similares de documentos de mayor nivel que puedan existir (ej. Pliego de Descripciones Técnicas, identificación de necesidades etc.).

4. Términos y acrónimos
** Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento

4.1 Términos
En este apartado se incluyen, organizados por orden alfabético, los términos que se utilizan en el propio documento para facilitar su lectura y comprensión.
• Excepción: Cualquier error capturado y manejado por la aplicación durante su ejecución

4.2 Acrónimos
En este apartado se incluyen, organizados por orden alfabético, los acrónimos que se utilizan en el propio documento para facilitar su lectura y comprensión.
• DGT: Dirección General de Tráfico

5. Referencias

**Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona. Como mínimo debe incluirse en esta lista el pliego de prescripciones técnicas

6. Descripción general

**Indicar si es un producto independiente o parte de un sistema mayor. En este punto se puede incluir un diagrama que identifique las relaciones con el entorno para facilitar la comprensión.

6.1 Perspectiva del producto

** Visión general tanto interna como externa del producto, modos de operación si los hubiera y cualquier condición del entorno o contexto de la aplicación que deba ser tenida en cuenta más adelante en esta especificación etc.

6.2 Necesidades del producto a satisfacer

**Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.
En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema, Memoria Técnica o del Pliego de Prescripciones Técnicas).
Las funcionalidades deben estar organizadas de manera que el usuario o cualquier interlocutor pueda entenderlo. Para ello se pueden utilizar métodos textuales o gráficos.
Se podrán organizar las necesidades en apartados y se identificarán obligatoriamente con un identificador NEC-XX que coincidirá si se han definido en las matrices de trazabilidad.

6.3 Características de los usuarios

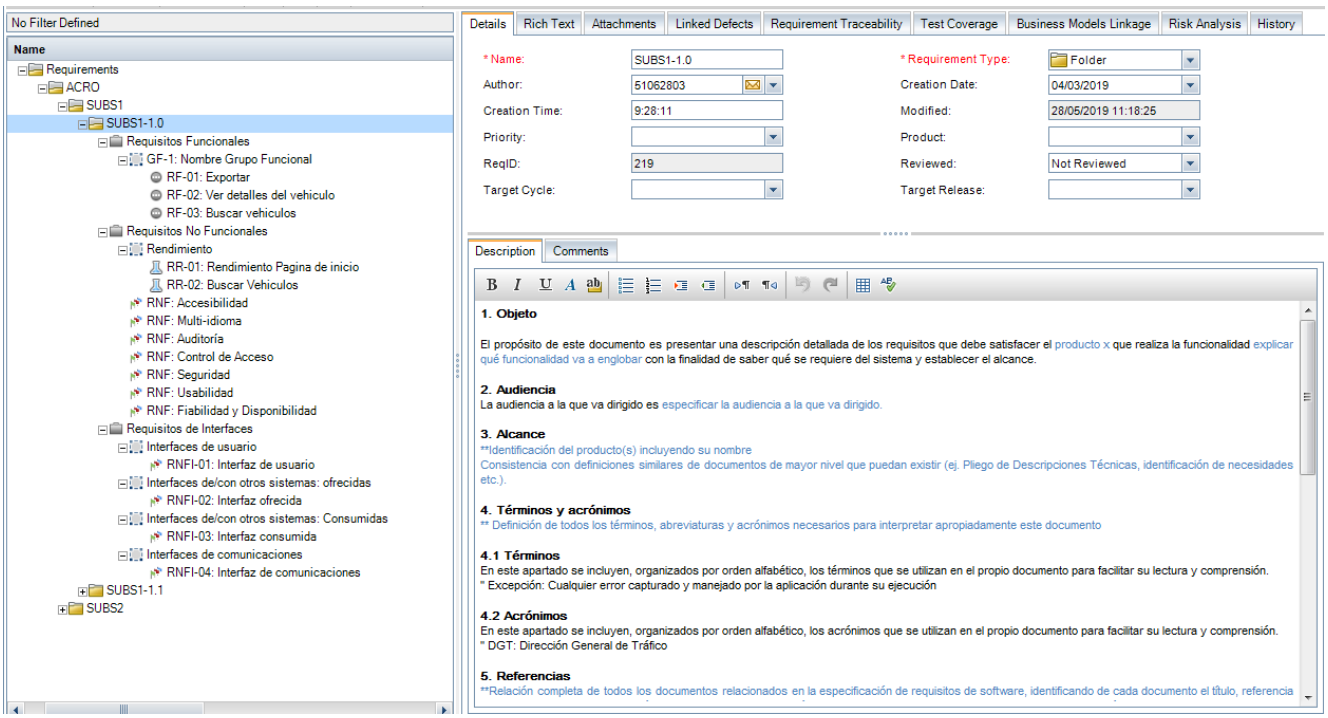
**Descripción de los usuarios, identificando los roles que interactúan con el sistema.
Se podrán organizar los actores en apartados y se identificarán obligatoriamente con un identificador ACT-XX que coincidirá si se han definido en las matrices de trazabilidad.

6.4 Sistemas externos

** Indicar si hay que integrar el producto con otros productos de software, sistemas, subsistemas o aplicaciones.
Se podrán organizar los actores de tipo sistema en apartados y se identificarán obligatoriamente con un identificador ACT-XX que coincidirá si se han definido en las matrices de trazabilidad.

7 Diagrama de contexto

Incluir en el diagrama los usuarios y sistemas externos con los que interactúa el producto **(Añadir adjunto)
Añadir a continuación una breve descripción que ilustre lo que se muestra en el diagrama.



2.2.4 Requisitos

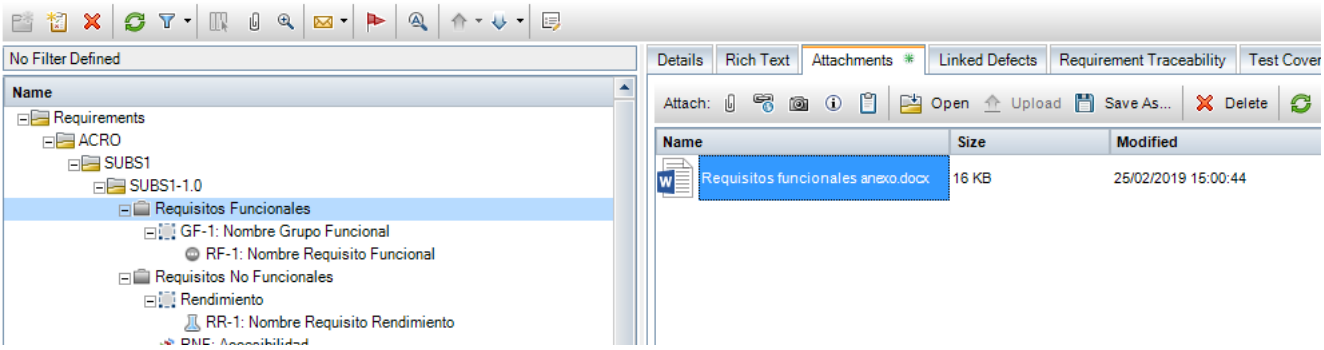
A continuación se introducirán colgando del nivel "Release" los requisitos funcionales, los no funcionales y los de interfaces.

2.2.4.1 Requisitos funcionales

A este nivel se introducirá un requisito tipo "***Business***" con el Name = Requisitos Funcionales en el cual se introducirá la siguiente información en el campo "Description":

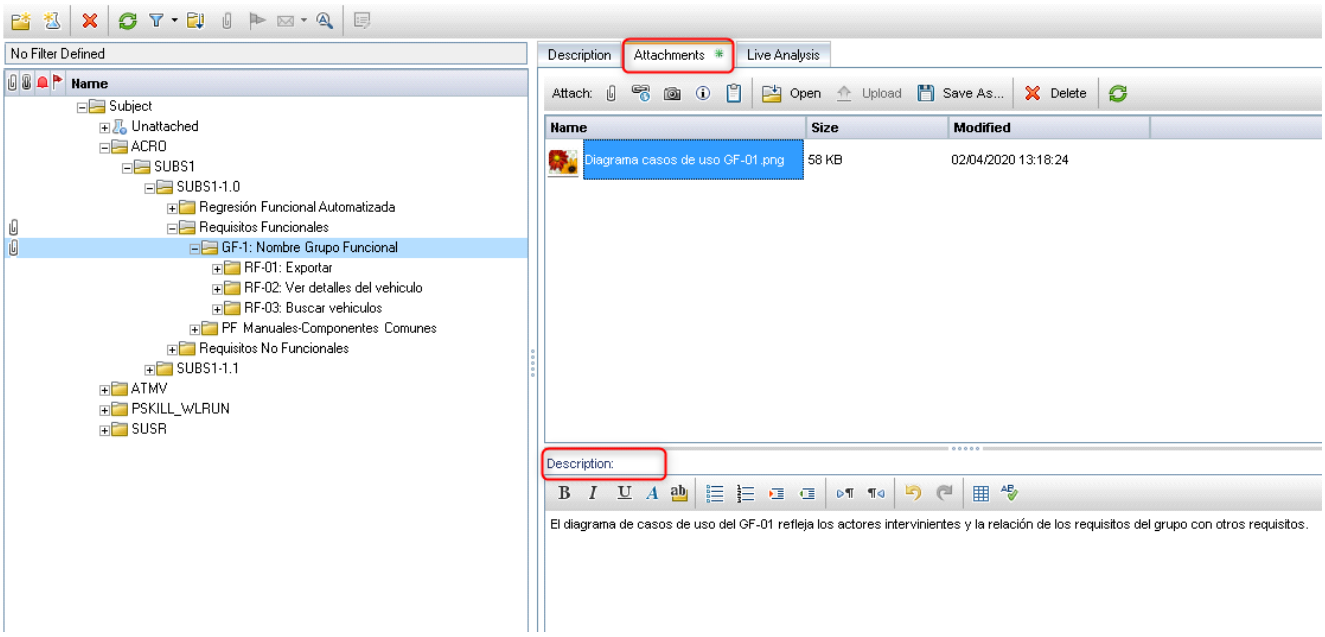
Identificación y descripción de la funcionalidad que realiza el producto.

Si se trata de un producto cerrado y el producto dispone de una especificación de requisitos, se pueden poner referencias directas a esa especificación así como adjuntar los archivos oportunos en la pestaña "Attachments".



Puede organizarse el apartado en subapartados por Grupos Funcionales (GF-XX). En ese caso se deberían insertar los GF como tipo "**Group**" (Requisito de grupo).

Cada **Grupo Funcional (GF-XX)** debe llevar como adjunto su **diagrama de casos de uso** junto con una breve descripción del mismo como se puede ver en la imagen adjunta.



Por último se introducirán los requisitos funcionales como tipo "**Functional**" y que se identificarán como RF-XX: Nombre del RF.

Los **campos obligatorios** a introducir son:

- Priority (1 a 5). En los requisitos de tipo funcional establece el nivel de detalle del plan de pruebas a realizar. 5 representa la mayor prioridad (en al "Anexo 03.05 Procedimiento de Pruebas Funcionales" se detallan los criterios para la priorización de requisitos). **Al menos un Requisito Funcional de los definidos deberá tener prioridad 5**.
- Target Release
- Target Cycle. Únicamente los requisitos de tipo funcional o rendimiento deben tener asignado el ciclo. Este campo debe estar vacío para los requisitos de carpeta y grupo.
- Description

Dentro del campo "Description" se puede estructurar con la siguiente información:

- Identificación de la funcionalidad o caso de uso
- Descripción de la funcionalidad
- Entradas: información que aporta el actor del caso de uso.
- Salidas: información o mensaje que el sistema le envía al actor del caso de uso al finalizar su ejecución.
- Precondiciones: estado necesario en el que debe encontrarse el sistema para que pueda ejecutarse el caso de uso.
- Postcondiciones: estado alterado en el sistema, tras la ejecución del caso de uso.

Pudiéndose incluir la siguiente información:

- Usuarios y sistemas externos que en él intervienen
- Descripción de los pasos que describen la interacción entre los actores y el sistema.
- Comprobación de validez de las entradas
- Secuencia de operaciones
- Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)
- Relaciones entre entradas y salidas (secuencias de entradas y salidas, fórmulas para la conversión de información)
- Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)

2.2.4.2 Requisitos no funcionales

A este nivel se introducirá un requisito tipo "***Business***" con el Name = Requisitos No Funcionales.

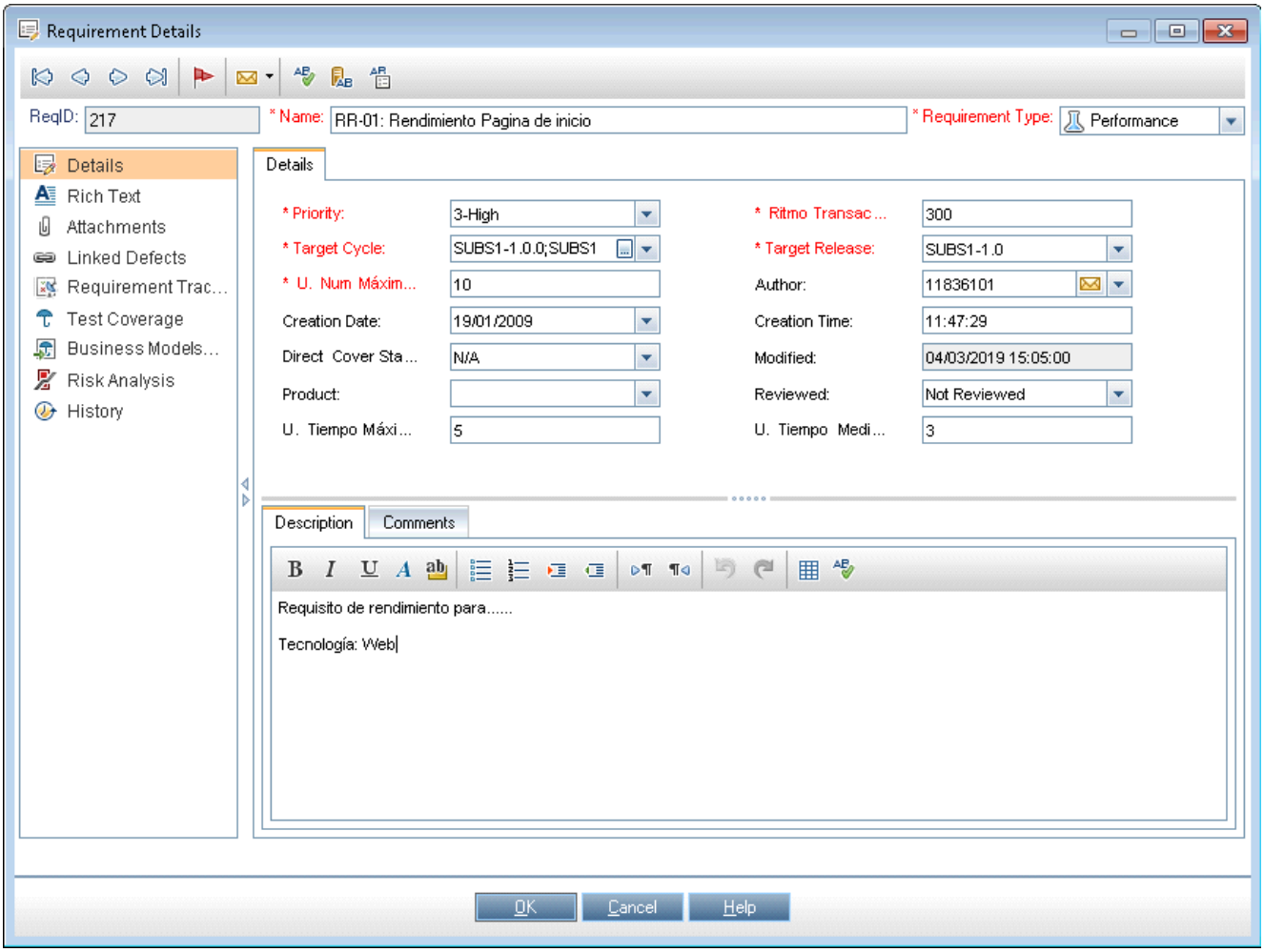
Colgando de este nivel se introducirán los Requisitos de Rendimiento así como cualquier otro tipo de requisitos no funcionales que apliquen de los indicados más abajo.

2.2.4.2.1 Rendimiento

A este nivel se introducirá un requisito tipo "***Group***" con el Name = Rendimiento.

Colgando de este nivel se introducirán cada uno de los requisitos de Rendimiento que se hayan identificado de acuerdo al Anexo 03.07 Procedimiento de Pruebas de Rendimiento. Éstos serán de tipo "Performance", seguirán la nomenclatura "**RR-XX Nombre del Requisito de rendimiento**" y además de los campos ya comentados de los requisitos funcionales deberán incorporar los siguientes campos:

- En la pestaña Details.
  - Número Máximo de Usuarios
  - Tiempo Medio de Respuesta (sg.) (opcional)
  - Tiempo Máximo de Respuesta (sg.) (opcional)
  - Ritmo transaccional, es decir, el número de veces que se ejecuta el proceso de negocio (script) por hora y por el total de usuarios del RR.
- En la pestaña Description.
  - Tecnología (Web, Webservice, …)



2.2.4.2.2 RNF Usuario de control

Se indicará en el Requisito No Funcional “Usuario de Control” los sistemas a los que accederá el usuario de control de la aplicación

2.2.4.2.3 RNF Accesibilidad

En los casos en la aplicación tenga interfaz de usuario de cualquier tipo (web, dispositivo móvil etc.), ésta deberá cumplir el Real Decreto vigente en cada momento.

2.2.4.2.4 RNF Auditoría

Se registrarán, mediante el sistema de auditoría de la DGT (componente común desarrollado al efecto), las acciones realizadas por los usuarios de forma que sea posible determinar el usuario en el momento temporal en las que fueron realizadas.

2.2.4.2.5 RNF Control de acceso

Todos los módulos controlarán el acceso a la información a través de un sistema de gestión de usuarios y deberán implementar la autorización a través de las consideraciones que marca la especificación de seguridad para aplicaciones J2EE (de obligado cumplimiento).

2.2.4.2.6 RNF Usabilidad

La aplicación tendrá la capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y conforme a estándares, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

2.2.4.2.7 RNF Fiabilidad

La aplicación deberá ser capaz de gestionar eventos externos ajenos a la operativa habitual de la aplicación y recibiendo valores erróneos en sus ejecuciones.

2.2.4.2.8 RNF Disponibilidad

El sistema deberá funcionar en régimen 24x7 y estar dotado de los mecanismos necesarios para recuperarse ante fallos.

2.2.4.2.9 RNF Multi-idioma

La aplicación deberá poder mostrarse en los idiomas que se definan para su uso, como por ejemplo los oficiales de España (castellano, catalán, gallego, euskera y valenciano), en inglés o francés, gestionando las peculiaridades de cada idioma y la relación con sistemas y aplicaciones de terceros. Si la aplicación se incluye dentro de la **sede** electrónica y/o se dirige al **ciudadano** deberá cumplir los requisitos de multi-idioma.

2.2.4.2.10 RNF Seguridad

En los servicios desarrollados por las aplicaciones (independientemente de la tecnología: servicios web, ejb, etc.) para ser ofrecidos a terceros, las operaciones deberán estar protegidas mediante las indicaciones recogidas en las especificaciones de seguridad y de desarrollo de servicios de la DGT.

2.2.4.2.11 RNF Configuración de las aplicaciones

Las aplicaciones deben estar configuradas en todos los puntos de variación tanto a nivel de código como a nivel de parámetros de entorno.

2.2.4.2.12 RNF Cohesión

Las aplicaciones deben seguir una partición funcional de código en módulos “vertical” que segmente las distintas áreas de negocio y/o persistencia que cubran en componentes independientes (o dicho de otra forma, los módulos o componentes se generaran en forma cohesiva por áreas funcionales en un grano fino). La partición horizontal ya viene marcada por la estructura de proyecto de las especificaciones (módulos POJO, DAO, etc.).

2.2.4.2.13 RNF Almacén de datos

Se deberán elaborar los ficheros de intercambio basándose en el Documento de Interfaz que proporcionará el Almacén de Datos.

2.2.4.2.14 RNF Geolocalización

La aplicación deberá ser capaz de realizar un posicionamiento geográfico, indicando como atributos el/los métodos y precisión del posicionamiento.

2.2.4.2.15 RNF Eficiencia

Desde el lado cliente, la aplicación deberá mostrar un consumo contenido de recursos del equipo en el que esté desplegada, estos recursos serán referentes al consumo de CPU, RAM, consumo de ancho de banda y batería (en caso de tratarse de un dispositivo móvil).

2.2.4.2.16 RNF Almacenamiento

Deberá indicarse la ubicación exacta donde se guardan los datos manejados por la aplicación, ya sean datos almacenados en el servidor o en el cliente (ficheros temporales, caché, FS, etc). Quedan excluidos los indicados en el RNF Almacén de Datos.

2.2.4.2.17 RNF Portabilidad

La aplicación deberá mostrar un funcionamiento homogéneo en cualquier plataforma y versión de Sistema Operativo.

2.2.4.2.18 RNF Compatibilidad

La aplicación deberá mostrar un funcionamiento homogéneo en cualquier navegador.

2.2.4.2.19 RNF Interoperabilidad

La aplicación tendrá la capacidad de intercambiar información y utilizar la información intercambiada con uno o más componentes o sistemas externos.

2.2.4.2.20 RNF Información sensible

No se deberá mostrar información cuya revelación, alteración, pérdida o destrucción puede producir daños importantes al propietario de la misma, como pueden ser datos personales y bancarios, contraseñas, domicilio en algunos casos, etc.

2.2.4.3 Requisitos de interfaces

A este nivel se introducirá si aplica un requisito tipo "***Business***" con el Name = Requisitos de Interfaces.

De este nivel colgarán:

2.2.4.3.1 Interfaces de usuario

Se creará como tipo "Group" y de él colgarán cada uno de los requisitos que apliquen.

2.2.4.3.1.1 RNFI-YY:

Se creará del tipo "No Functional".

Especificar si aplica los requisitos de interfaz de usuario del producto. Si se establece alguna convención a la hora de representar elementos en pantalla, también se definirá en este apartado

2.2.4.3.2 Interfaces de/con otros sistemas: Ofrecidas

Se creará como tipo "Group" y de él colgarán cada uno de los requisitos que apliquen.

2.2.4.3.2.1 RNFI-XX:

Se creará del tipo "No Functional".

Para cada producto de software puede especificarse lo siguiente:

- Identificación (nombre y versión) y descripción de la interfaz que se ofrece
- Propósito del interfaz
- Definición del interfaz: contenido y formato

2.2.4.3.3 Interfaces de/con otros sistemas: Consumidas

Se creará como tipo "Group" y de él colgarán cada uno de los requisitos que apliquen.

2.2.4.3.3.1 RNFI-XY:

Se creará del tipo "No Functional".

Para cada producto de software con el que se interactúa, debe especificarse lo siguiente:

- Identificación (nombre y versión) y descripción del producto software utilizado
- Propósito del interfaz
- Definición del interfaz: contenido y formato

2.2.4.3.4 Interfaces de comunicaciones

Se creará como tipo "Group" y de él colgarán cada uno de los requisitos que apliquen.

2.2.4.3.4.1 RNFI-ZZ:

Describir los requisitos de interfaces de comunicación si hay comunicaciones con otros sistemas o subsistemas y cuáles son los protocolos (ej, web service, restful, etc.) de comunicación -> Maqueta servicios interfaz

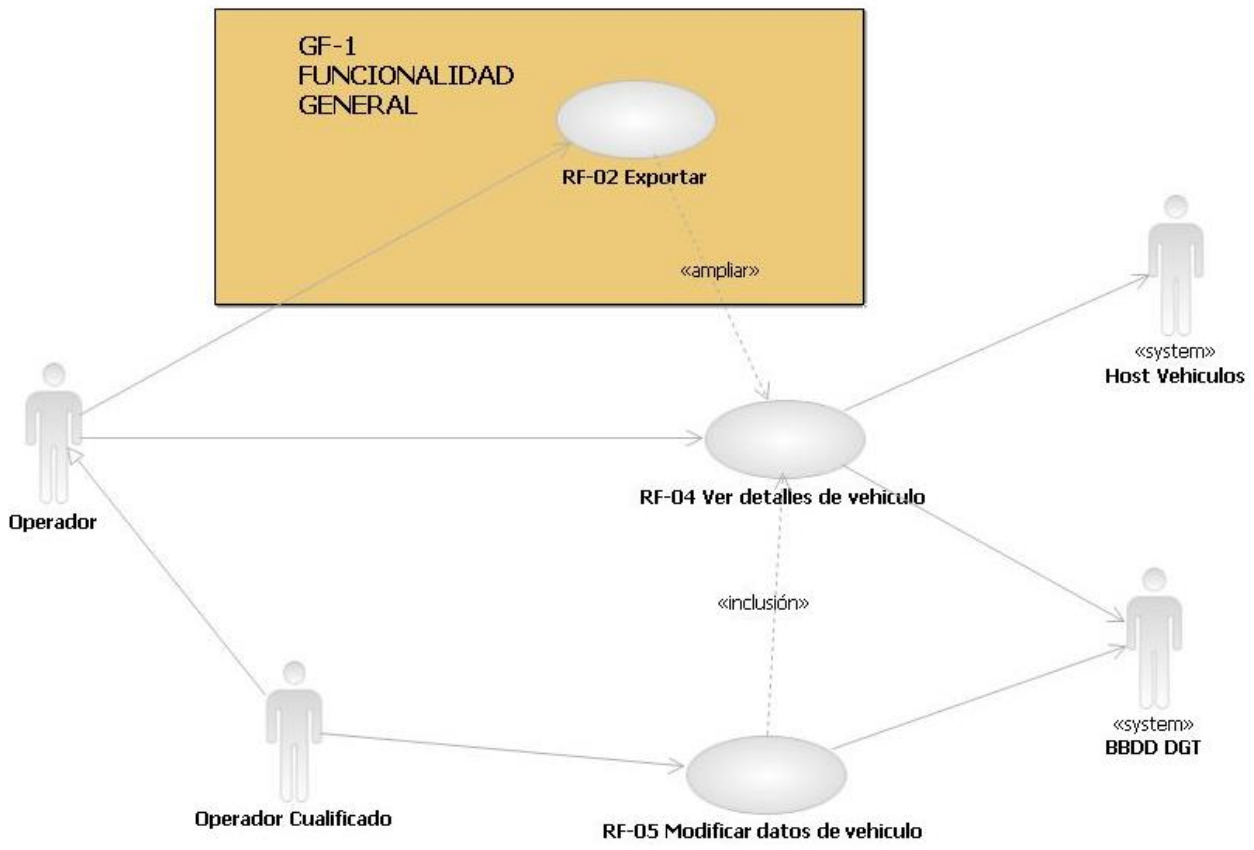
2.2.5 Trazabilidad de Requisitos

En la pestaña "**Requirements Traceability**" del módulo "Requirements" se definen las relaciones entre dos o más requisitos.

Cuando las necesidades o características del requisito cambien, se podrá rastrear y monitorizar el impacto de dichos cambios.

Para trazar las relaciones entre requisitos, deberemos determinar "qué requisitos pueden verse afectados ante los cambios en un determinado requisito"

Ejemplo:



En este caso, cualquier cambio en el Requisito "*RF-04: Ver detalles del vehículo*" afectaría a los requisitos "*RF-02: Exportar*" y "*RF-05: Modificar datos del vehículo*".

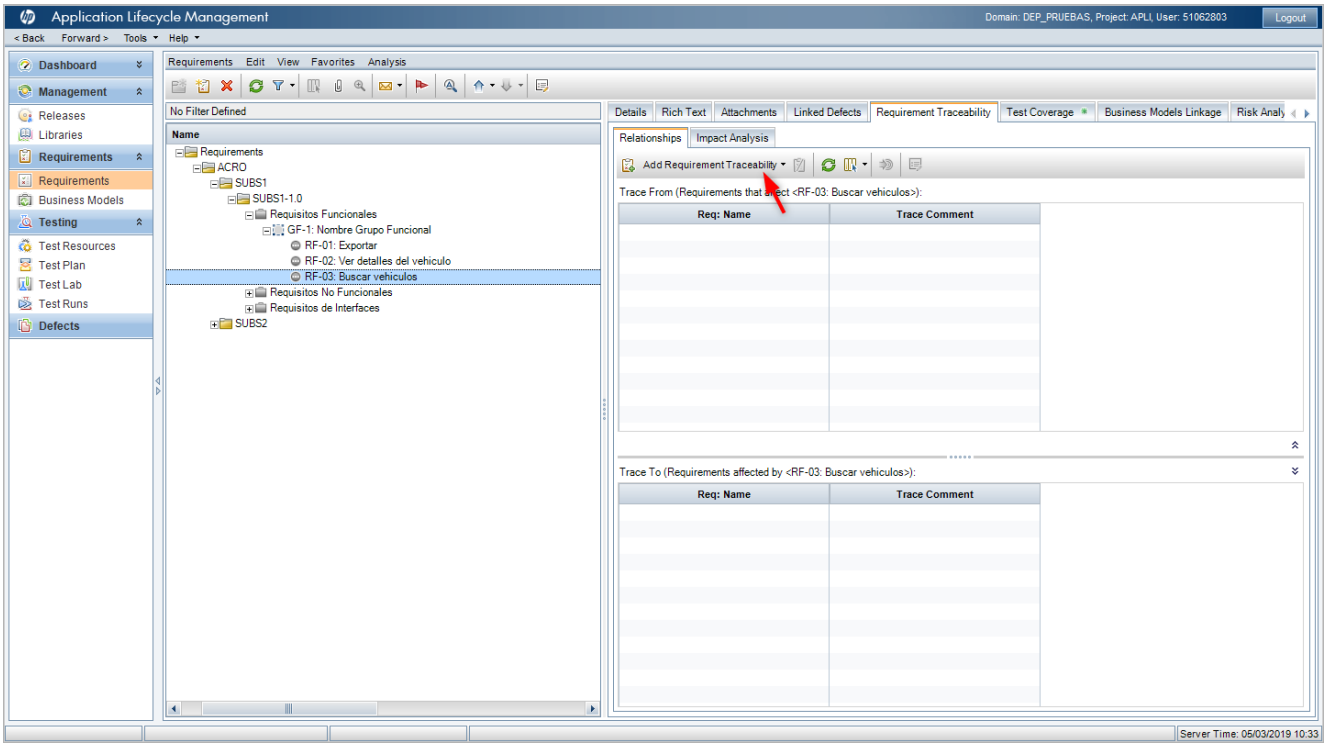
Dicha relación en ALM se establece mediante la opción "**Trace To (requirements affected by)**" que indica **los requisitos que se ven afectados** ante cualquier cambio en el requisito seleccionado.

Cuando en ALM se establece esta relación, automáticamente en el requisito afectado se establece la relación "**Trace From (requirements that affect)**" que indica **los requisitos que afectan** a dicho requisito.

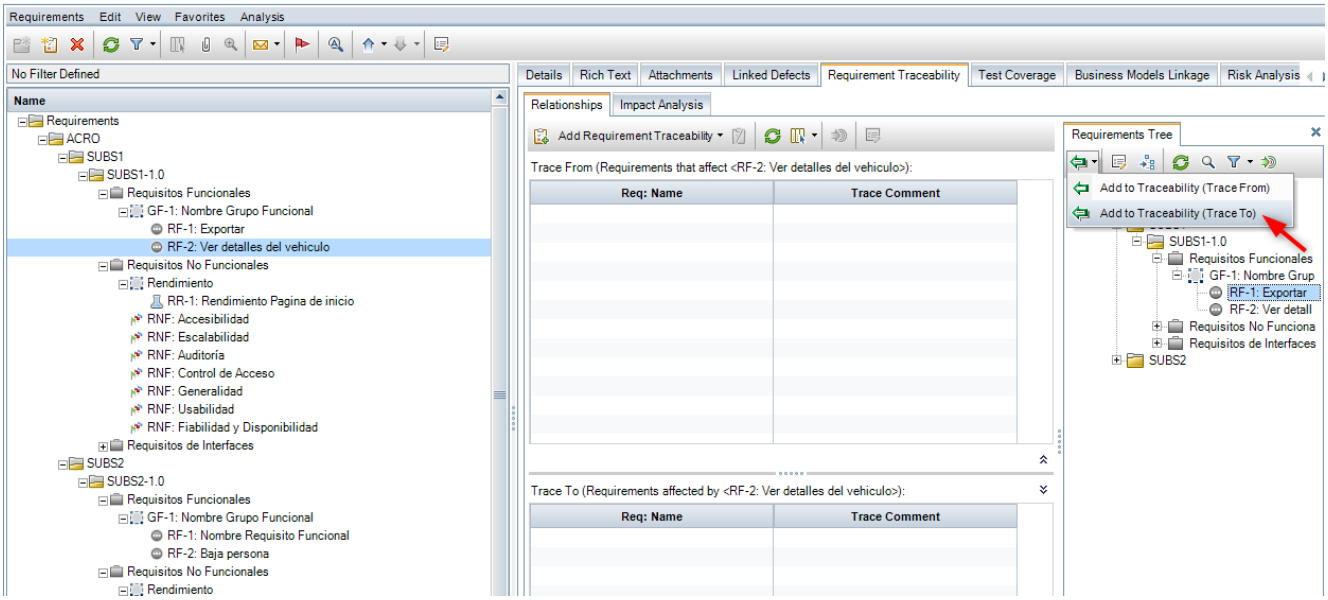
Los pasos a realizar desde la vista "**View/Requeriment Details**" son:

1. Seleccionar el requisito y pulsar en la pestaña "**Requirements Traceability**"

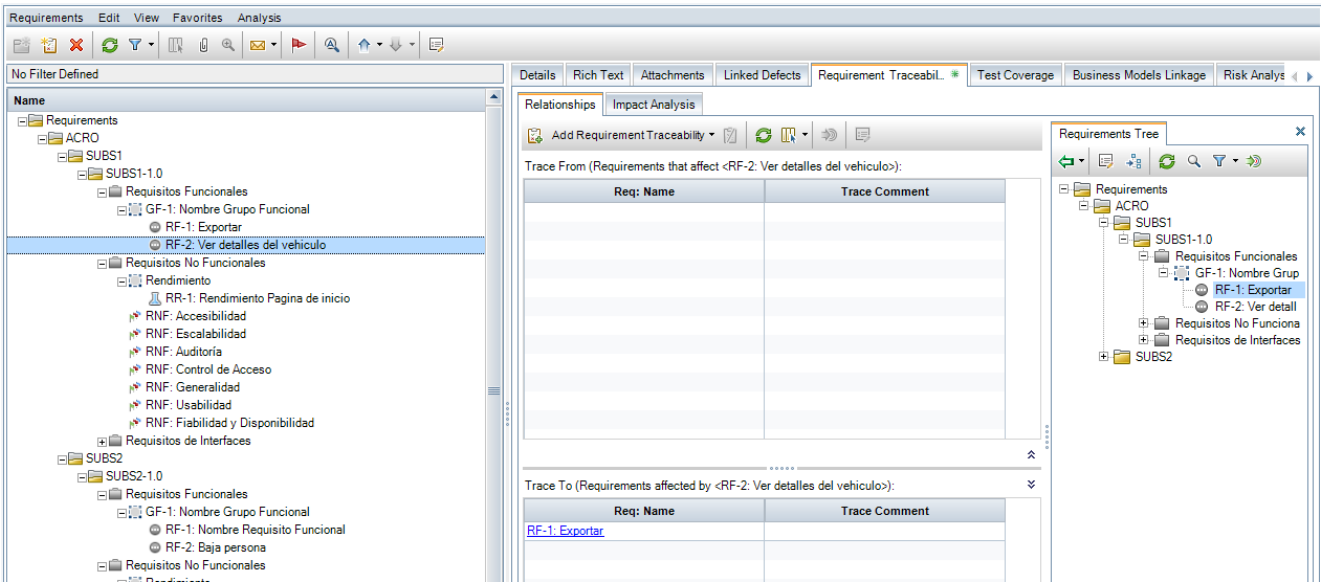
En nuestro ejemplo, seleccionamos el requisito "*RF-02: Ver Detalles del vehículo*"



2. Pulsar en el botón "Add to Traceability (Trace To)"



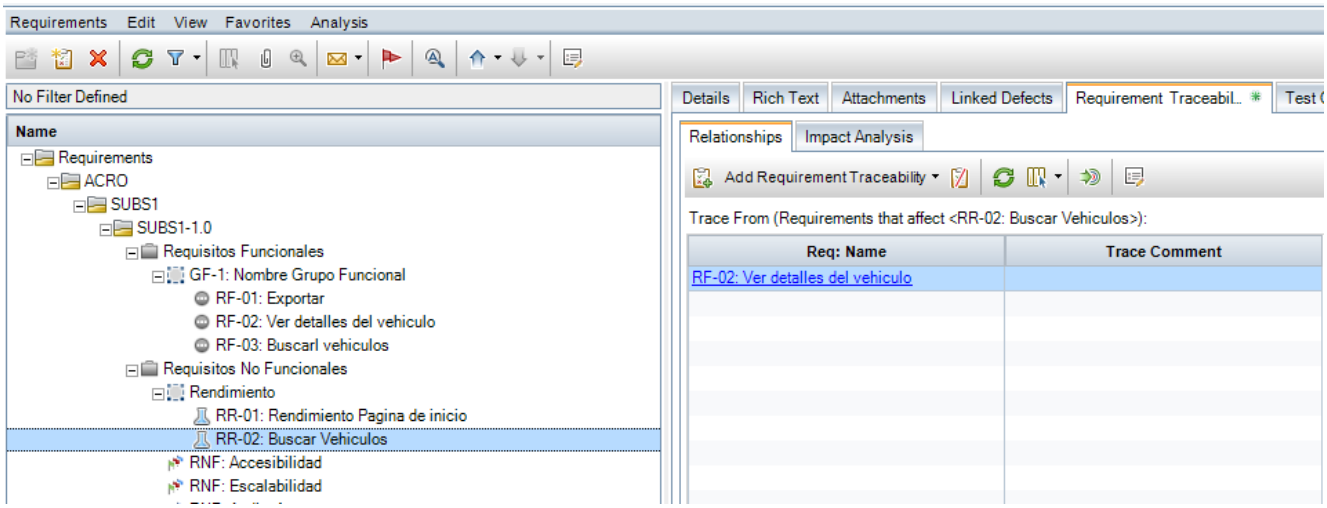
En nuestro ejemplo, seleccionar el requisito "**RF-01: Exportar**" y la opción "**Trace To**", debido a que si cambia el requisito funcional "*RF_02: Ver detalles del Vehículo*" puede verse afectado el requisito "*RF_01: Exportar*".



La relación entre un requisito funcional y uno de rendimiento se realizaría de la siguiente forma ya que si cambia el requisito funcional el requisito de rendimiento se ve afectado:

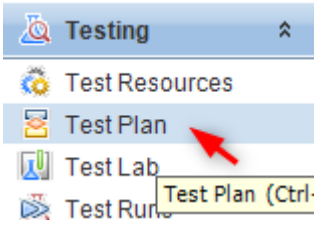
El requisito funcional tendrá una relación de "**Trace To (requirements affected by)**" con el de rendimiento

El requisito de rendimiento tendrá una relación de "**Trace From (requirements that affect)**" con el funcional.



2.3 Módulo Test Plan

2.3.1 Objetivo



En el módulo Test Plan de ALM es donde se va a realizar la documentación de los planes de pruebas funcionales (manuales y automatizadas) y de rendimiento de los aplicativos.

2.3.2 Estructura

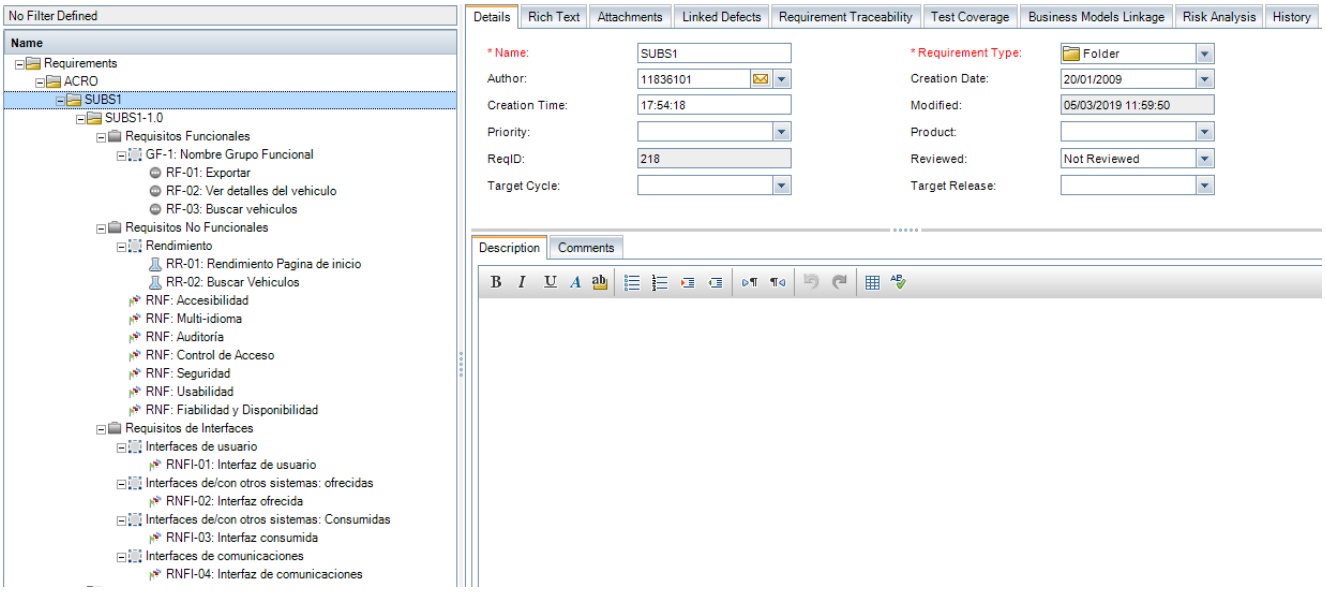
Una vez introducidos los requisitos en el módulo "**Requirements**" se deben crear su correspondencia en el módulo **"Test Plan"**.

Desde ALM se puede generar automáticamente la estructura de carpetas para que cubran todos los requisitos especificados en el módulo "**Requirements**".

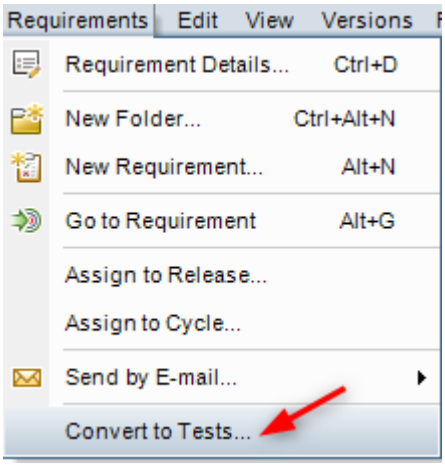
Los pasos a realizar son:

1. Ir al módulo "**Requirements**"
2. Seleccionar la carpeta con el nombre de la aplicación.

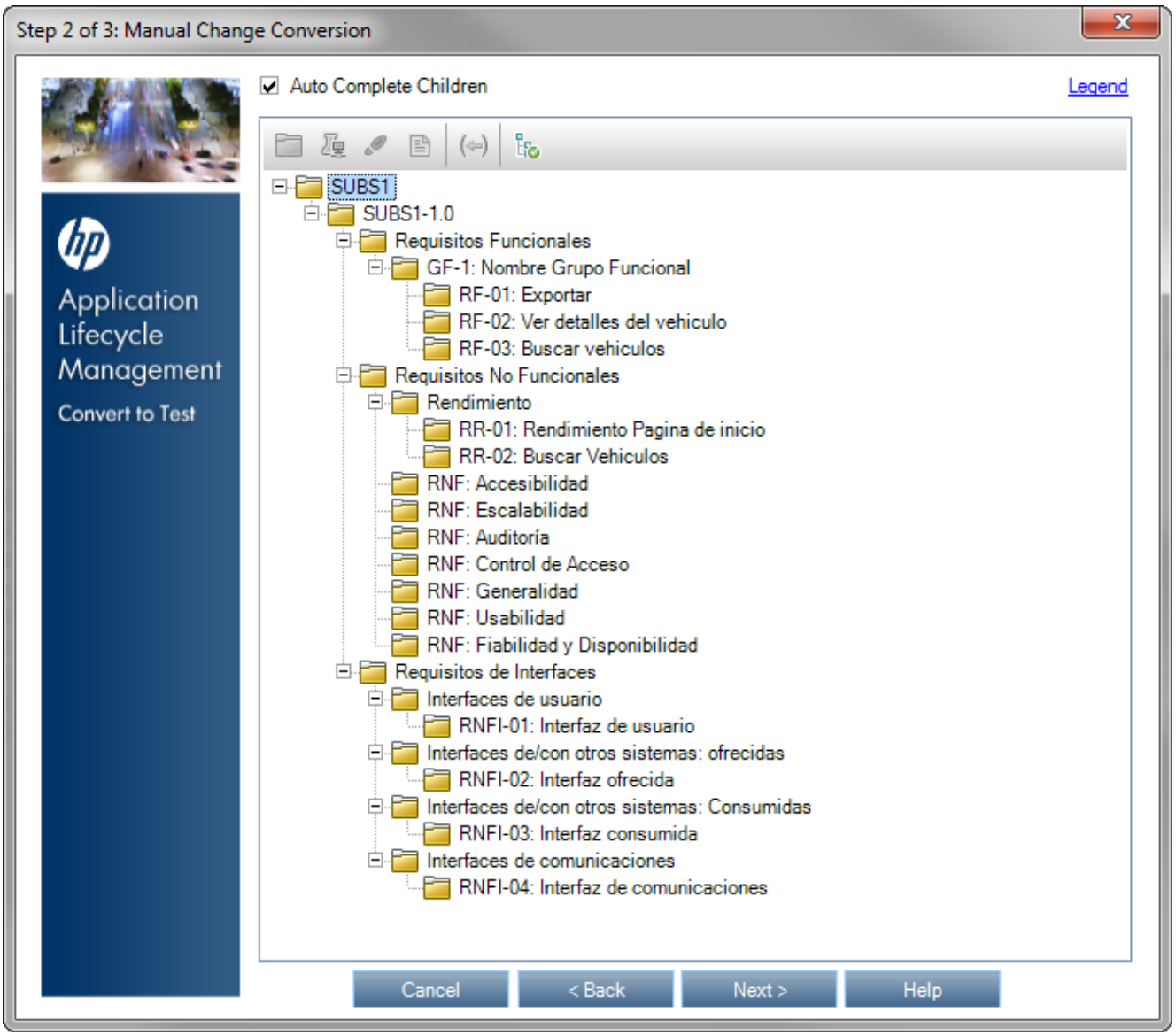
Por ejemplo, la carpeta "**SUBS1**"



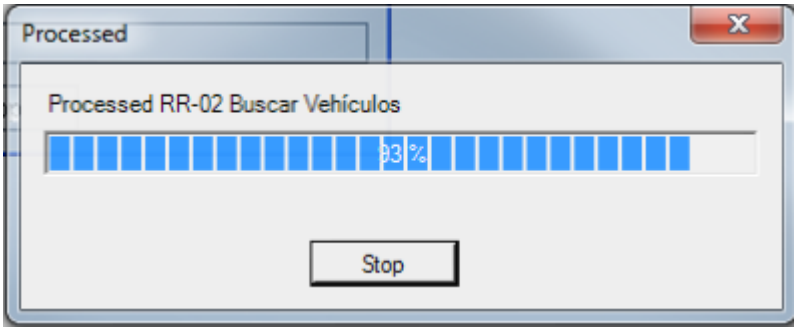
3. Ir al menú y ejecutar "Requirements\Convert to Tests…"



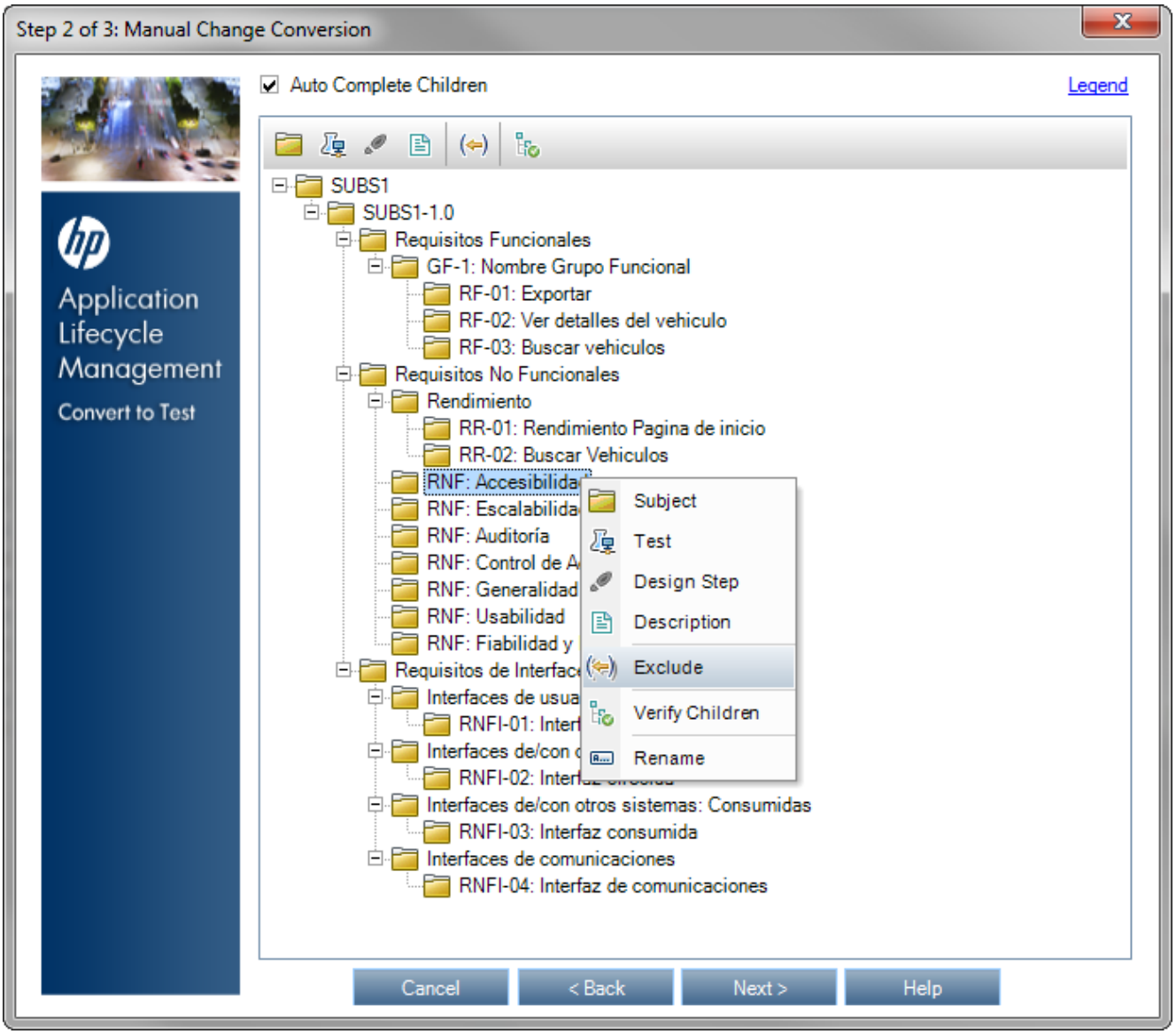
4. Seleccionar la opción "**Convert all requirements to subjects**." Pulsar en el botón "**Next**"



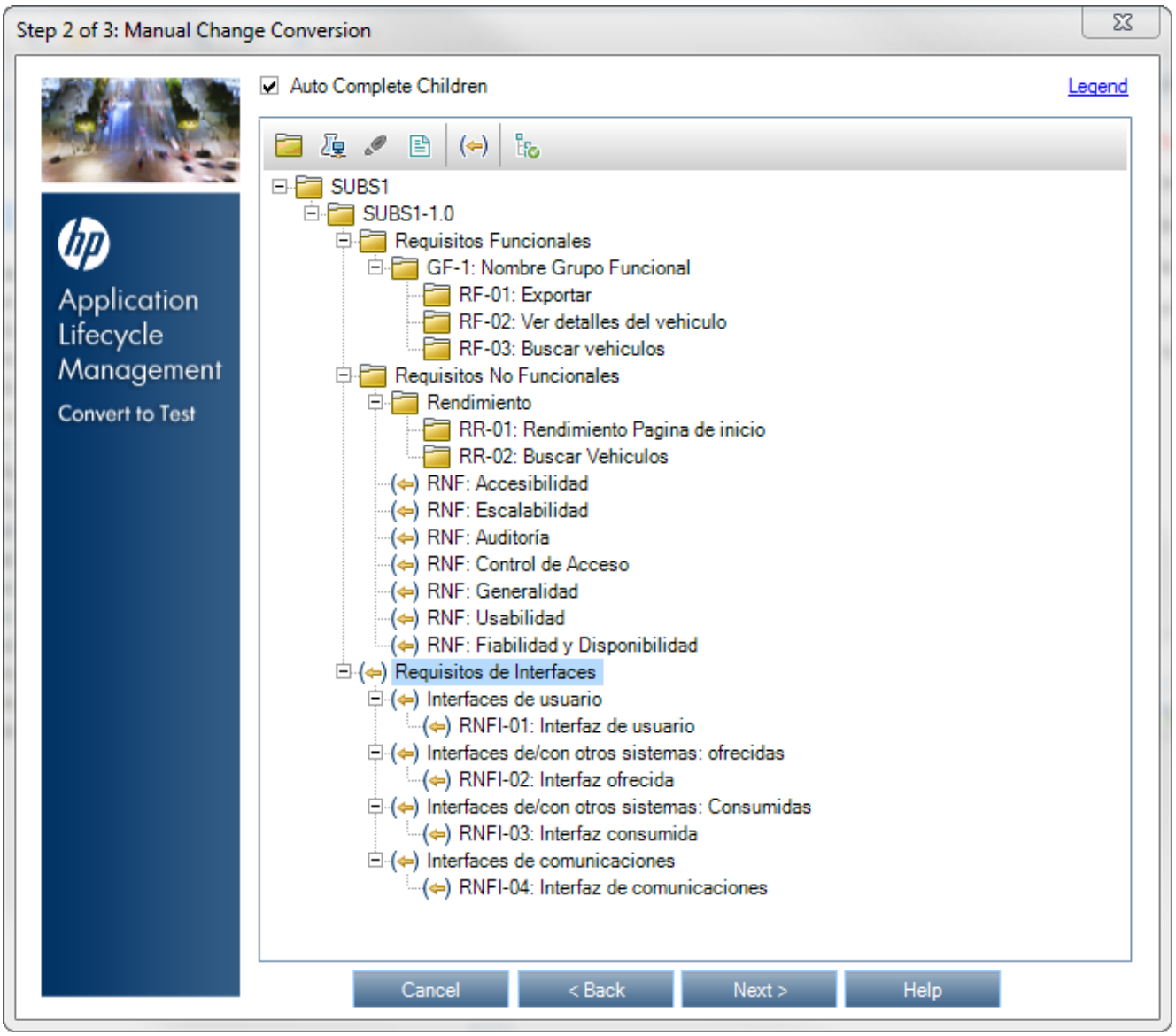
Aparecerá la barra de progreso



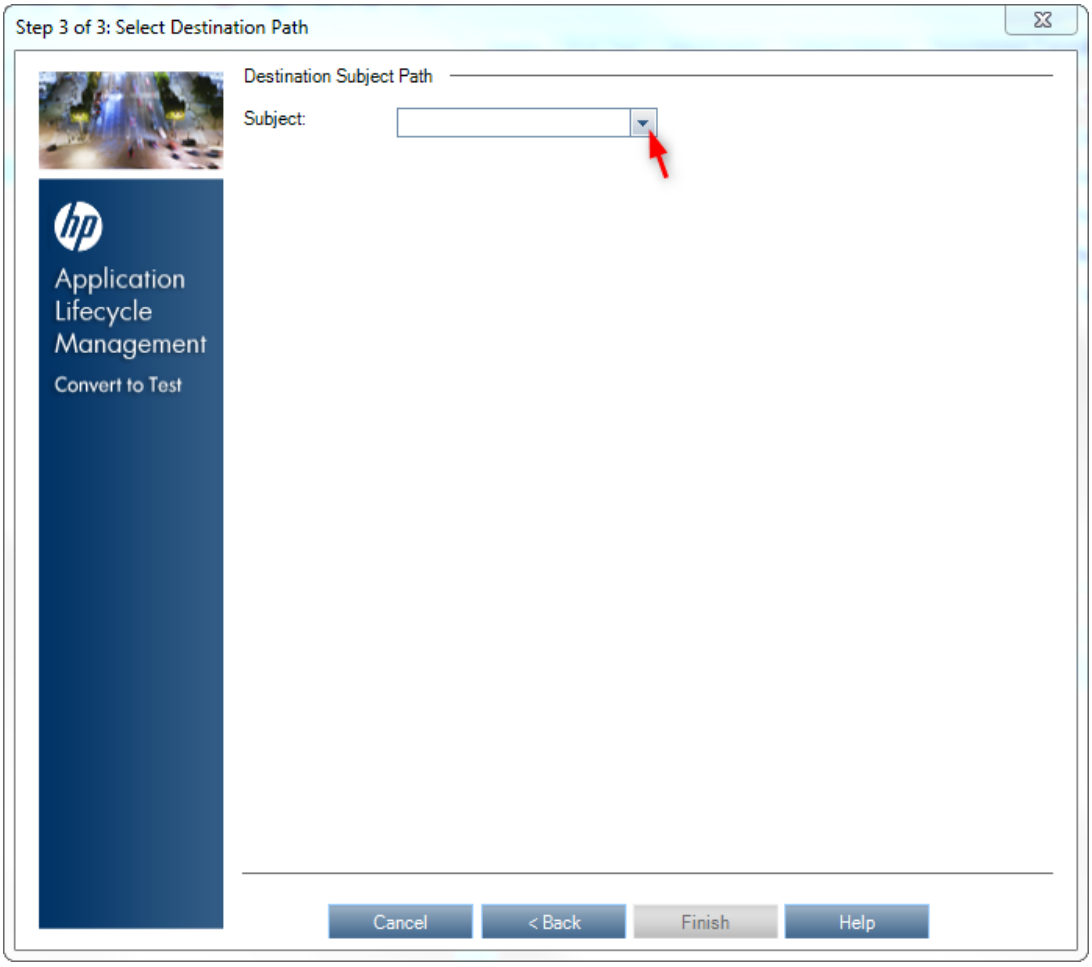
5. Excluimos los Requisitos No funcionales que no sean de Rendimiento así como los Requisitos de Interfaces. Para ello nos situamos sobre cada uno de ellos, pulsamos botón derecho y seleccionamos "**Exclude**"



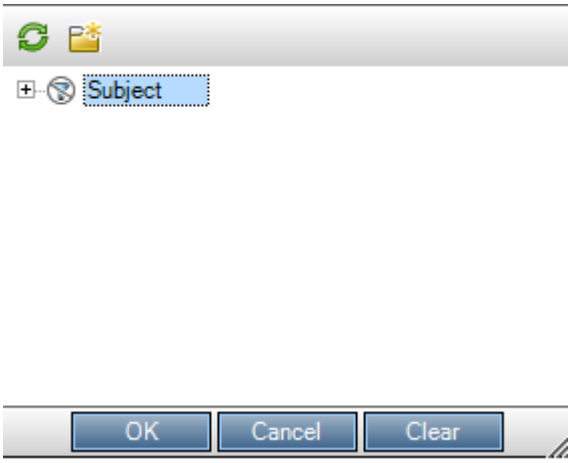
Quedando así:



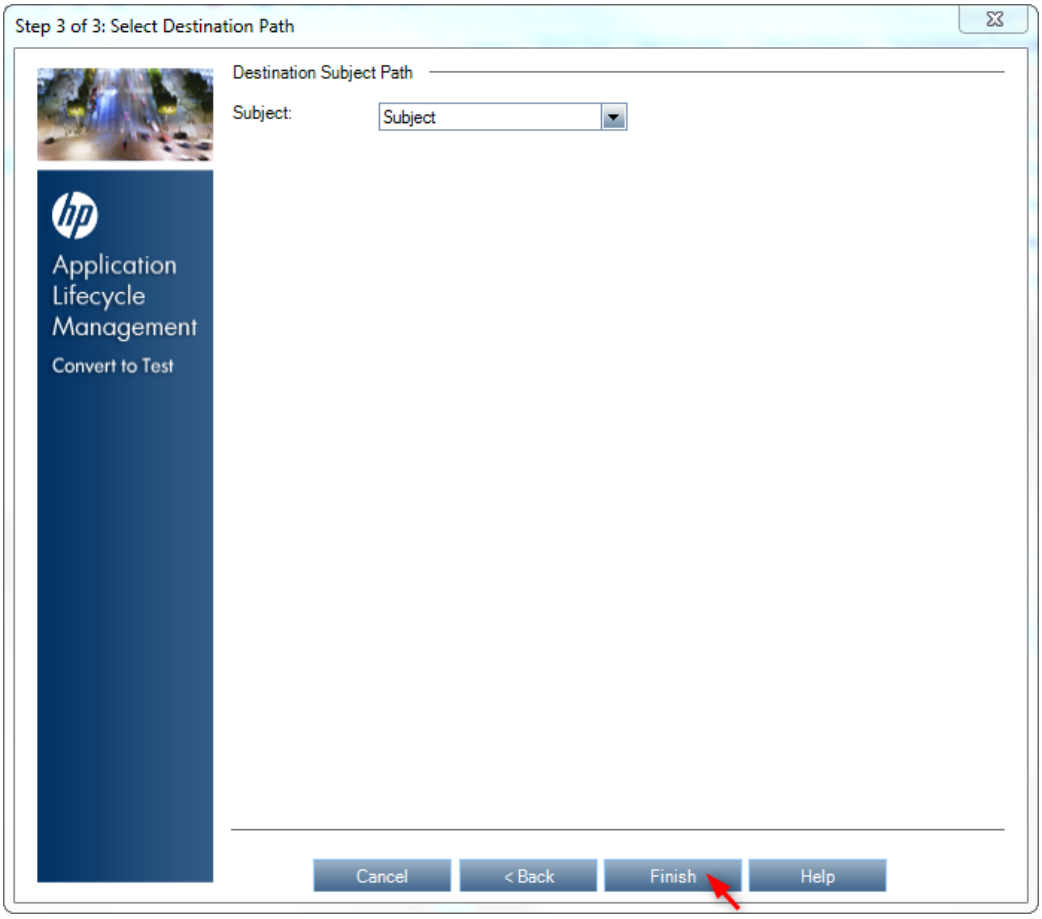
6. Pulsar en el botón "**Next**" y seleccionar la ruta destino pulsando en el botón



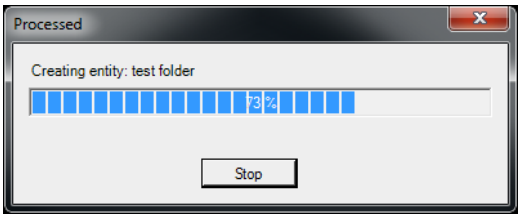
7. Seleccionar la carpeta "**Subject**\" y pulsar en el botón "**OK**"



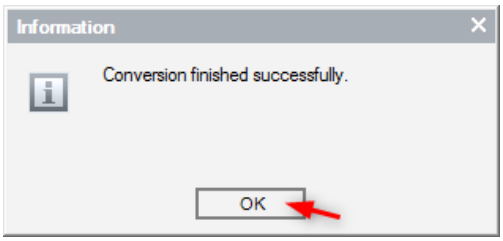
8. Pulsar en el botón "**Finish**"



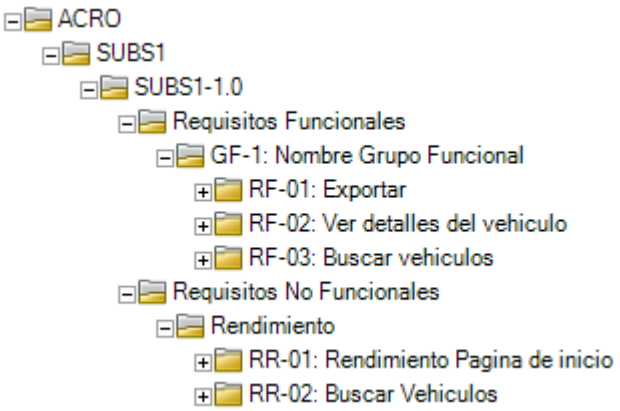
Aparecerá la barra de progreso



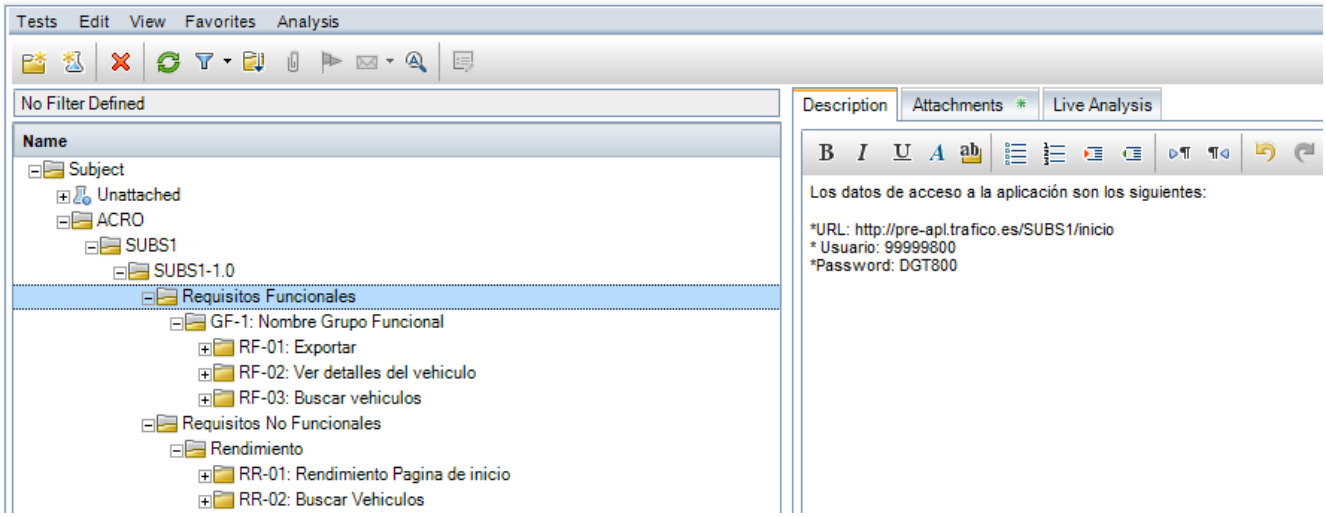
9. Si todo ha ido correctamente, mostrará el siguiente mensaje y se pulsará en el botón "**OK**"



En el módulo "**Test Plan**" se habrá generado la estructura de carpetas correspondiente.



En la pestaña **"Description"** de la carpeta "**Requisitos Funcionales**" debe documentarse la url (o urls) de acceso a la aplicación así como un usuario y password para acceder a la misma.



2.3.3 Ubicación ficheros de datos de entrada para auditorías

Para la auditoría de los planes de prueba la ubicación de los ficheros de datos de entrada deberá estar en la siguiente ubicación:

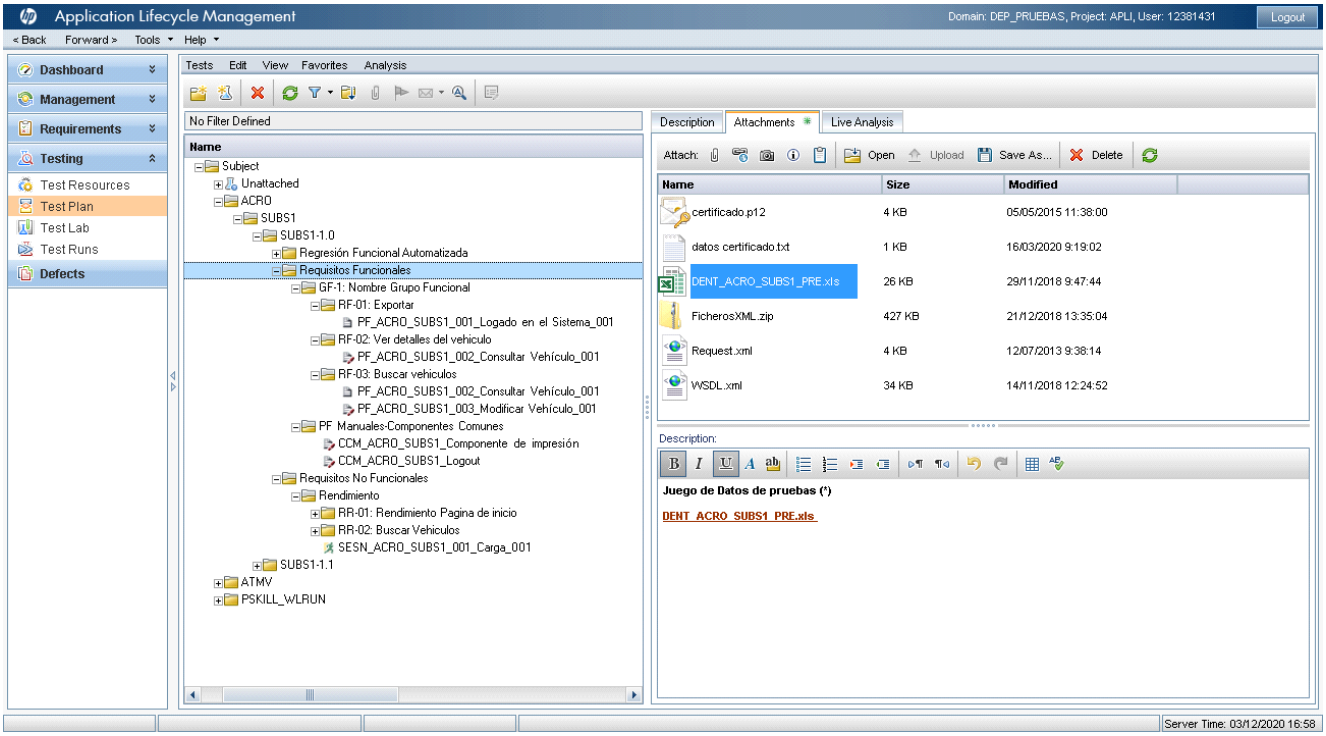
2.3.3.1 Fichero XLS de pruebas funcionales

El fichero con los datos de prueba para las pruebas funcionales deberá estar en la pestaña "**Attachments**" de la carpeta "**Requisitos funcionales**". El fichero XLS debe cumplir la estructura de

pestañas y la nomenclatura especificada en el documento "*Anexo 03.01 Guía de datos de entrada de pruebas*".

En el campo "**Description**" de la pestaña "**Attachments**" del fichero deberá aparecer el nombre del fichero, como se indica en la imagen:

Ejemplo:

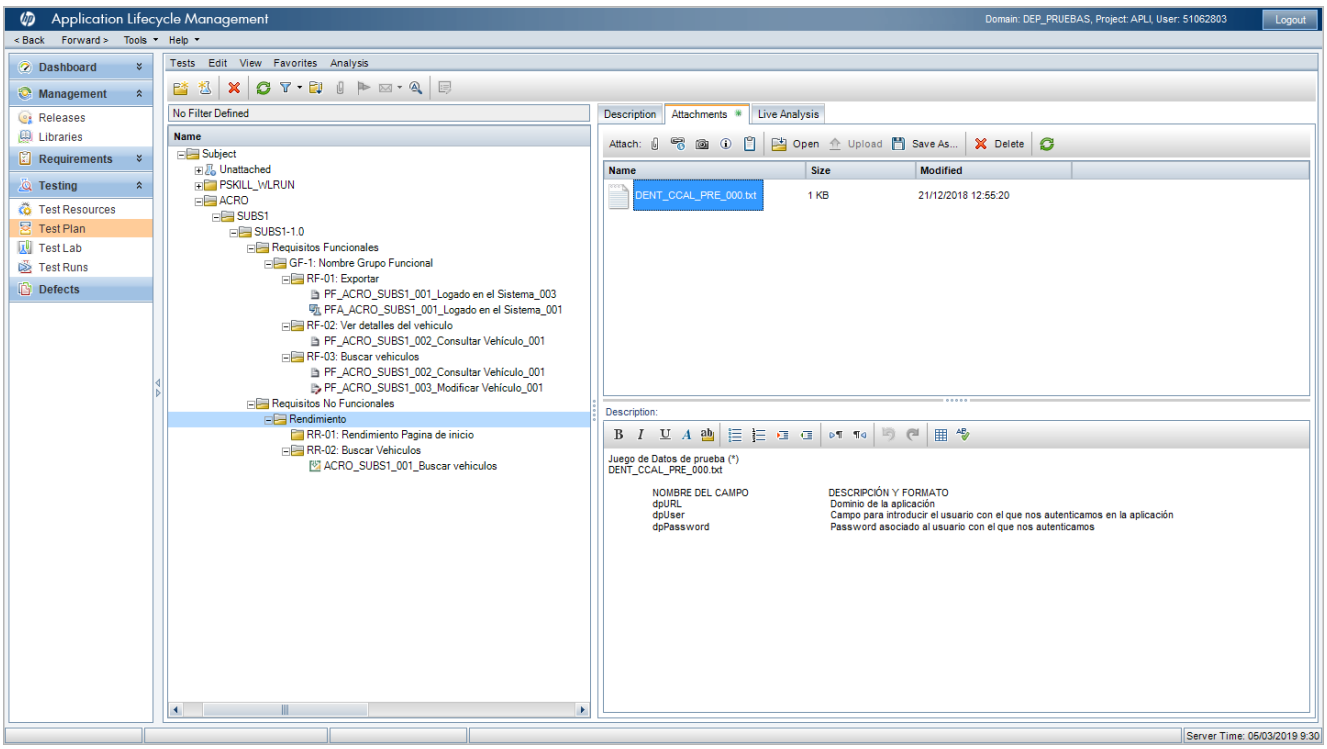


2.3.3.2 Ficheros de pruebas de rendimiento

El fichero con los datos de acceso a la aplicación deberá estar en la pestaña "**Attachments**" de la carpeta "**Rendimiento**" y se ha de incluir en la pestaña "Description" del propio fichero la siguiente información:

- Declaración del nombre del fichero
- Declaración de los campos del fichero

Ejemplo del fichero con los datos de acceso en la pestaña "**Attachments**" de la carpeta "**Rendimiento**":

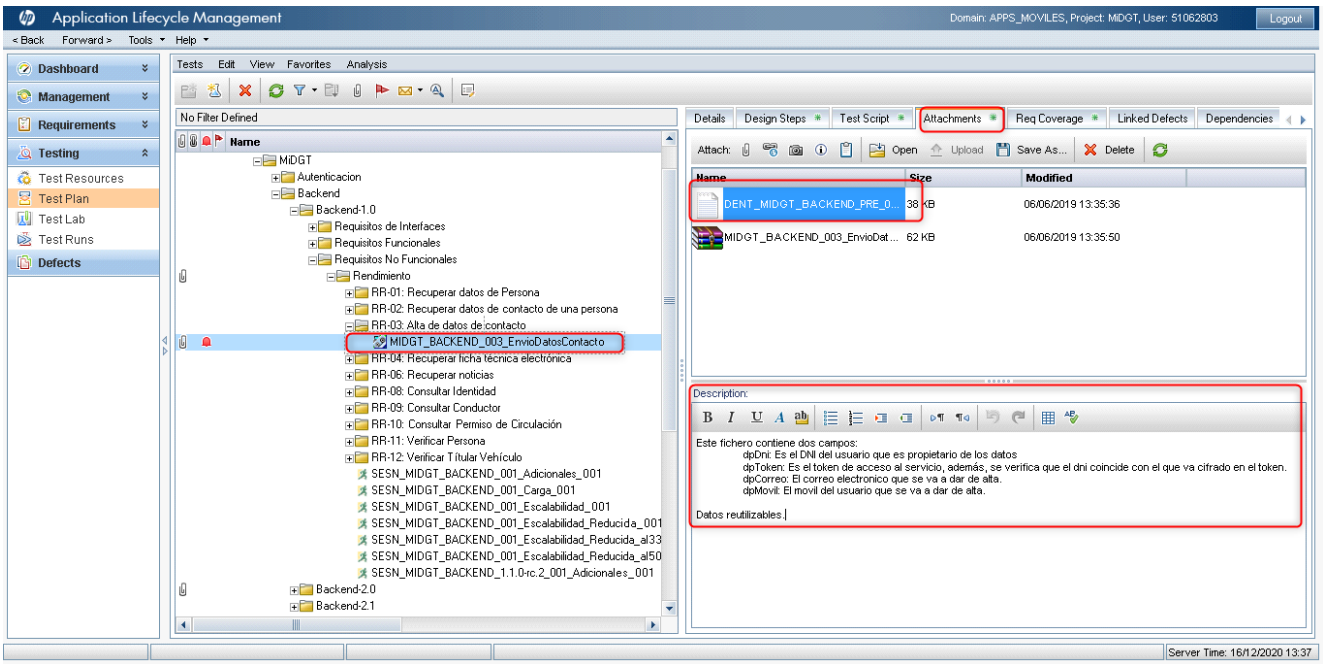


Por otro lado los ficheros con los datos de las pruebas de rendimiento deberán estar en la pestaña "**Attachments**" de los casos de prueba. La declaración de los ficheros y sus campos debe realizarse en el campo "**Description**" de la pestaña "**Attachments**" del fichero tal y como se indica en las siguientes imágenes y cumplir la nomenclatura especificada en el documento "*Anexo 03.01 Guía de datos de entrada de pruebas*".

En el campo "**Description**" de la pestaña "**Attachments**" del fichero deberá aparecer:

- Declaración del nombre del fichero
- Declaración de los campos del fichero
- Descripción de las dependencias de cada uno de los campos con respecto a los demás parámetros.
- Si los datos son reutilizables o consumibles.

Ejemplo de declaración de fichero de datos de un *script*:



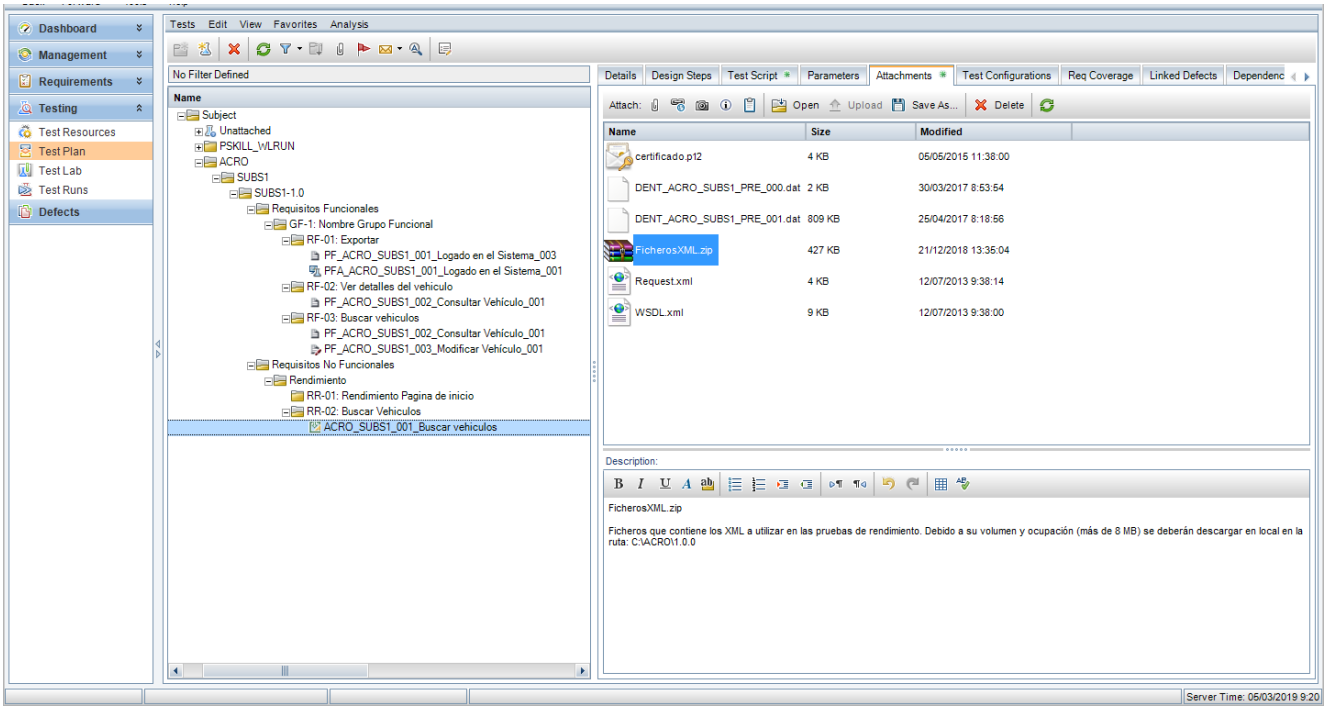
2.3.3.3 Otros ficheros

Además de los ficheros con los datos a utilizar en las pruebas en algunas ocasiones necesitamos de otro tipo de ficheros como:

- Certificados y sus datos asociados
- Proyecto SoapUI en servicios web
- XML de entrada y salida y llamada SOAP en servicios web
- fichero YAML o JSON con la definición del API REST (Swagger – Open API 2.0)
- Proyecto Postman para servicios REST
- documentos explicativos para realizar algún tipo de procedimiento previo (en formato Word o PDF)
- .ZIP (con documentación variada)
- Otros ficheros que puedan ser interesantes

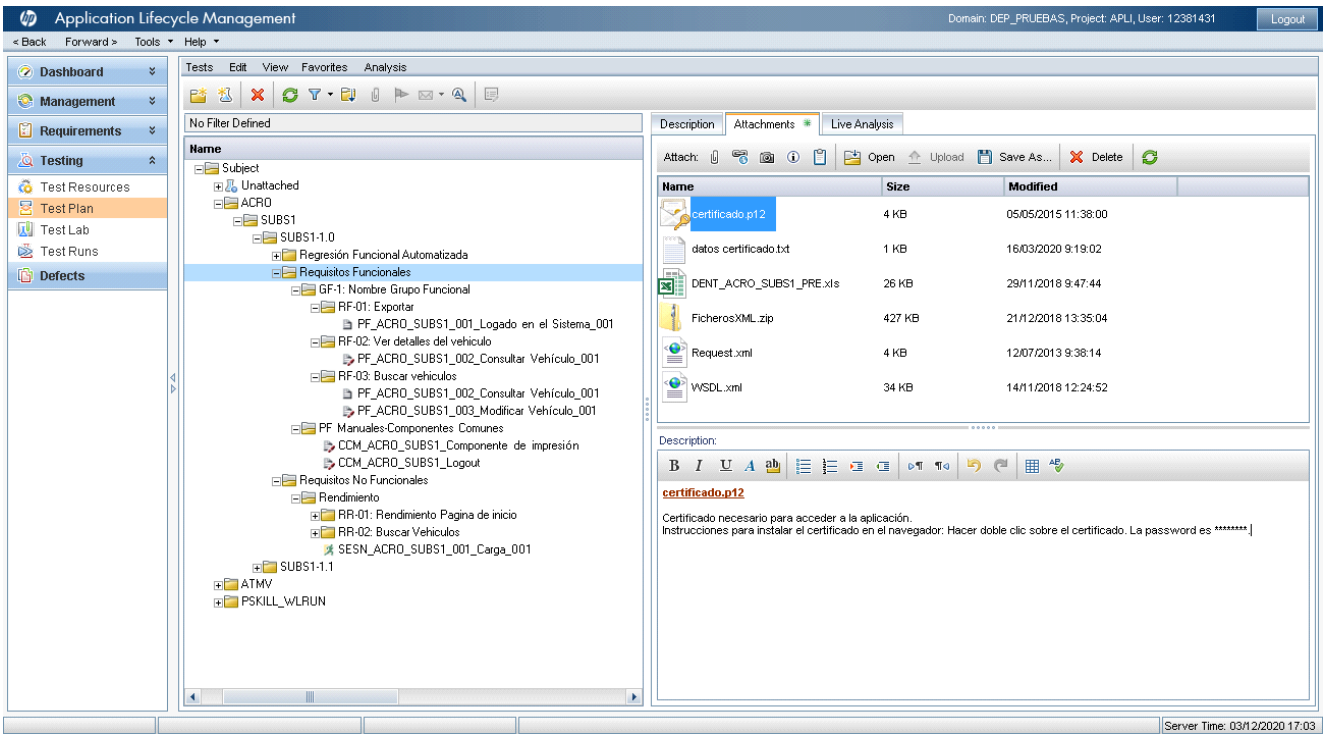
2.3.3.3.1 Pruebas de rendimiento

En estos casos los ficheros y su declaración se deberán anexar al nivel del caso de prueba de rendimiento:

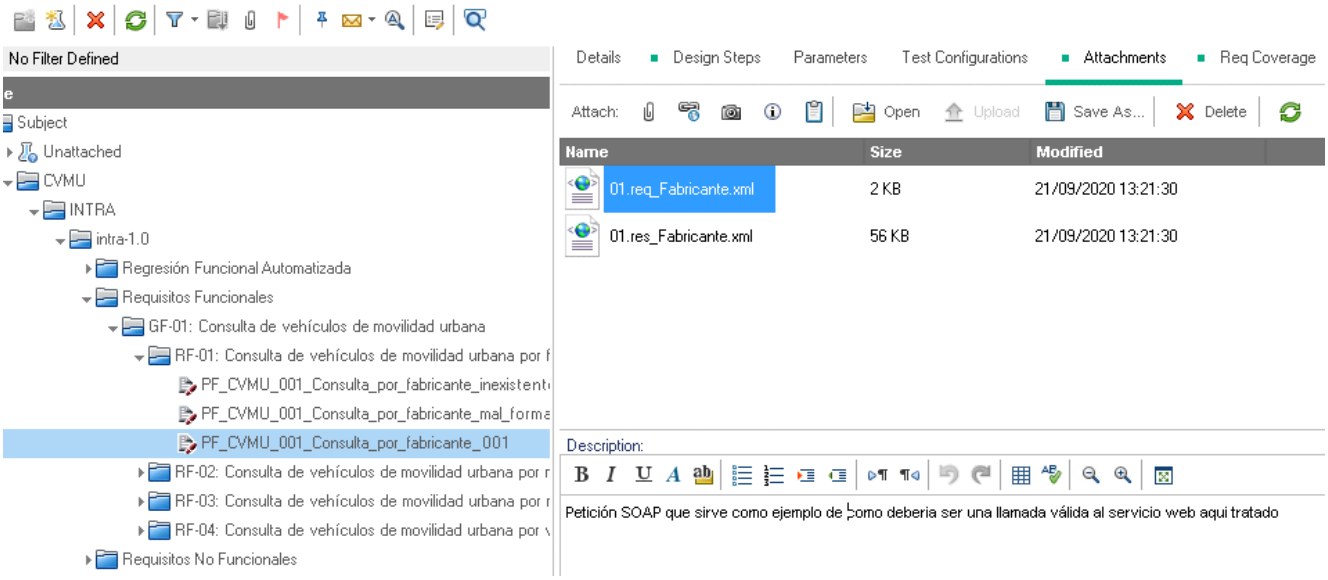


2.3.3.3.2 Pruebas funcionales

Cuando sean necesarios otros ficheros para las **pruebas funcionales** y sean comunes a todos los casos, se colocarán en "Attachments" en la carpeta "**Requisitos funcionales**" y se indicará para ellos su nombre y descripción:



En el caso de que el fichero sea aplicable solo a un caso de prueba, entonces se anexará a nivel de caso de prueba en la pestaña “Attachments” y se incluirá una descripción del contenido del fichero:



2.3.4 Casos de prueba

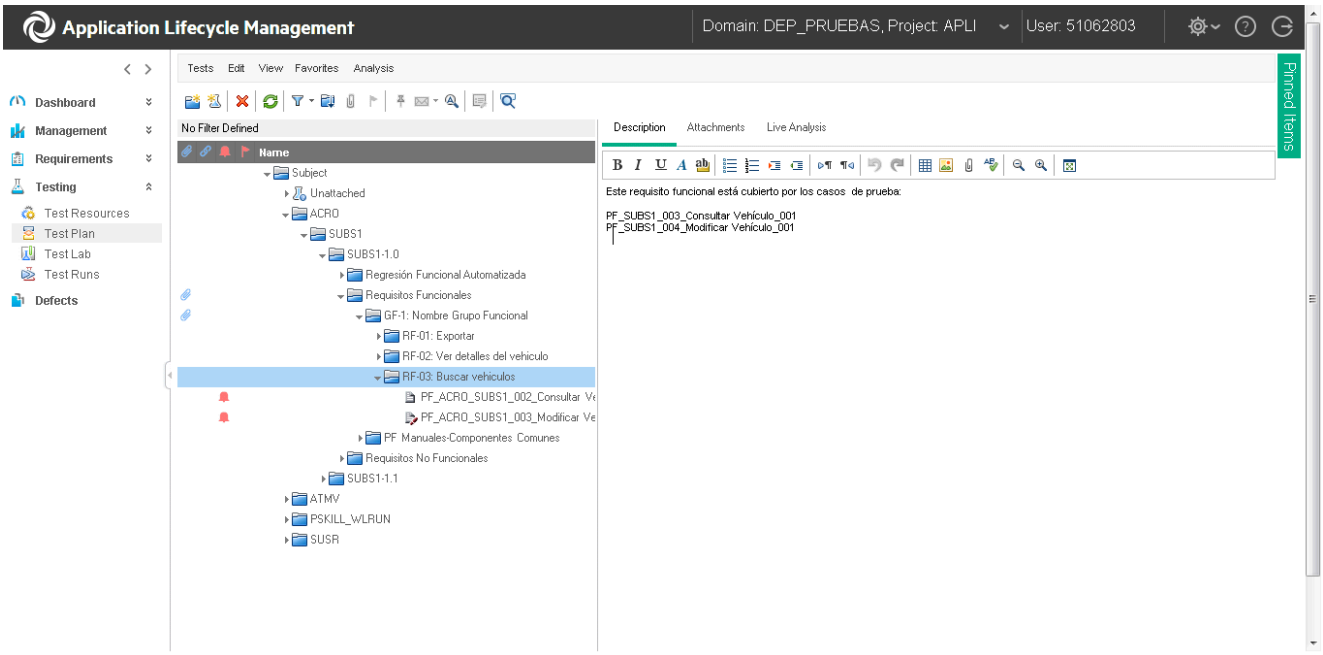
2.3.4.1 Normativa de ubicación de casos de prueba

Al empezar a crear los casos de prueba que cubren los diferentes requisitos, se debe hacer especial hincapié en su ubicación.

Cada requisito debe estar cubierto al menos por un caso de prueba. Y dentro de cada carpeta de requisito funcional o de rendimiento deben existir los casos de prueba que cubren dicho requisito.

Si un caso de prueba cubre varios requisitos funcionales, el caso de prueba deberá colgar de uno de los requisitos funcionales que cubre y en la pestaña descripción del otro requisito funcional indicar por qué casos de prueba están cubiertos.

Ejemplo:



2.3.4.2 Llamada a casos de prueba (Call to test)

En algunos situaciones debido a la dependencia funcional que existe entre casos de prueba se puede dar la situación que un caso de prueba realice la llamada a otro caso de prueba. En estos casos

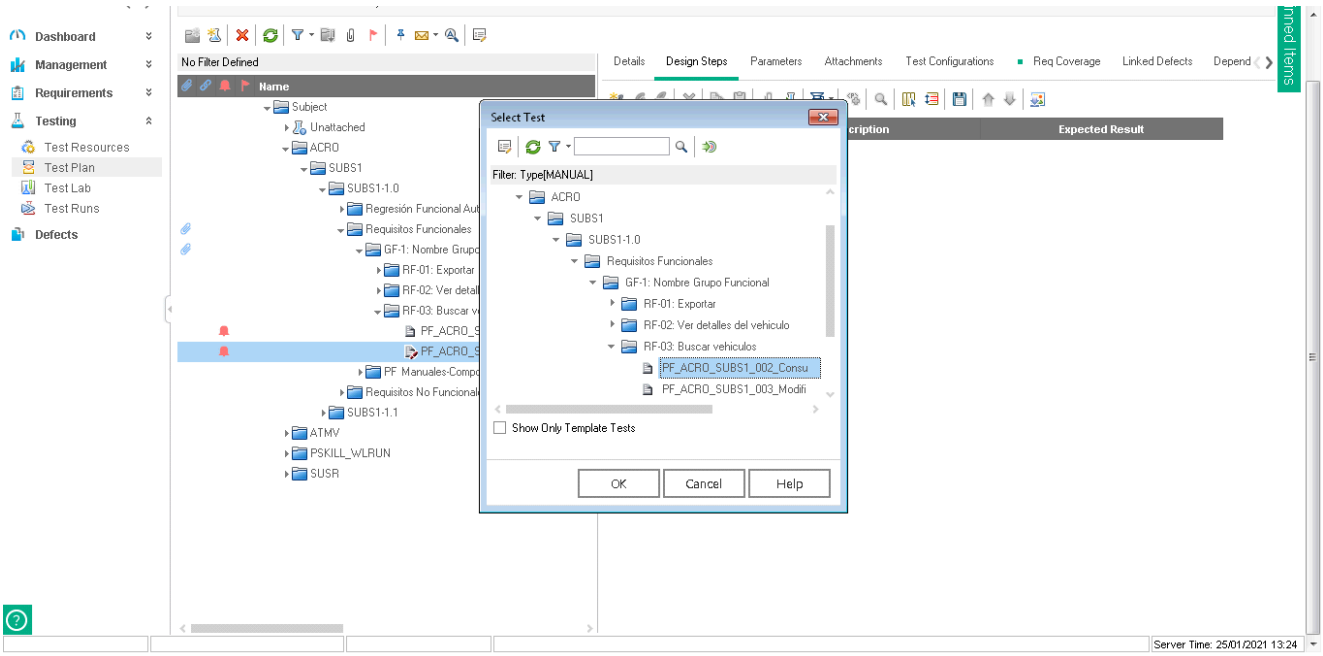
existe siempre una dependencia funcional y no puede existir una dependencia de datos entre ellos.

También puede darse el caso de que existan fragmentos de navegación manual que son comunes a múltiples tests y que por eficiencia sea conveniente definirlos una única vez y hacer llamadas desde todos los tests que lo requieran. A este tipo los llamaremos "Componentes Comunes Manuales" y se explican más adelante en el apartado "2.3.4.3.4 Componentes comunes de casos de prueba funcionales manuales".

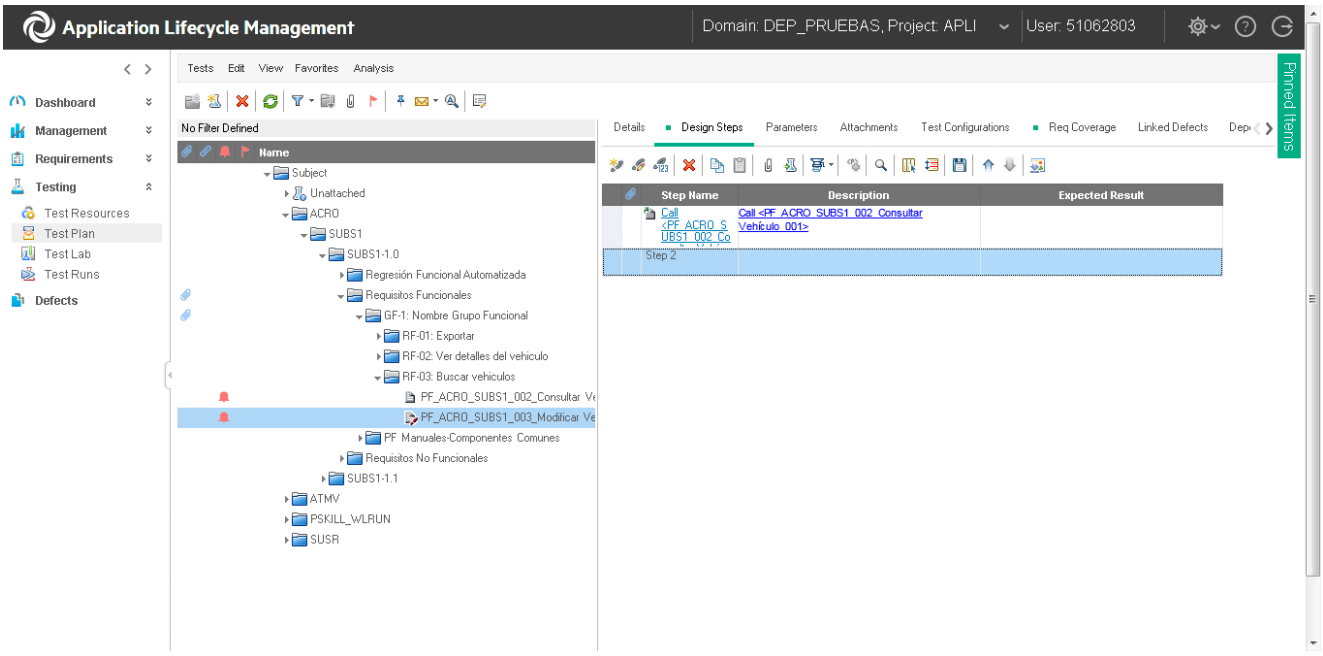
El procedimiento a seguir en ambas situaciones es la de realizar una llamada a un caso de prueba dentro de la pestaña "Design Steps" del otro caso de prueba "*Call to test*".

Los pasos a realizar son:

1. Seleccionar el Step y pulsar en el botón "**Call to Test (Ctrl.+L)"**
2. En la pestaña "**Select a Test**", navegar por la estructura y seleccionar el caso de prueba al que queremos llamar.



En la pestaña “**Design Steps**” aparecerá la llamada al caso de prueba y nos quedará como en la siguiente imagen:



La pestaña de datos de entrada que utilice el caso de prueba que hace la llamada (el que tiene el Call to test) contendrá **todos los datos necesarios** para ejecutar ese caso de prueba, incluidos los que se usen en el componente común o el test invocado. Para más información sobre el fichero de datos de prueba ver el apartado “2.3.3.1 Fichero XLS de pruebas funcionales”.

En la “Description” del caso de prueba que hace la llamada se informarán el nombre y descripción de **todos** los campos de esta pestaña. Para más información sobre el campo “Description” de un caso de prueba, ver el apartado “2.3.4.4.2 Detalles de un caso de prueba funcional”.

2.3.4.3 Tipos de casos de prueba

2.3.4.3.1 Casos de prueba de rendimiento

Los casos de prueba de rendimiento no se crean desde ALM, se generan cuando se crea un nuevo script a través de la herramienta LoadRunner-Vugen. De manera que cada caso de prueba de rendimiento de ALM quedará asociado a un script LoadRunner-Vugen.

2.3.4.3.1.1 Reglas de Nomenclatura

2.3.4.3.1.1.1 Sintaxis

<Aplicación_Subsistema>_<999>_<Descripción>

Donde:

Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
Subsistema	**Es el acrónimo del subsistema**
999	**Es un índice secuencial dentro de la misma Aplicación. Comienza por 001 y se reinicia por aplicación y por cada subsistema. Ver ejemplos más abajo.**
Descripción	**Es la descripción de la Transacción de negocio**

2.3.4.3.1.2 Ejemplo

Por aplicación:

ACRO_001_Consulta_Vehiculos

ACRO_002_Consultas_DNI

ACRO_003_Consultas_Sanciones

Por subsistema:

SUB1: Subsistema 1 dentro de ACRO

ACRO_SUBS1_001_Consulta_Vehiculos

ACRO_SUBS1_002_Consulta_DOI

SUB2: Subsistema 2 dentro de ACRO

ACRO_SUBS2_001_Consulta_Sanciones

ACRO_SUBS2_002_Consulta_Pago_Sancion

2.3.4.3.1.3 Pasos a realizar

Ver apartado de creación de scripts Vugen en ALM del anexo “Anexo 03.08 Guia de programacion en LoadRunner-Vugen 12.53”.

2.3.4.3.2 Casos de prueba funcionales (Manuales)

2.3.4.3.2.1 Reglas de Nomenclatura

2.3.4.3.2.1.1 Sintaxis

PF_<Aplicación>_<Subsistema>_<999>_<Descripción>_<000>

Donde:

Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
Subsistema	**Es el acrónimo que asigna la DGT a los subsistemas de la**
999	**En un índice que se corresponde con una agrupación funcional dentro de cada requisito.** **Comienza por 001 y es único por aplicación-subsistema. Ver ejemplos.**
Descripción	**Es la descripción de la agrupación funcional.**
000	**Es el nº del caso de prueba dentro de cada agrupación funcional que se corresponde con el índice (999).** **Comienza por 001 y es único por cada agrupación funcional.**

2.3.4.3.2.2 Ejemplo

Por aplicación:

GF-01 Consultas de Vehículo:

PF_ACRO_001_Consulta_Matricula_001

PF_ACRO_001_Consulta_Matricula_002

PF_ACRO_001_Consulta_MatriculaError_003

PF_ACRO_001_Consulta_MatriculaError_004

PF_ACRO_002_Consulta_DOI_001

PF_ACRO_002_Consulta_DOI_002

PF_ACRO_002_Consulta_DOIError_003

Por subsistema:

SUBS1: Subsistema 1 dentro de ACRO

PF_ACRO_SUBS1_001_Consulta_Vehiculos_001

PF_ACRO_SUBS1_001_Consulta_Vehiculos_002

PF_ACRO_SUBS1_002_Consulta_DOI_001

PF_ACRO_SUBS1_002_Consulta_DOI_002

SUBS2: Subsistema 2 dentro de ACRO

PF_ACRO_SUBS2_001_Consulta_Sanciones _001

PF_ACRO_SUBS2_001_Consulta_Sanciones_002

PF_ACRO_SUBS2_002_Consulta_Pago_Sancion_001

PF_ACRO_SUBS2_002_ Consulta_Pago_Sancion _002

2.3.4.3.2.3 Pasos a realizar

1. Ubicarse en la zona del "**TestPlan"** y seleccionar la carpeta–ruta donde vamos a introducir nuestro

2. Pulsar el botón de "**New Test**" .

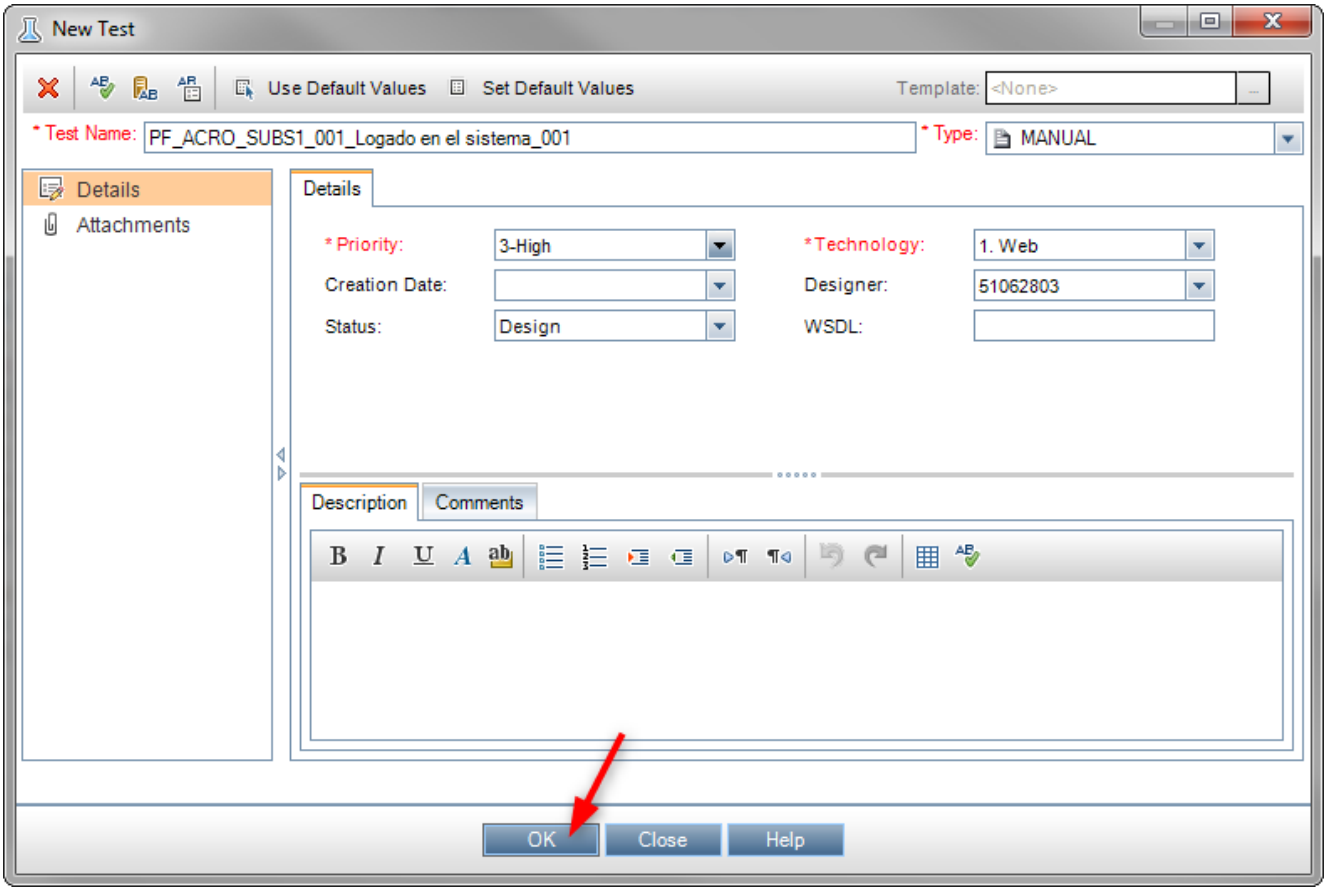
En la pantalla que aparece introducir los siguientes datos:

Test Type: "**Manual**"

Test Name: Nombre del caso según la nomenclatura dada

Priority: Prioridad del caso de prueba

Technology: Tipo de tecnología sobre la que se realizará la prueba (Web, WebService, Batch, Otros…)



NOTA: Se repite el proceso por cada prueba "***Manual***" a introducir.

2.3.4.3.3 Casos de prueba funcionales (Automáticas)

(Nota: la Oficina de Pruebas es la encargada de la automatización de casos de pruebas funcionales, se indica el procedimiento solo a título informativo)

2.3.4.3.3.1 Reglas de Nomenclatura

2.3.4.3.3.1.1 Sintaxis

PFA_<Aplicación>_<Subsistema>_<999>_<Descripción>_<000>

Donde:

Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
Subsistema	**Es el acrónimo que asigna la DGT a los subsistemas de la**
999	**En un índice que se corresponde con una agrupación funcional dentro de cada requisito. Comienza por 001 y es único por aplicación-subsistema. Ver ejemplos.**
Descripción	**Es la descripción de la agrupación funcional.**
000	**Es el nº del caso de prueba dentro de cada agrupación funcional que se corresponde con el índice (999). Comienza por 001 y es único por cada agrupación funcional.**

2.3.4.3.3.2 Ejemplo

Por aplicación:

GF-01 Consultas de Vehículo:

PFA_ACRO_001_Consulta_Matricula_001

PFA_ACRO_001_Consulta_Matricula_002

PFA_ACRO_001_Consulta_MatriculaError_003

PFA_ACRO_001_Consulta_MatriculaError_004

PFA_ACRO_002_Consulta_DOI_001

PFA_ACRO_002_Consulta_DOI_002

PFA_ACRO_002_Consulta_DOIError_003

Por subsistema:

SUBS1: Subsistema 1 dentro de ACRO

PFA_ACRO_SUBS1_001_Consulta_Vehiculos_001

PFA_ACRO_SUBS1_001_Consulta_Vehiculos_002

PFA_ACRO_SUBS1_002_Consulta_DOI_001

PFA_ACRO_SUBS1_002_Consulta_DOI_002

SUBS2: Subsistema 2 dentro de ACRO

PFA_ACRO_SUBS2_001_Consulta_Sanciones _001

PFA_ACRO_SUBS2_001_Consulta_Sanciones_002

PFA_ACRO_SUBS2_002_Consulta_Pago_Sancion_001

PFA_ACRO_SUBS2_002_ Consulta_Pago_Sancion _002

2.3.4.3.3.3 Pasos a realizar

1. Ubicarse en la zona del "**TestPlan"** y seleccionar la carpeta–ruta donde vamos a introducir nuestro
2. Pulsar el botón de "**New Test**" .

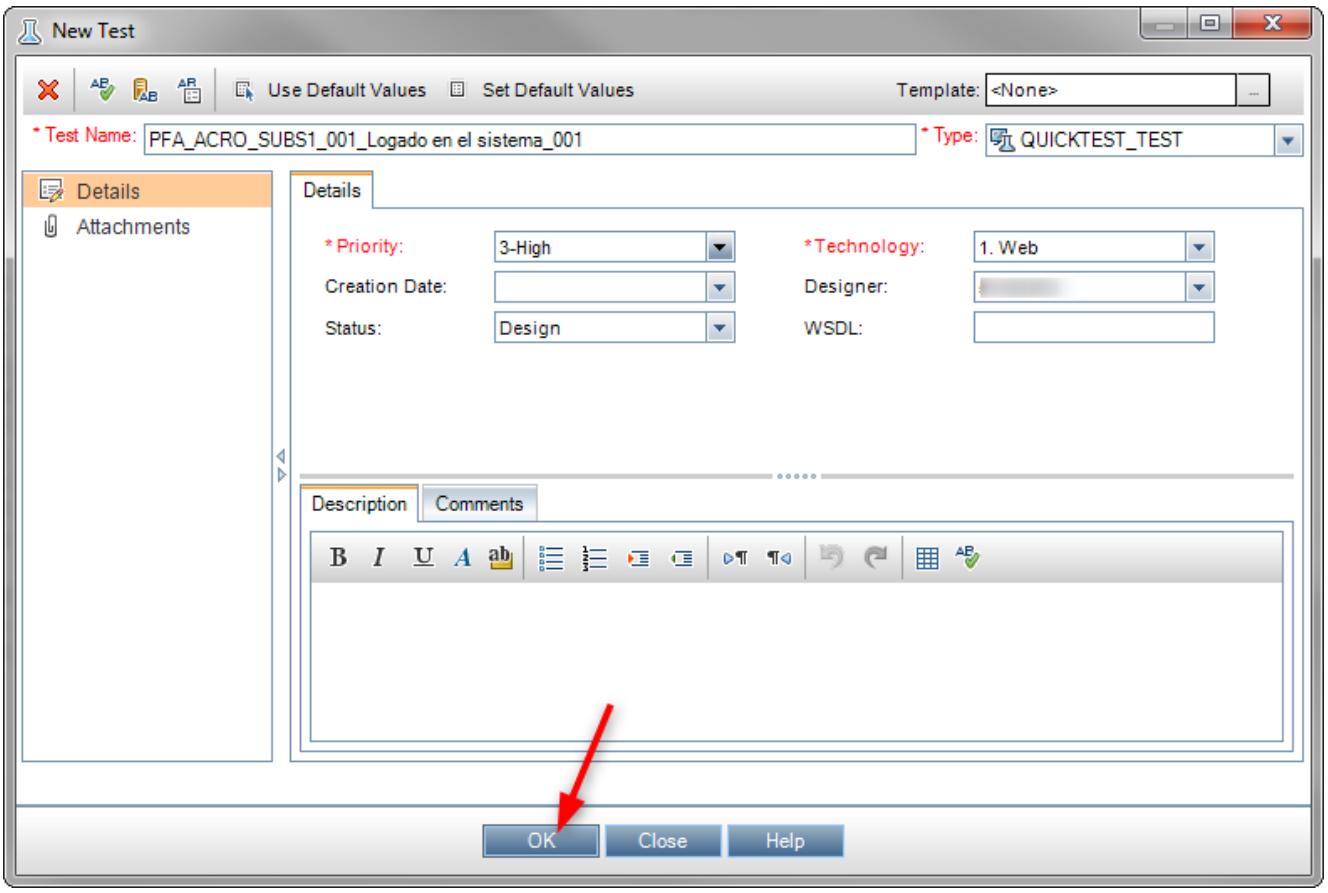
   En la pantalla que aparece introducir los siguientes datos:

   Test Type: "**QUICKTEST_TEST**"

   Test Name: Nombre del script según la nomenclatura dada

   Priority: Prioridad del caso de prueba

   Technology: Tipo de tecnología sobre la que se realizará la prueba



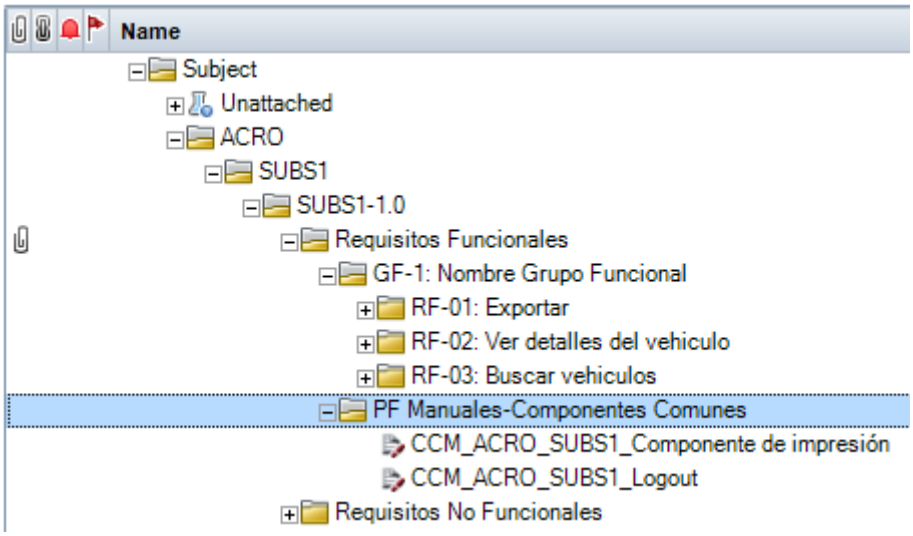
NOTA: Se repite el proceso por cada prueba "***Automática***" a introducir.

2.3.4.3.4 Componentes comunes de casos de prueba funcionales manuales

2.3.4.3.4.1 Reglas de Nomenclatura

2.3.4.3.4.1.1 Sintaxis

Este tipo de tests se ubicarán en una carpeta llamada "PF Manuales-Componentes Comunes" (PF=Pruebas Funcionales), que estará dentro de la carpeta "Requisitos Funcionales" de la aplicación/Subsistema y fuera de cualquier grupo funcional, porque son fragmentos que pueden ser reutilizados en cualquier test de cualquier Requisito Funcional, por ello estos componentes no llevarán asociados ningún requisito.



La sintaxis de este tipo de componentes es:

CCM_<Aplicación>_<Subsistema>_<Descripción>

Donde:

Aplicación **Es el acrónimo que asigna la DGT a cada Aplicación**

Subsistema **Es el acrónimo que asigna la DGT a los subsistemas de la**

Descripción **Es la descripción del componente común.**

2.3.4.3.4.2 Ejemplo

Por aplicación:

PF Manuales-Componentes Comunes:

CCM_ACRO_Componente de impresión

CCM_ACRO_Logout

CCM_ACRO_Descarga PDF

Por subsistema:

SUBS1: Subsistema 1 dentro de ACRO

CCM_ACRO_SUBS1_Componente de impresión

CCM_ACRO_SUBS1_Logout

CCM_ACRO_SUBS1_Descarga PDF

SUBS2: Subsistema 2 dentro de ACRO

CCM_ACRO_SUBS2_Login

CCM_ACRO_SUBS2_Logout

2.3.4.3.4.3 Pasos a realizar

1. Ubicarse en la zona del "**TestPlan"** y seleccionar la carpeta–ruta donde vamos a introducir nuestro
2. Pulsar el botón de "**New Test**" .

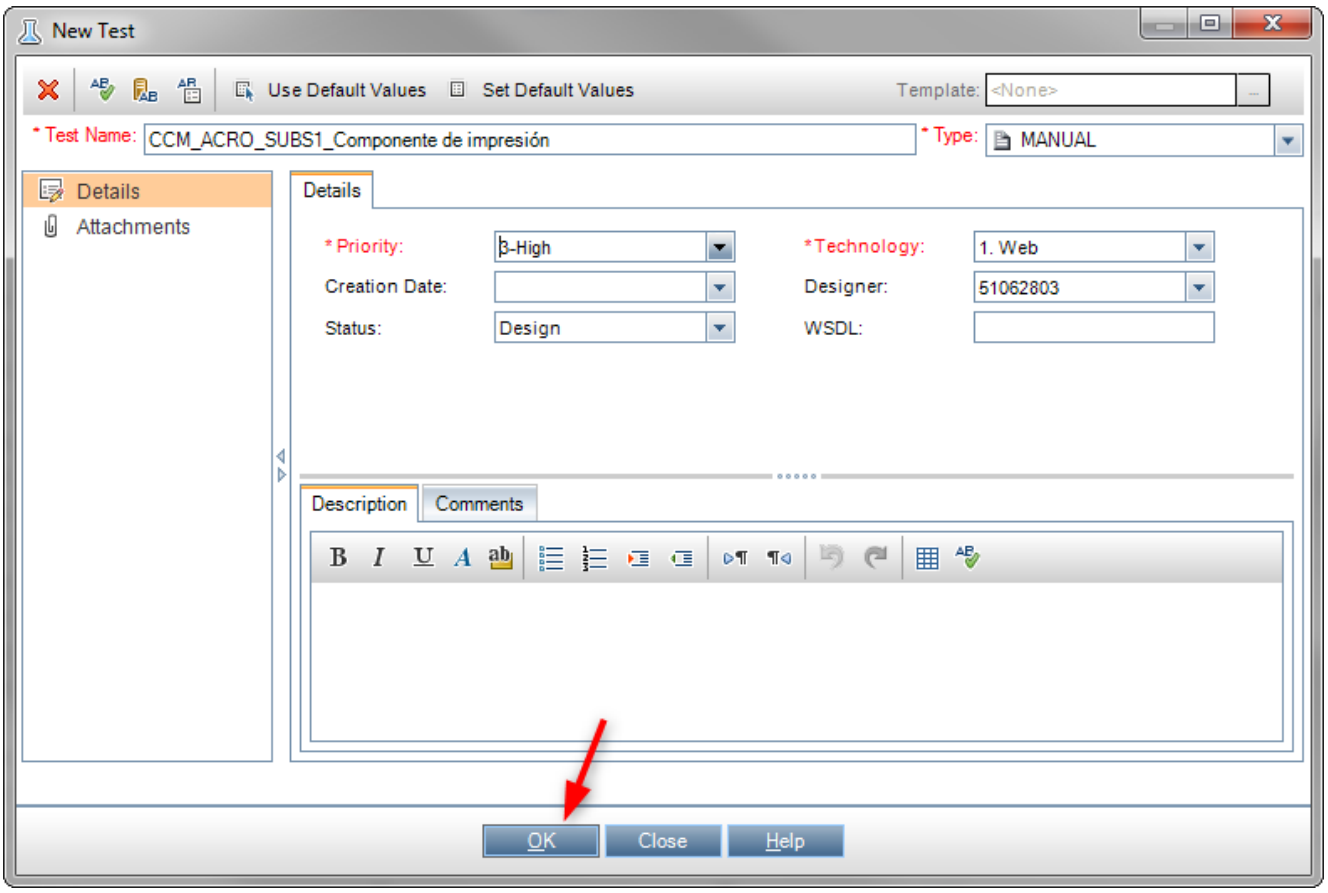
   En la pantalla que aparece introducir los siguientes datos:

   Test Type: "**Manual**"

   Test Name: Nombre del componente según la nomenclatura dada

   Priority: Prioridad del componente común

   Technology: Tipo de tecnología sobre la que se realizará la prueba



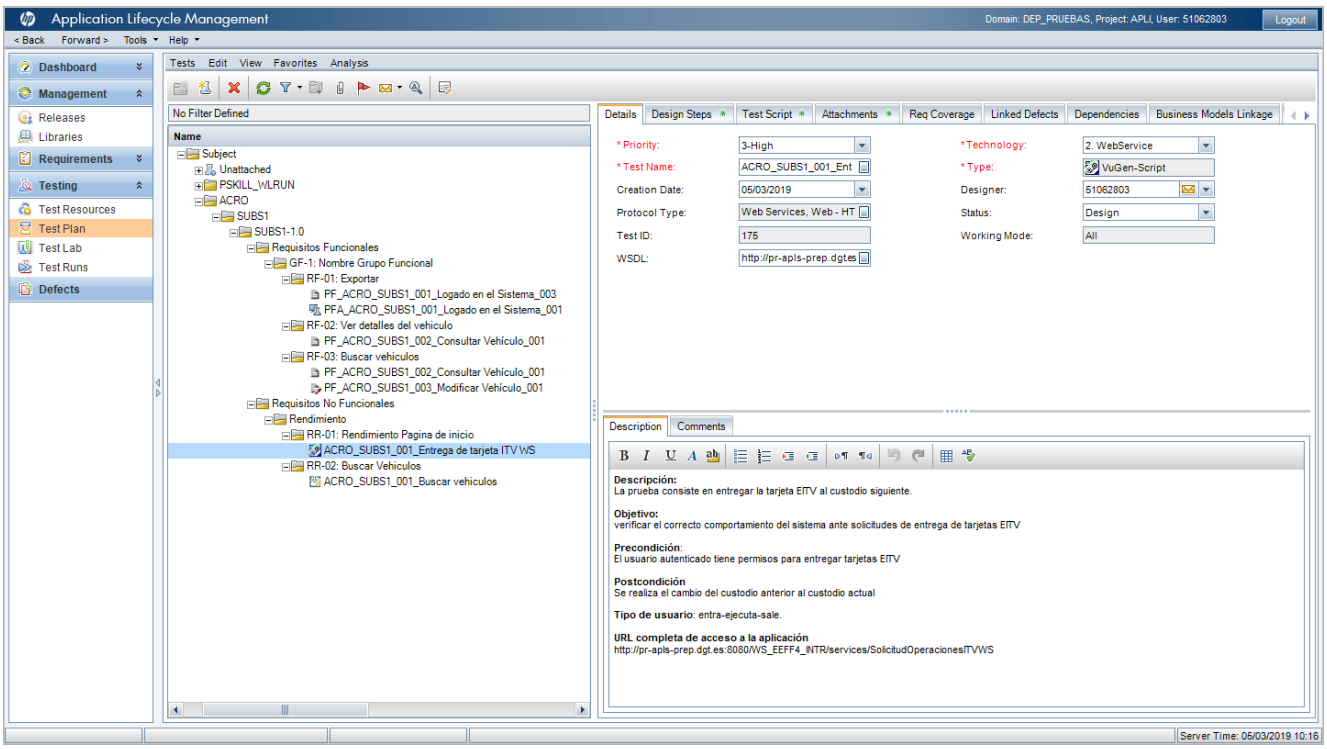
NOTA: Se repite el proceso por cada prueba "***Componente Común Manual***" a introducir.

2.3.4.4 Documentación de casos de prueba

Por cada caso de prueba creado en ALM se deberá incorporar su documentación, para ello habrá que rellenar las pestañas "**Details**" y "**Design Steps**".

2.3.4.4.1 Detalles de un caso de prueba de rendimiento

En la pestaña "**Details"** de un caso de prueba de rendimiento se deben rellenar los campos "**Priority**", "**Technology**", "**WSDL"** si es de tipo **WebService** y la pestaña "**Description**".

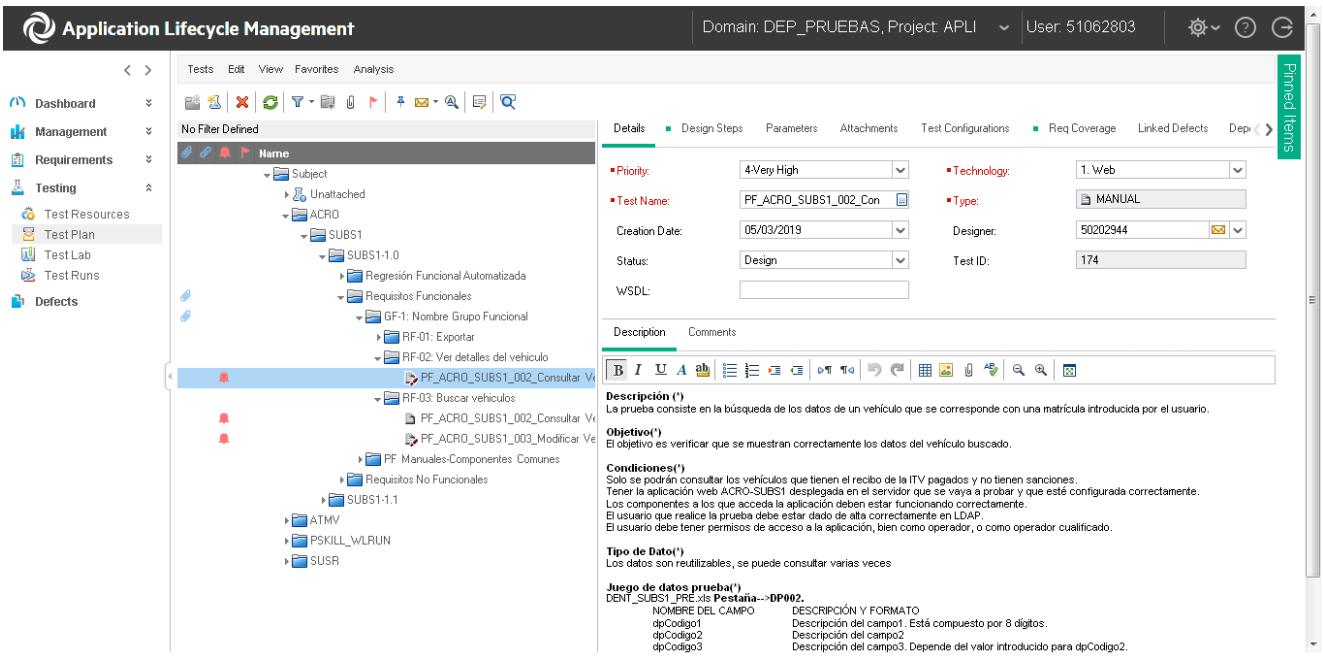


En el campo "**Description**" deberá aparecer toda la información referente al caso de prueba, como por ejemplo:

- Descripción
- Objetivo
- Condiciones necesarias para la ejecución
- URL completa de acceso a la aplicación
- Tipo de usuario, si es persistente en sesión o no, es decir, indicar si el usuario accede en vuser_init o en Action.
- Juego de datos de entrada a utilizar

2.3.4.4.2 Detalles de un caso de prueba funcional

En la pestaña "**Details"** de un caso de prueba funcional tanto manual como automático se deben rellenar los campos "**Priority**", "**Technology**" y la pestaña "**Description**".



En la pestaña "**Description**" deberá aparecer toda la información referente al caso de prueba, como por ejemplo:

- Descripción
- Objetivo
- Condiciones necesarias para la ejecución
- Descripción de las pestañas que contienen los datos utilizados en el caso de prueba: nombre, descripción y formato de los campos, si hay dependencias entre campos, y si los datos son reutilizables o se consumen al ejecutar el caso de prueba
- Otros ficheros de entrada a utilizar en la prueba además del xls

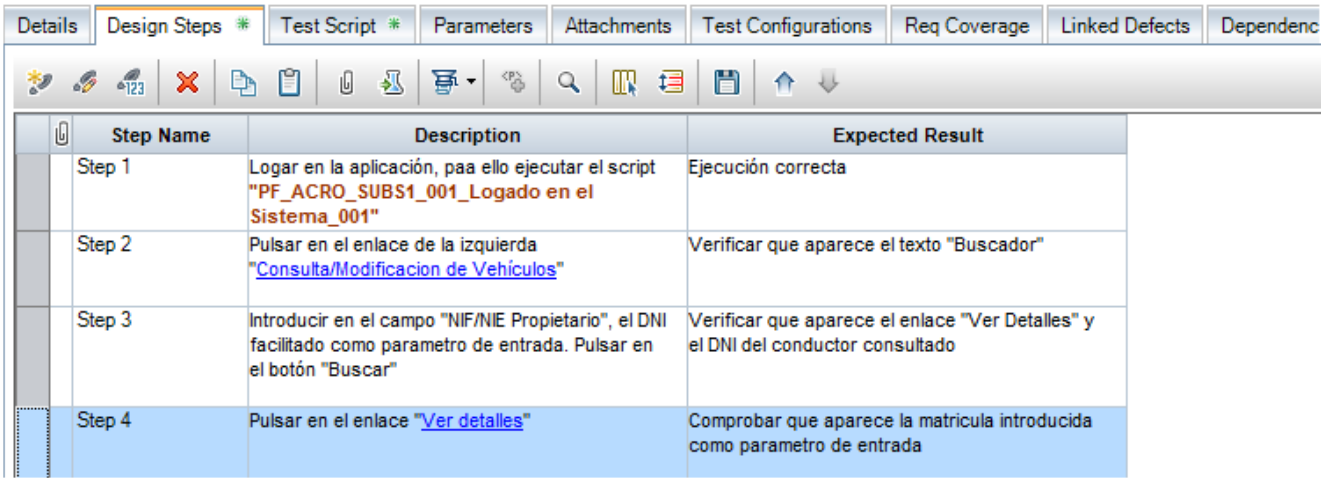
2.3.4.4.3 Diseño de pasos

Pulsar en la pestaña "**Design Steps**" e introducir toda la secuencia de acciones y resultados esperado necesarios para la ejecución del caso de prueba.

Las acciones a realizar para introducir los pasos son:

Pulsar el botón y rellenar la información relativa a:

- Step Name: nombre del paso.
- Description: descripción detallada de las acciones a realizar sobre el sistema.
- Punto de chequeo (TRAN) (solo para casos de prueba de rendimiento).
- Llamada a sistema externo (solo para casos de prueba de rendimiento): Detalle del sistema externo al cual se llama, por ejemplo, aplicación y proceso (webservices, ejb, etc)
- Expected: resultado esperado tras la ejecución del paso, especificando la cadena de texto, imagen,… a chequear, que verifique que el proceso de negocio se ha ejecutado correctamente.



Details | Design Steps | Test Script | Parameters | Attachments | Test Configurations | Req Coverage | Linked Defects | Dependenc

Step Name	Description	Expected Result
Step 1	Logar en la aplicación, paa ello ejecutar el script "PF_ACRO_SUBS1_001_Logado en el Sistema_001"	Ejecución correcta
Step 2	Pulsar en el enlace de la izquierda "Consulta/Modificacion de Vehículos"	Verificar que aparece el texto "Buscador"
Step 3	Introducir en el campo "NIF/NIE Propietario", el DNI facilitado como parametro de entrada. Pulsar en el botón "Buscar"	Verificar que aparece el enlace "Ver Detalles" y el DNI del conductor consultado
Step 4	Pulsar en el enlace "Ver detalles"	Comprobar que aparece la matricula introducida como parametro de entrada

En los requisitos de rendimiento donde los tiempos medios y máximos de respuesta excedan de los definidos en la guía de desarrollo es necesario indicar que paso del caso de prueba está asociado a dicho requisito. El resto de pasos se medirán teniendo en cuenta los tiempos medios y máximos de respuesta estándar definidos en la guía.

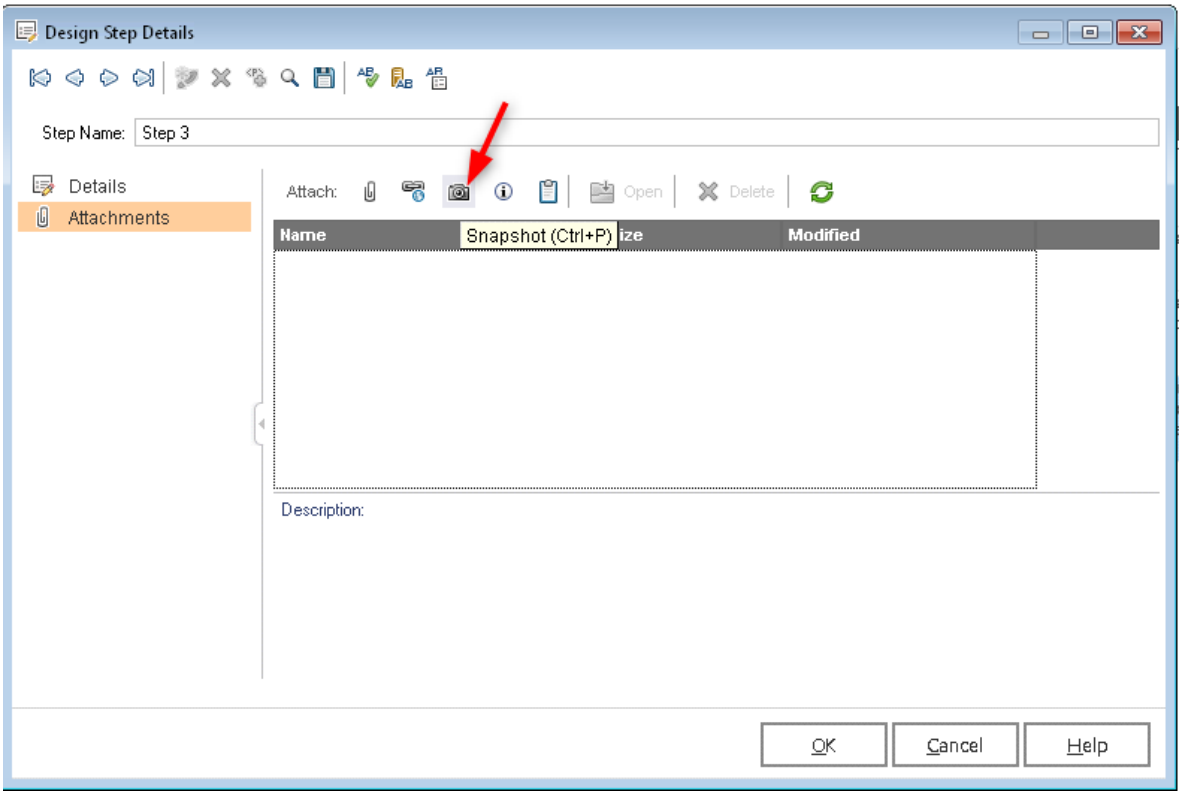
Ejemplo:

“Punto de Chequeo "TRAN_CCAL_002_Pagina_Consulta_Conductores" --> --> Req Relacionado RR-02:Consulta Conductor”

	Step Name	Description	Expected Result
	Step 1	- Tener la aplicacin web de CCAL desplegada en el servidor que se vaya a	
	Step 2	Abrir el navegador e introducir la url de acceso a la aplicación **lr_think_time (8);**	Chequear que aparece el contenido "**Introduzca sus datos de acceso**"
	Step 3	Introducir en el campo "Usuario" y "Password" los datos de acceso a la aplicación. Dichos datos están en el fichero adjunto DENT_CCAL_PRE_000.txt **lr_think_time (15);** **Punto de Chequeo "TRAII_CCAL_000_Logado"**	Chequear que aparece el contenido "**Está en: Menú de Aplicaciones > CURSO CALIDAD** " y aparece el texto "**Bienvenido**"
	Step 4	El usuario selecciona el enlace "**Consulta Conductores**". **lr_think_time (7);** **Punto de Chequeo "TRAII_CCAL_002_Pagina_Consulta_Conductores"** **--> --> Req Relacionado RR-02:Consulta Conductor**	El sistema muestra la pantalla para la selección del criterio de búsqueda de vehículos. Chequear que aparece el contenido "**NIF/NIE/CIF conductor** "

2.3.4.4.3.1 Anexar capturas a los pasos

Para anexar capturas a los pasos pulsaremos sobre el botón y se anexarán las capturas de pantallas necesarias para poder ejecutar la prueba.



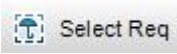
2.3.4.4.4 Attachments

En la pestaña “Attachments” de cada caso de prueba se anexarán los archivos específicos necesarios para la ejecución del caso de prueba. Por ejemplo:

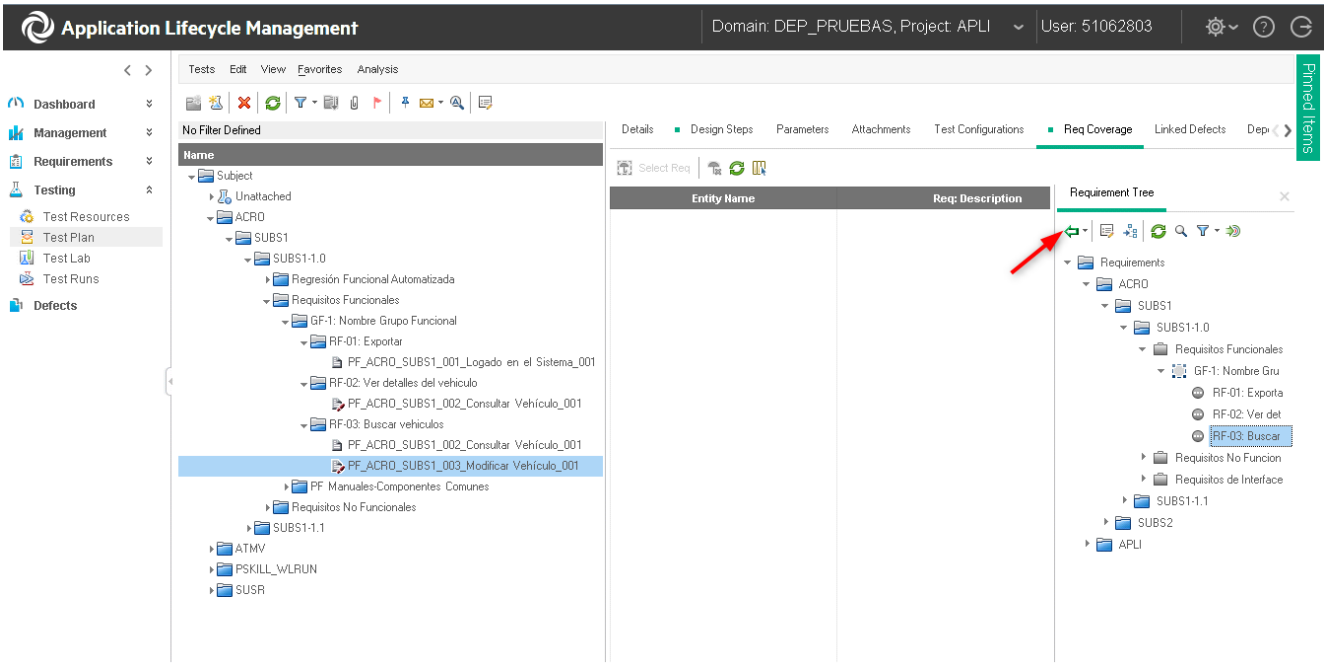
- Caso de prueba funcional: el XML (o json) de petición y de respuesta en el caso de un WebService
- Caso de prueba de rendimiento: el archivo de datos específico para el script, el XML (o json) de petición y de respuesta en el caso de un WebService

2.3.4.4.5 Req. Coverage

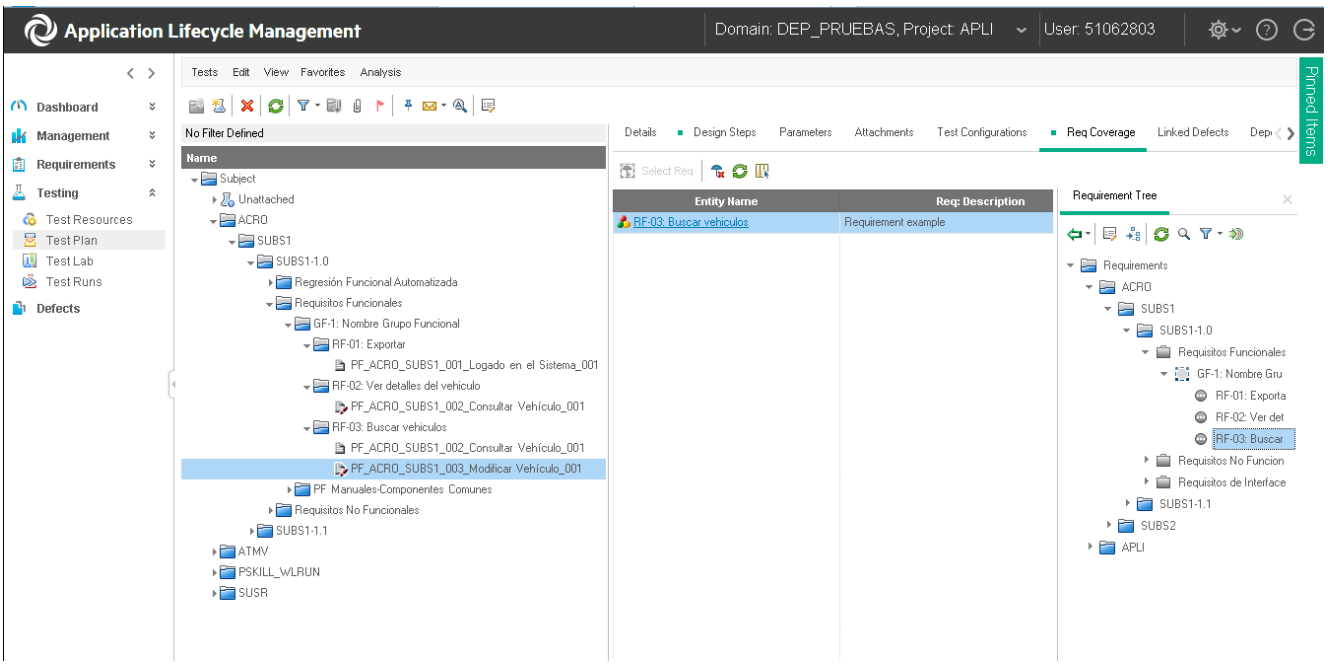
En la pestaña “**Req. Coverage**” se asociará el caso de prueba con el o los requisitos que esté relacionado.

Para ello habrá que seleccionar el caso de prueba y en la pestaña “Req. Coverage” , pulsaremos en “Select Req” 

y seleccionamos el requisito al que se quiere asociar.



Tras haberlo enlazado:



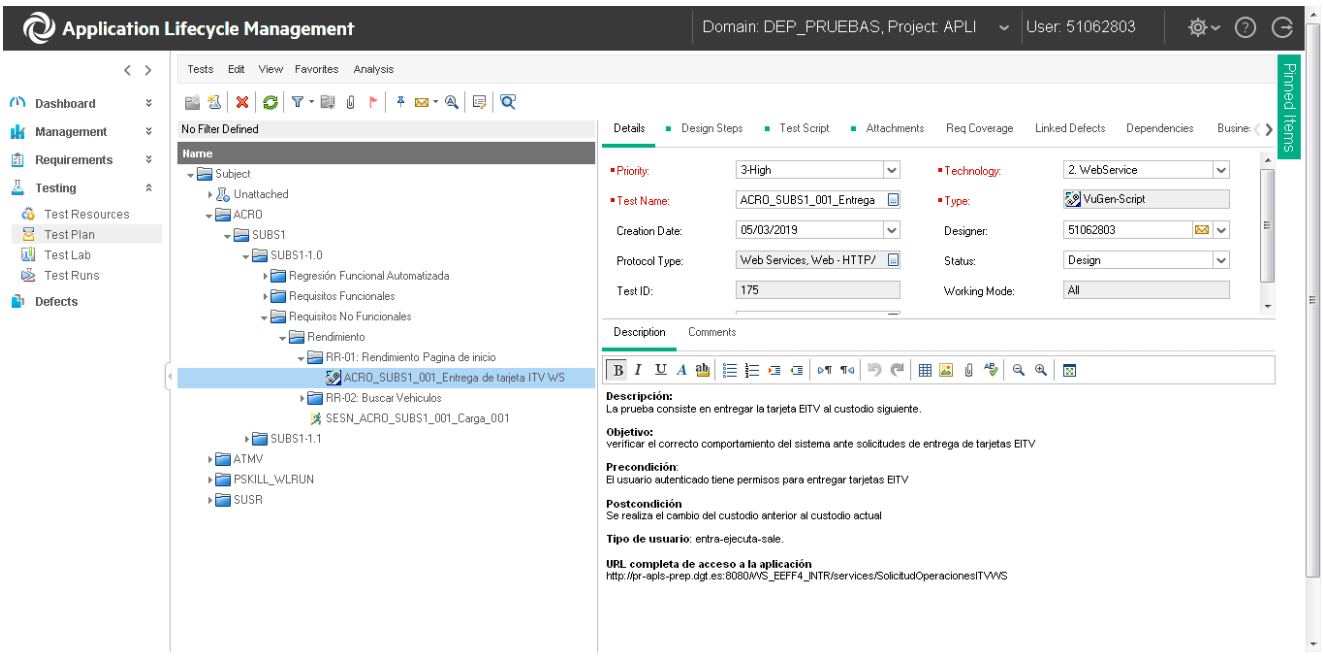
NOTA: Repetir el proceso por cada requisito a enlazar con el caso de prueba.

NOTA: los elementos de la carpeta de Componentes Comunes no llevan requerimientos asociados.

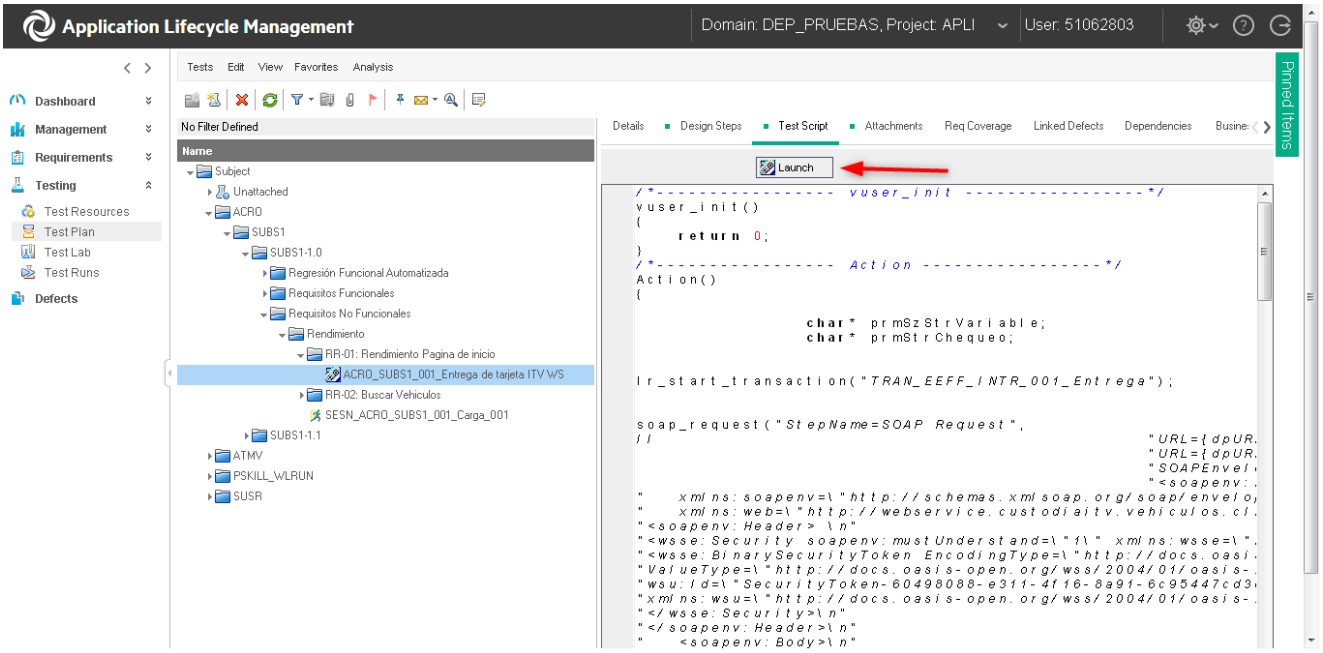
2.3.4.5 Abrir Scripts Vugen desde ALM

Para grabar un script de rendimiento ya existente se deberá abrir desde ALM. Los pasos a realizar son:

1. Seleccionar el script de rendimiento a abrir.



2. Pulsar en el botón “**Launch**” de la pestaña “Test Script”



A continuación se abrirá la herramienta Vugen de LoadRunner.

ACRO_SUBS1_001_Entrega de tarjeta ITV WS - HP Virtual User Generator - Web Services, Web - HTTP/HTML

File Edit View Search Design Record Replay ALM SOATools Tools Window Help

Solution Explorer

ution Untitled

ACRO_SUBS1_001_Entrega de tarjeta ITV WS

- Actions
  - vuser_init
  - Action
  - vuser_end
- Extra Files
  - globals.h
- Run-Time Settings
- Parameters
- Replay Runs [Passed]

ACRO_S...eta ITV WS : Action.c

```
Action()
{

                char* prmSzStrVariable;
                char* prmStrChequeo;


lr_start_transaction("TRAN_EEFF_INTR_001_Entrega");


soap_request("StepName=SOAP Request",
//                                          "URL={dpURL}/WS_EEFF4_INTR/services/SolicitudOperacionesITVWS",
                                            "URL={dpURL}/WS_EEFF4_INTR/services/SolicitudOperacionesITVWS_2",
                                            "SOAPEnvelope="
                                            "<soapenv:Envelope \n"
"   xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" \n"
"   xmlns:web=\"http://webservice.custodiaitv.vehiculos.cliente.trafico.es\">\n"
"<soapenv:Header> \n"
"<wsse:Security soapenv:mustUnderstand=\"1\" xmlns:wsse=\"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd\">\n"
"<wsse:BinarySecurityToken EncodingType=\"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary\" \n"
"ValueType=\"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3\" \n"
"wsu:Id=\"SecurityToken-60498088-e311-4f16-8a91-6c95447cd3e7\" \n"
"xmlns:wsu=\"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd\">{dpCertificado}</wsse:BinarySecurityToken>\n"
"</wsse:Security>\n"
"</soapenv:Header>\n"
"   <soapenv:Body>\n"
"      <web:entregarEITV>\n"
"         <localeId>es</localeId>\n"
"         <peticion><![CDATA[<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"
"<Solicitud_Registro_Entrada version=\"3.0\"><Datos_Firmados><Datos_Genericos><Remitente>"
                                            "<Nombre>Usuario de Prueba EEFF (211)</Nombre><Apellidos></Apellidos>"
                                            "<Documento_Identificacion><Numero>A28191211</Numero></Documento_Identificacion>"
                                            "<Correo_Electronico></Correo_Electronico></Remitente><Interesados></Interesados>"
```

Output

Replay status passed View summary Started at: 22/06/2017 11:16:17 Elapsed time: 00:35

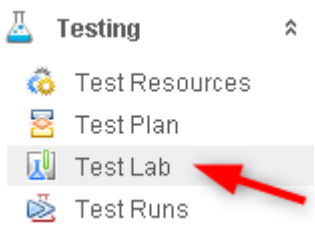
```
Starting action vuser_end.
Ending action vuser_end.
Vuser Terminated.
```

Solution Explorer | Step Navigator | Output | Tasks | Errors | Run-time Data

Ready | Script parsing succeeded | Replay status passed

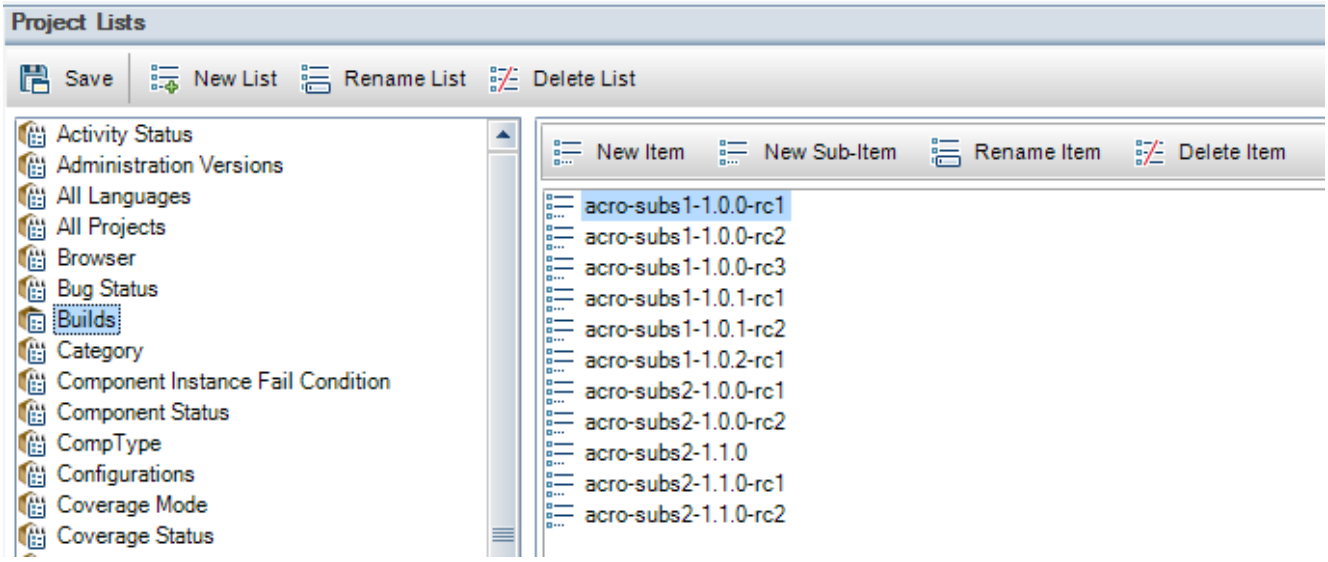
2.4 Módulo Test Lab

2.4.1 Objetivo

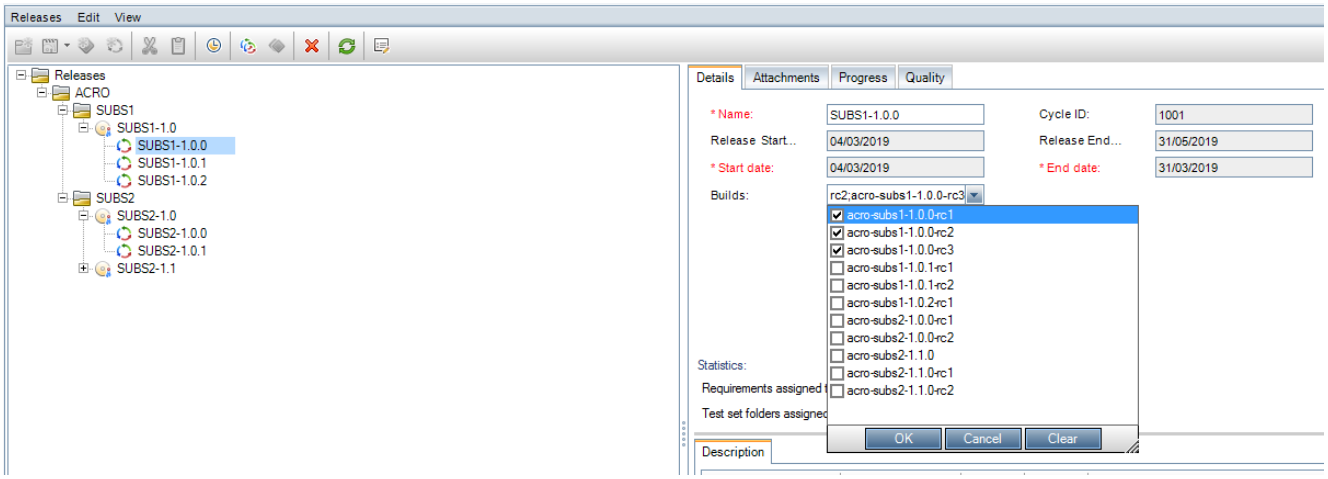


En el módulo Test Lab se crearán las baterías de pruebas y se registrarán las ejecuciones por cada Release, Ciclo y Build.

Para dar de alta las diferentes Builds asociadas a una aplicación sobre las que ejecutar las pruebas se solicitará a la Oficina de Pruebas que se dé de alta dicha Build en ALM dentro del proyecto.



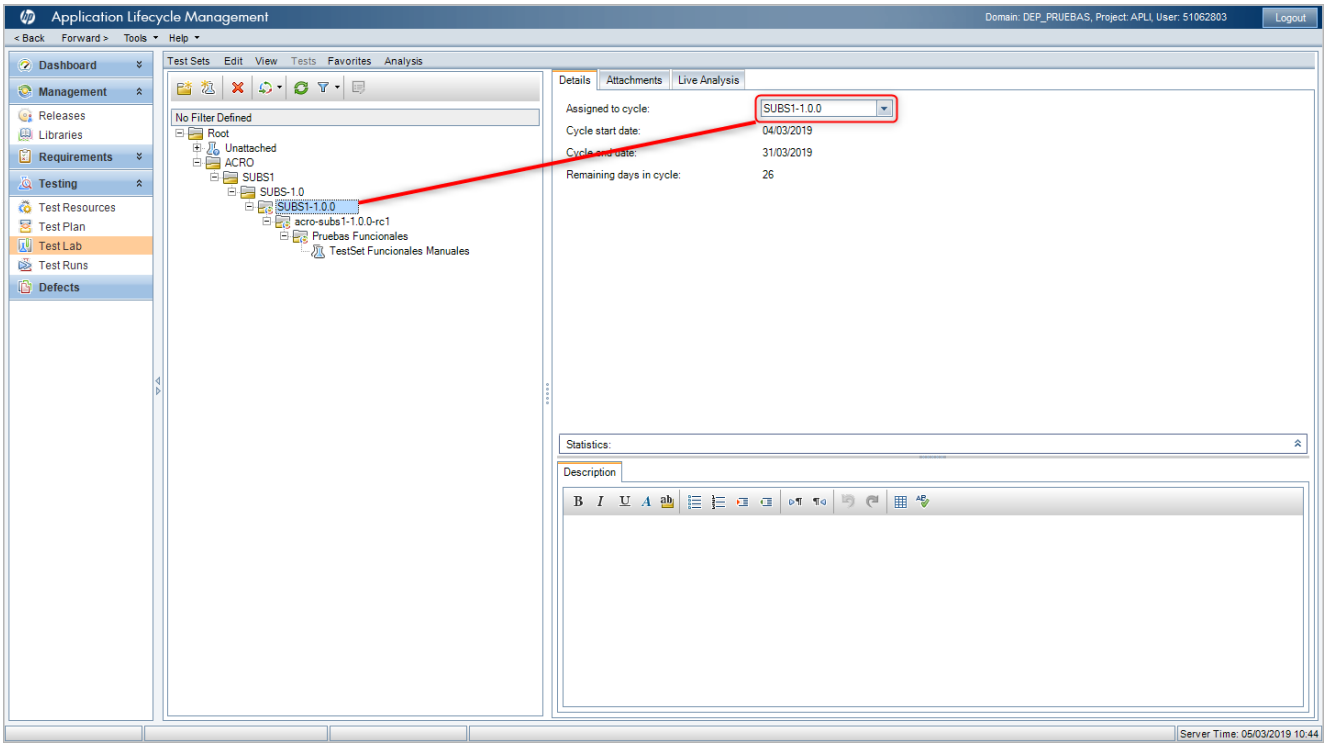
Las Builds se asociarán al ciclo oportuno y se deberá asignar una Build a cada Test Set.



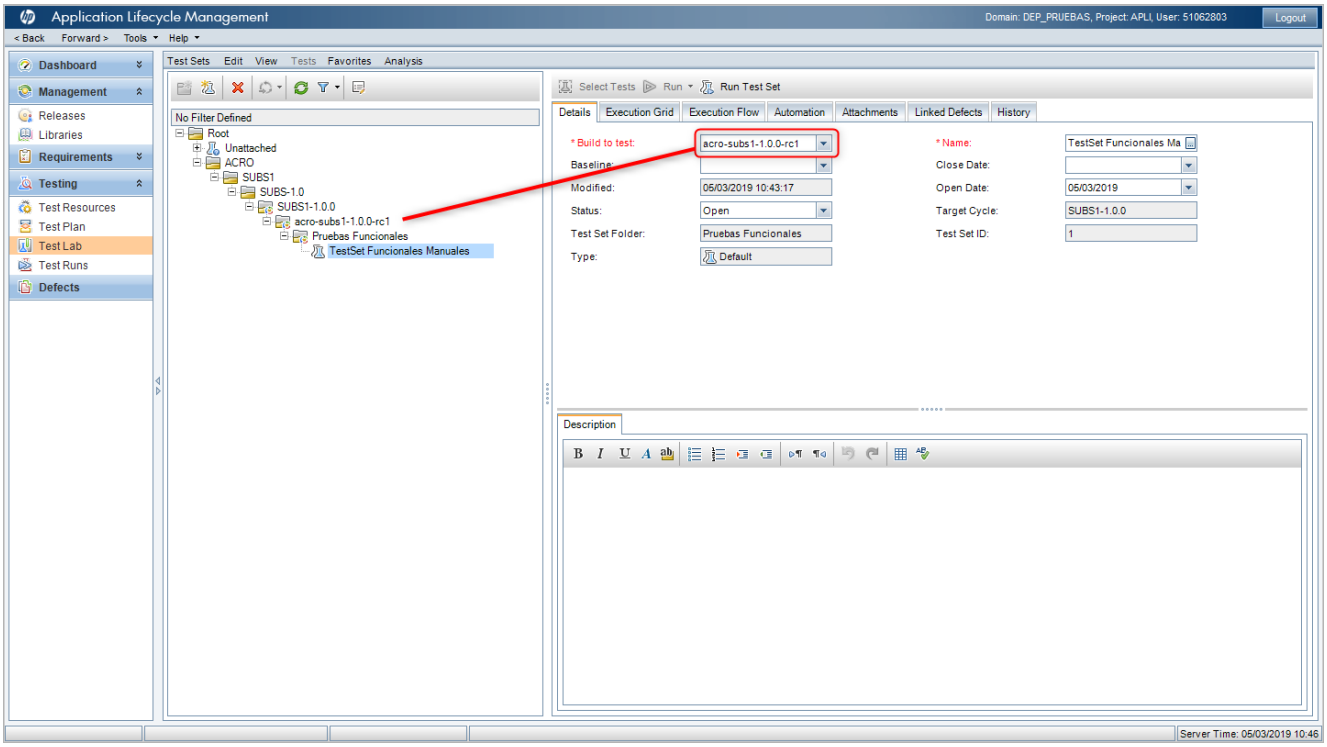
2.4.2 Creación de una batería de pruebas

Accedemos al módulo de "**Test Lab**" y estructuramos la batería de pruebas de la siguiente manera.

+ Nombre Aplicación
    + [Subsistema] (en caso de que tenga subsistemas)
        + Release
            + Ciclo
                + Build
                + Pruebas Funcionales
                + Test Set Funcionales Manuales
                + Test Set Funcionales Automatizadas
                + Pruebas Rendimiento
                + Test Set Carga
                + Test Set Sobrecarga
                + Test Set Escalabilidad
                + Pruebas Seguridad
                + Test Set Seguridad



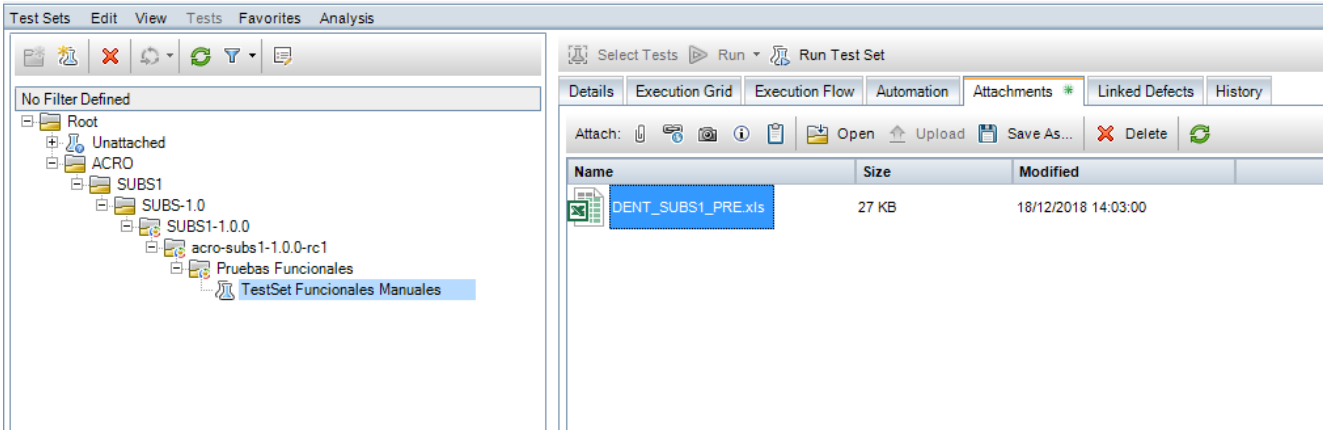
A cada Test Set deberemos asignarle la Build correspondiente dentro de las disponibles en el ciclo.



2.4.3 Ubicación ficheros de datos de entrada para ejecuciones

2.4.3.1 Pruebas funcionales Manuales

La ubicación de los ficheros con los datos de prueba para la ejecución de pruebas funcionales deberá estar en la pestaña "*Attachments*" del "*Test Set Funcionales Manuales*".

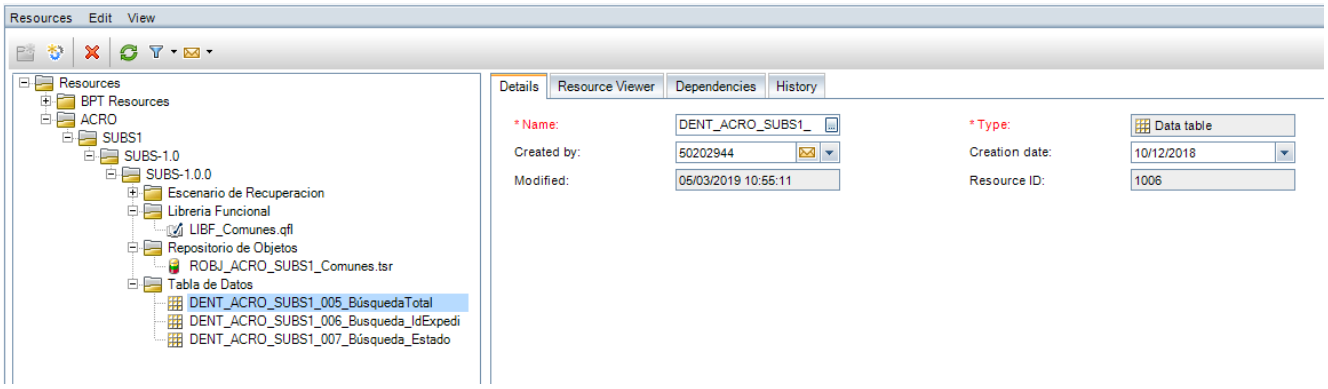


2.4.3.2 Pruebas funcionales Automatizadas

La ubicación de los ficheros con los datos de prueba para la ejecución de pruebas funcionales deberá estar en el menú "*Test Resources*" dentro del módulo "Testing". Tendremos la siguiente estructura de carpetas:

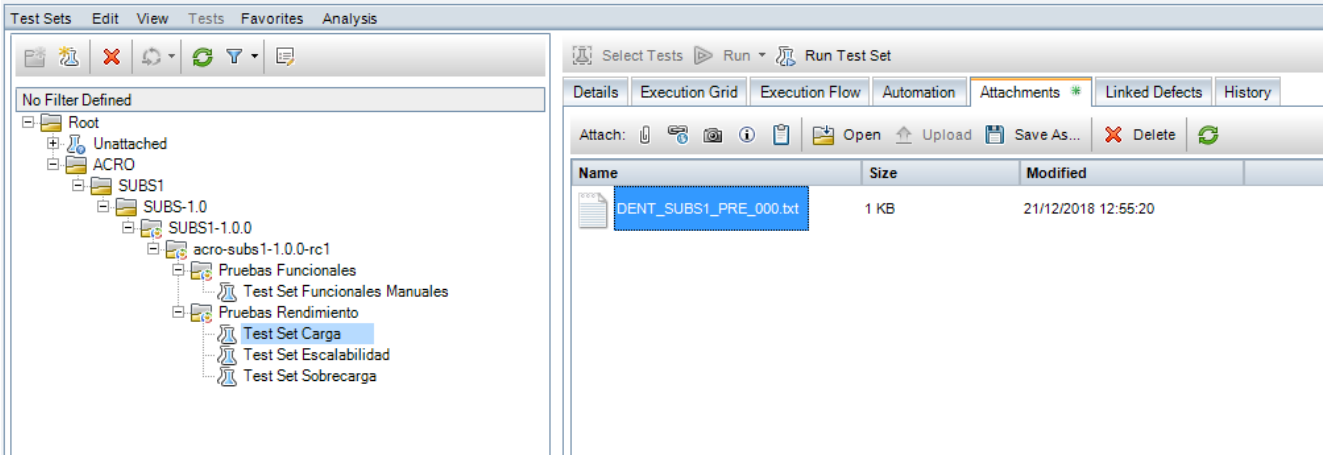
+ Nombre Aplicación
    + [Subsistema] (en caso de que tenga subsistemas)
        + Release
            + Ciclo
                + Escenario de Recuperación
                + Librería Funcional
                + Repositorio de Objetos
                + Tabla de Datos

En cada una de las carpetas se incluirán los distintos ficheros necesarios para la ejecución de cada uno de los casos de pruebas a ejecutar en el Test Set creado



2.4.3.3 Pruebas de carga

La ubicación de los ficheros con los datos de prueba para la ejecución de pruebas de carga deberá estar en la pestaña "*Attachments*" del "*Test Set Carga*".



2.4.3.4 Pruebas de sobrecarga

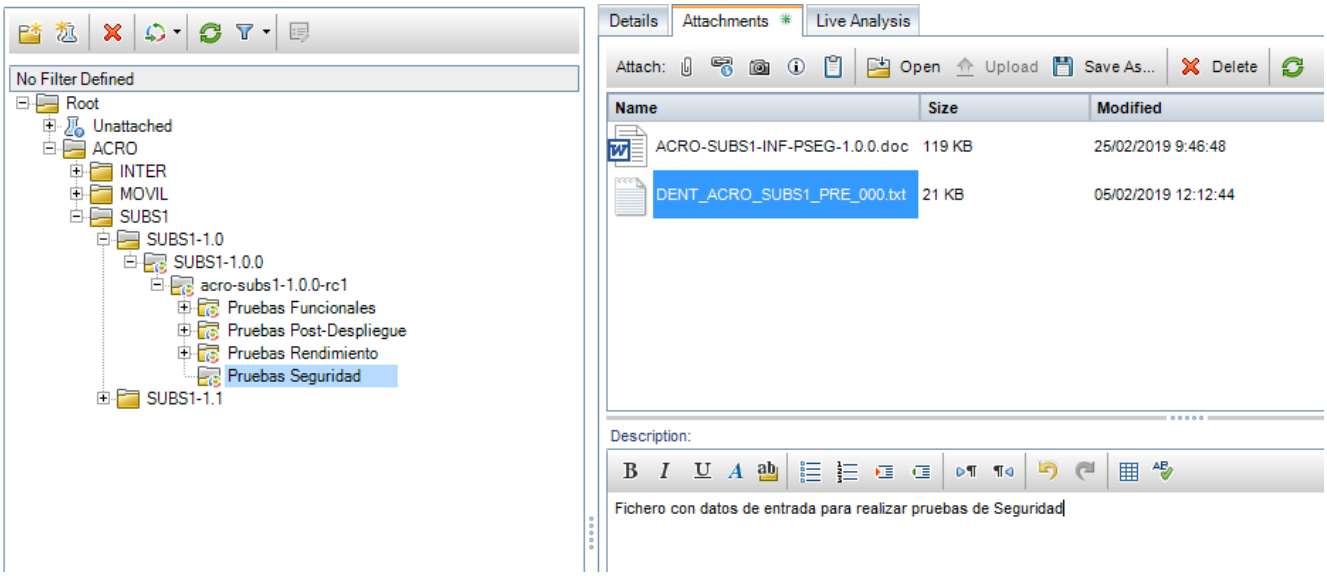
La ubicación de los ficheros con los datos de prueba para la ejecución de pruebas de sobrecarga deberá estar en la pestaña "*Attachments*" del "*Test Set Sobrecarga*".

2.4.3.5 Pruebas escalabilidad

La ubicación de los ficheros con los datos de prueba para la ejecución de pruebas de escalabilidad deberá estar en la pestaña "*Attachments*" del "*Test Set Escalabilidad*".

2.4.3.1 Pruebas seguridad

La ubicación de los ficheros con los datos de prueba para la ejecución de pruebas de seguridad deberá estar en la pestaña "*Attachments*" de la carpeta "*Pruebas Seguridad*".

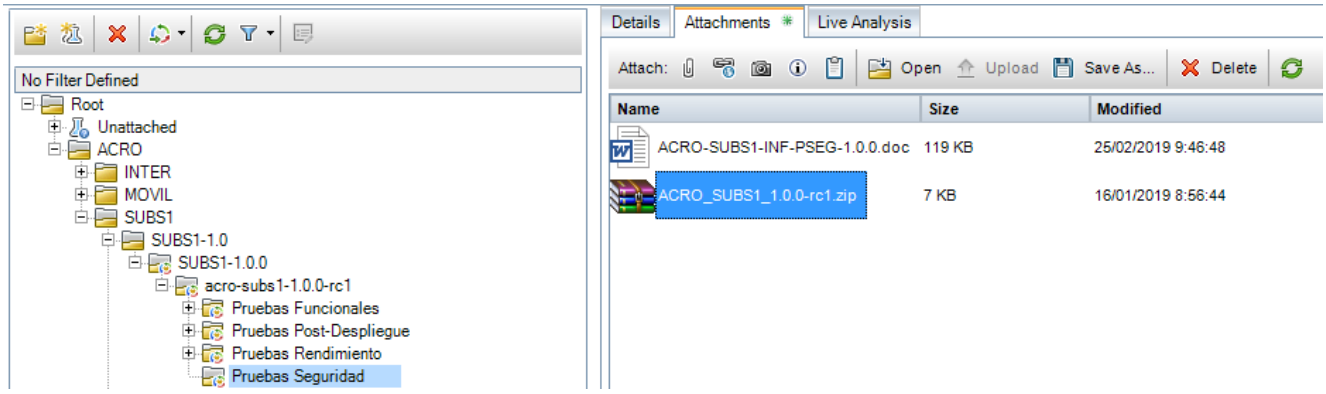


2.4.3.2 Fichero .ZIP

Si hubiera varios ficheros a anexar se podría incluir en un ZIP con el nombre de la aplicación y de la build a ejecutar.

Ejemplo:

ACRO_SUBS1_1.0.0-rc1.zip



2.4.3.3 Varias ejecuciones en una misma build

Si hubiera varias ejecuciones en una misma build la ubicación sería la misma pero se agruparía cada lanzamiento en un zip incluyendo al final un índice.

Ejemplo:

ACRO_SUBS1_1.0.0-rc1_1.zip

ACRO_SUBS1_1.0.0-rc1_2.zip

…

2.5 Módulo Defects

2.5.1 Objetivo

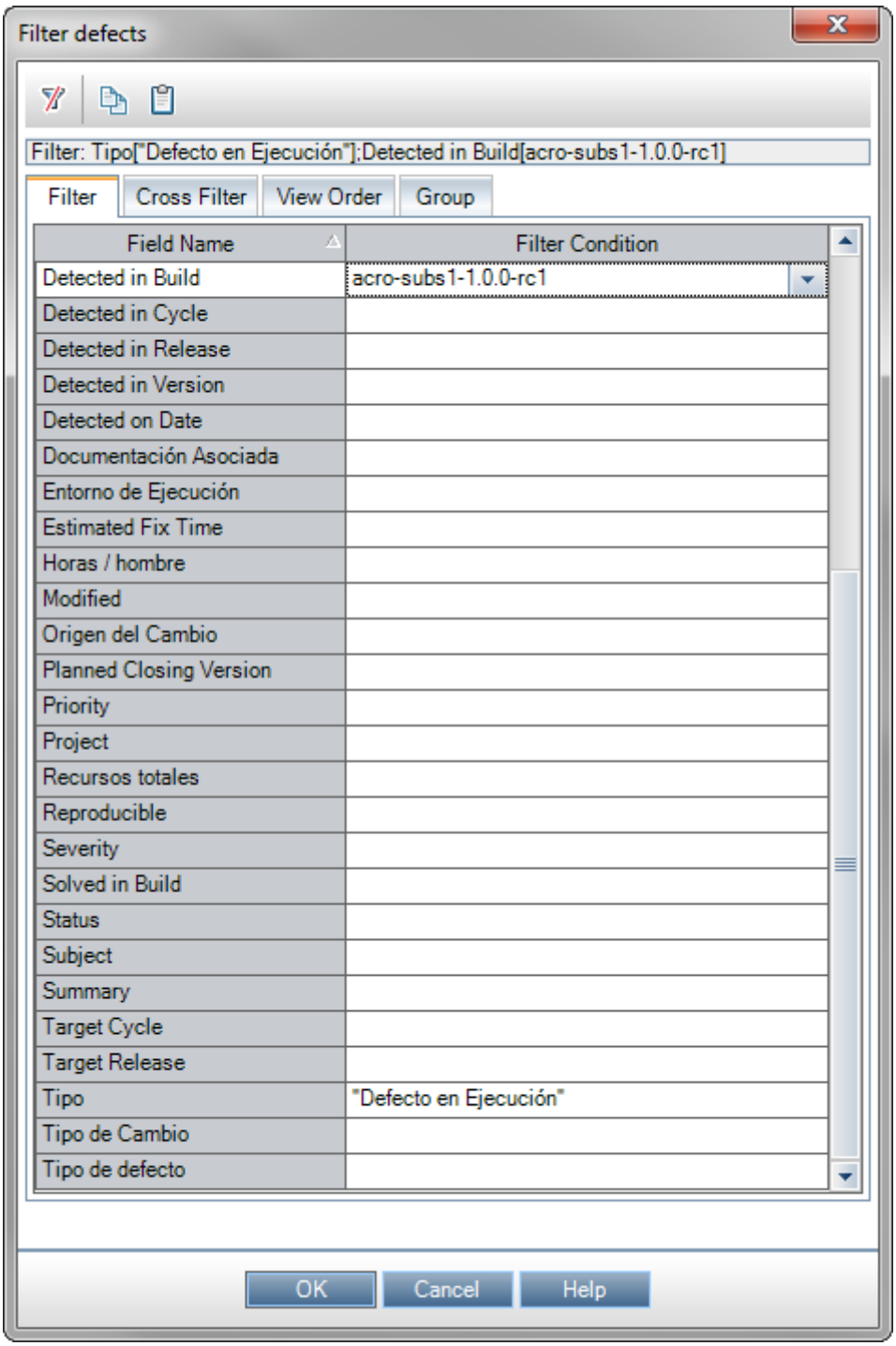
Este módulo permite el seguimiento de defectos durante todo el ciclo de vida de los mismos.

Los defectos en ejecución se definen como los errores encontrados por la Oficina de Pruebas durante la ejecución de los diferentes tipos de pruebas en el entorno de preproducción o producción.

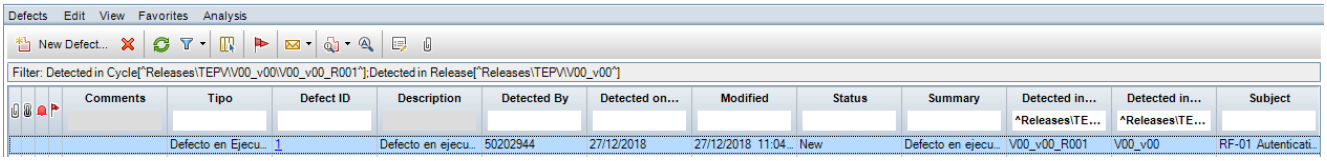
2.5.2 Visualización de defectos en una release-ciclo

Para la visualización en el grid de la entidad Defecto en Ejecución se deben realizar los siguientes pasos:

1. Ir al módulo “**Defects**”.
2. Pulsar en el botón “**Set Filter/Sort**” . Aparece la ventana “**Filter**”.
3. Seleccionar los siguientes campos:
    a. Tipo. Seleccionar el valor "**Defecto en Ejecución**".
    b. Detected in Release. Seleccionar la release sobre la que se quieren obtener los defectos dados de alta.
    c. Detected in Cycle. Seleccionar el ciclo sobre el que se quieren obtener los defectos dados de alta.
    d. Detected in Build. Seleccionar la Build concreta sobre la que se quieren obtener los defectos dados de alta

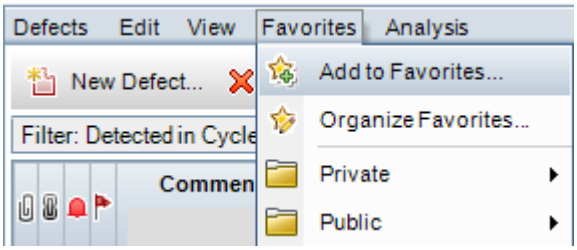


4. Pulsar el botón **OK**. En el grid aparece la lista de defectos en ejecución detectados en una release-ciclo-Build concreto.

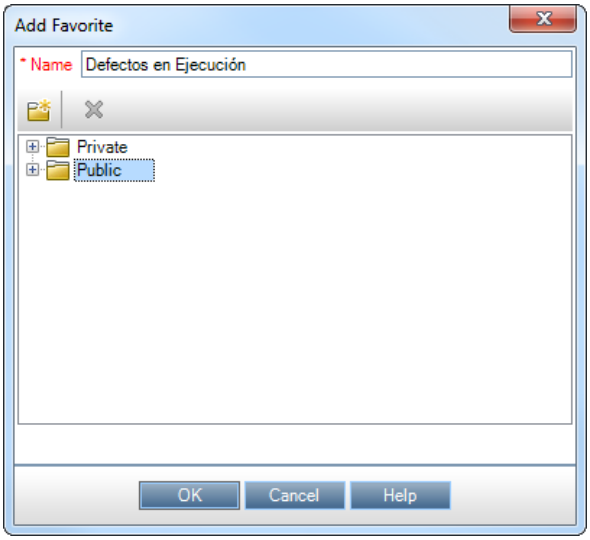


Si se quiere almacenar el filtro como una vista para un acceso más rápido a la información, se deben realizar las siguientes acciones:

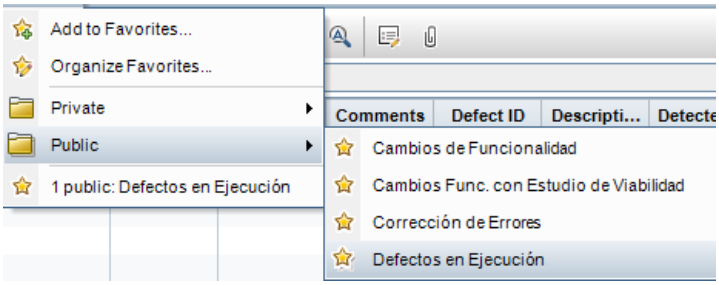
1. Ir a la barra de **Menú / Favorites** y seleccionar "**Add to Favorites**"



2. Aparece la ventana **Add Favorite** donde se introduce en el campo **Name** un nombre representativo de la búsqueda que se quiere almacenar. Seleccionar **Public** o **Private** dependiendo del ámbito de visualización de la vista.



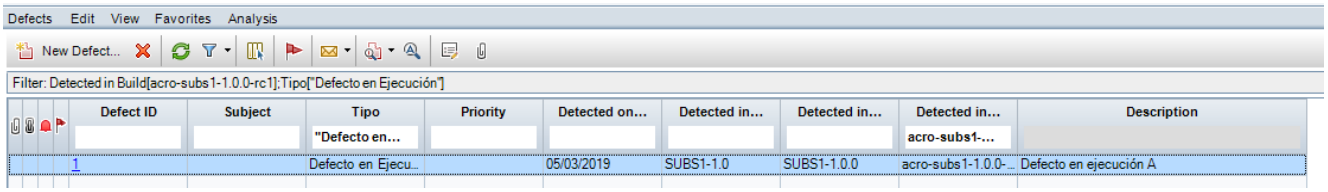
3. Pulsar el botón **OK.**
4. Para seleccionar el filtro guardado ir al campo **Favorite** y en la lista seleccionar el nombre de la vista creada.



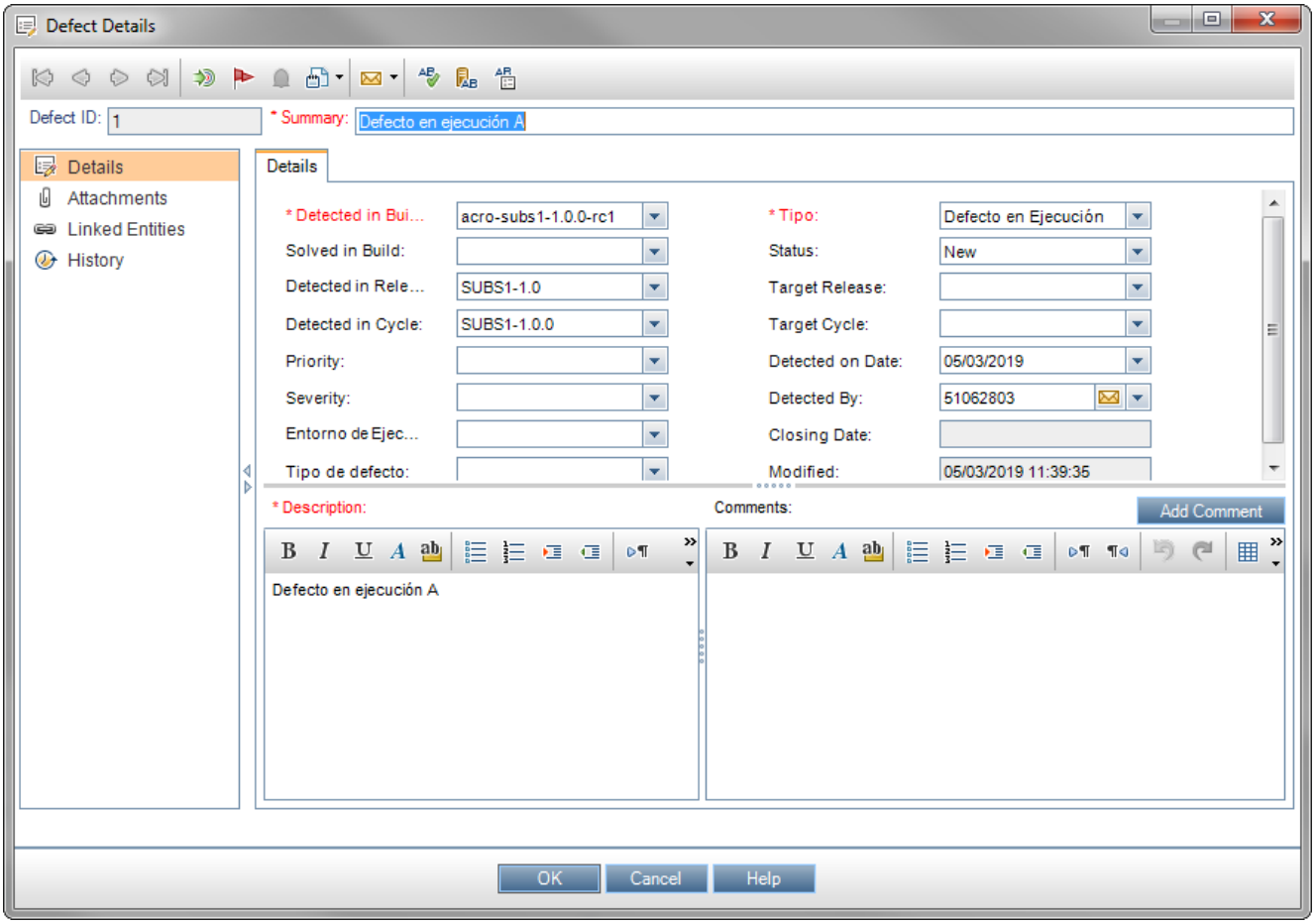
2.5.3 Modificación de defectos

Los pasos para la modificación de un defecto en ejecución son los siguientes:

1. Ir al módulo "**Defects**".
2. Seleccionar en el grid el defecto en ejecución que se quiere modificar.



3. Hacer doble clic sobre el registro del defecto en ejecución para editarlo. Se abre la ventana **Defect Details**.



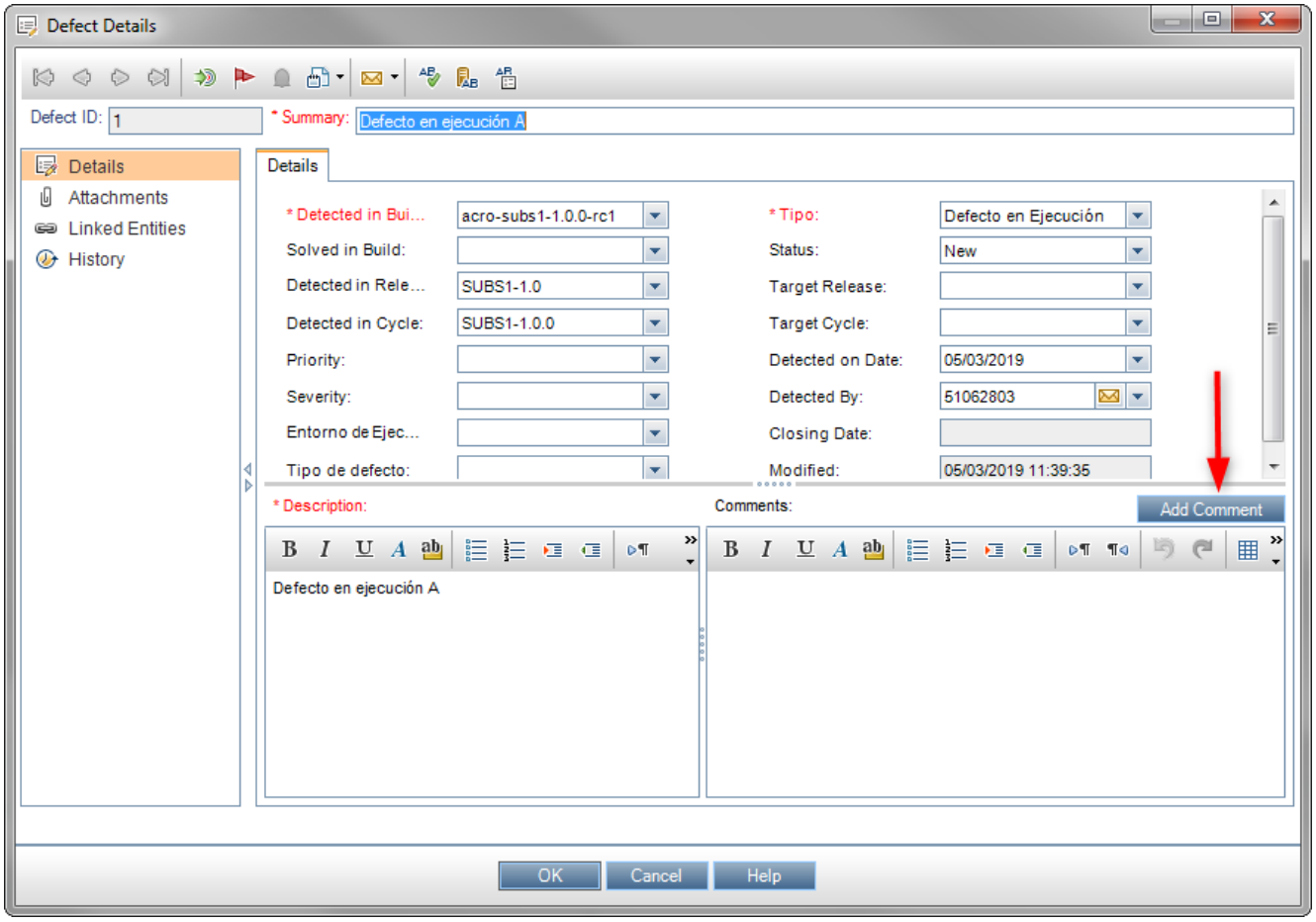
4. Se permite la modificación de los siguientes campos:
    - Summary.
    - Description.
    - Target Release.
    - Target Cycle.
    - Comments.
    - **Status**. Según el perfil del usuario el cambio de estado permitido es distinto:
        - Jefe de proyecto. De **In Progress** a **Rejected** u **Open**.
        - Desarrollo. De New a In Progress, y de Rejected by Pruebas a In Progress.
5. Una vez realiza la modificación de los campos se pulsa el botón **OK**. La ventana **Defect Details** se cierra y se retorna al grid del módulo de defecto.

2.5.4 Añadir comentarios

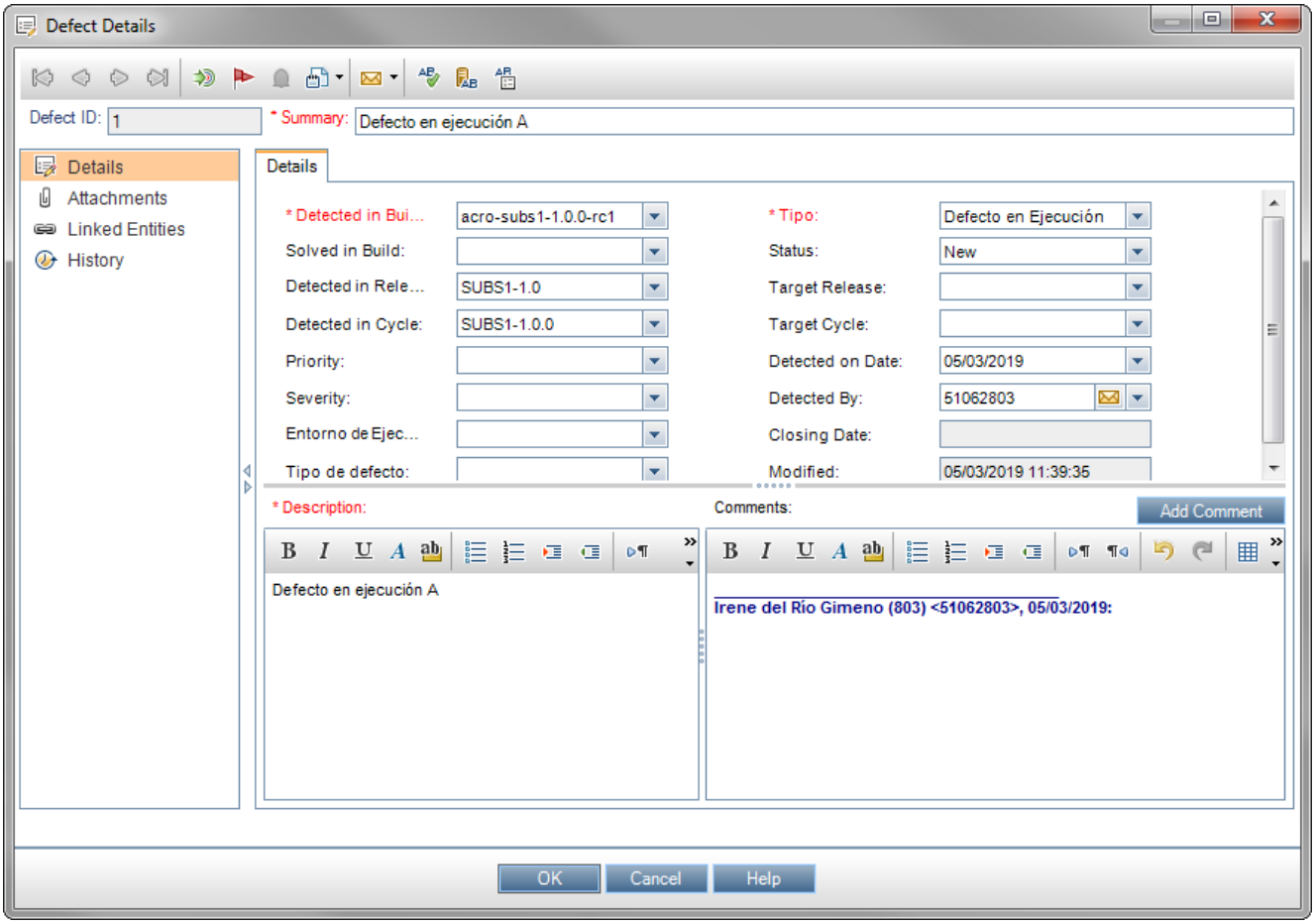
Todos los usuarios involucrados en los defectos deben introducir cualquier comentario que crean pueda servir de interés para la resolución del defecto en ejecución. Este campo también sirve para la comunicación entre los usuarios.

Los pasos para la introducción de un comentario en el campo **Comments** son los siguientes:

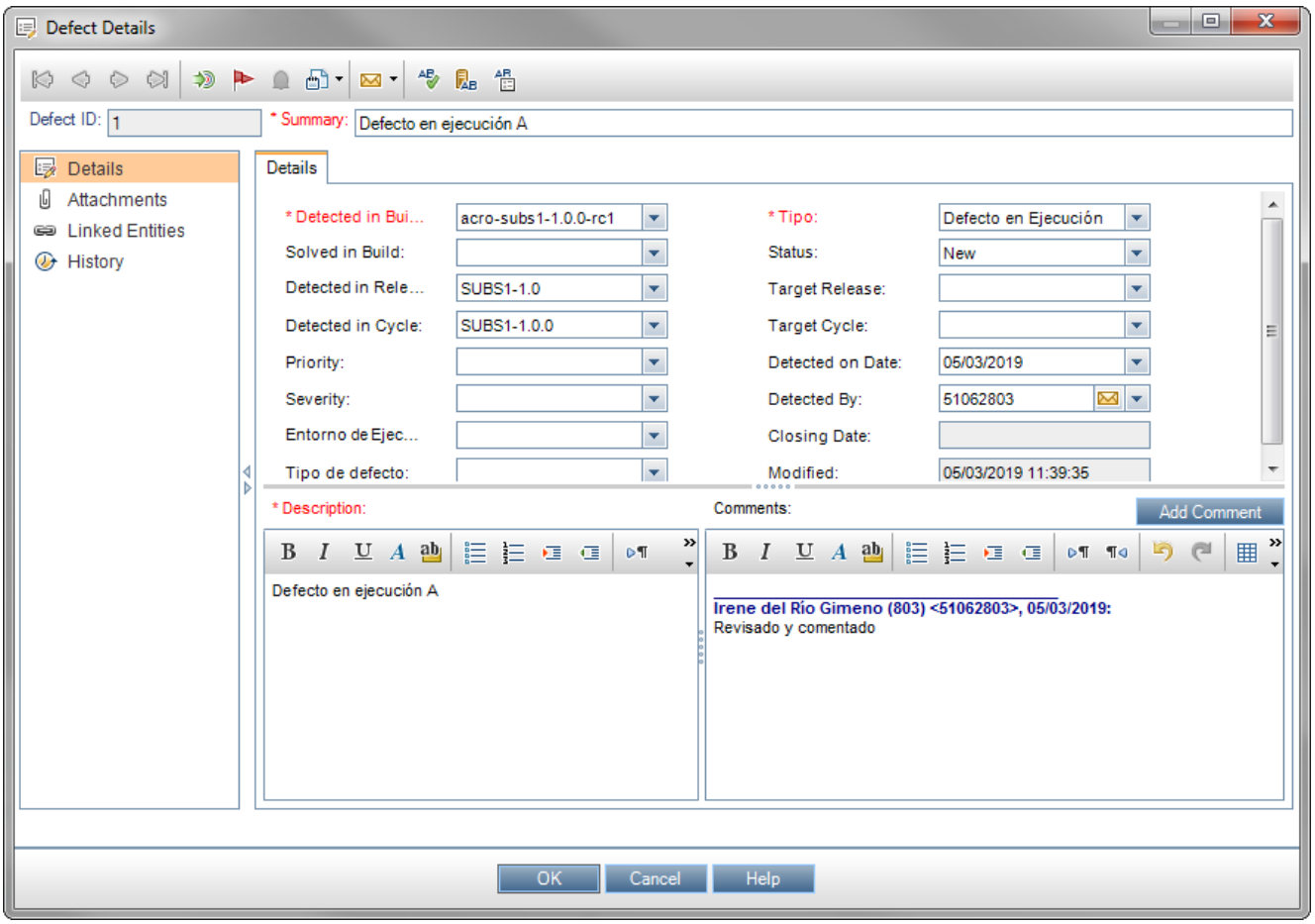
1. En la ventana **Defect Details** de un defecto en ejecución, pulsar el botón **Add Comment**.



2. En el campo **Comments** se introduce automáticamente el siguiente literal: **Nombre usuario QC <User QC>, Fecha Actual**:



3. Escribir el comentario que se quiere realizar a continuación del literal automático introducido.



4. Pulsar el botón **OK** de la ventana **Defect Details**. La ventana **Defect Details** se cierra y se retorna al grid del módulo de defecto.

3 Gestión de evolutivos

Una vez que ya se conoce cómo se gestionan los proyectos se describe la manera de proceder en ALM cuando arranca un evolutivo.

3.1 Descripción

Antes de comenzar hay que indicar que **es muy importante no modificar lo ya validado y aprobado en fases anteriores.** Cualquier cambio o modificación que se desee realizar en la

información existente en ALM deberá ser notificada por correo electrónico para su evaluación a la Oficina de Pruebas.

Las opciones posibles en un evolutivo en **ALM** (para más detalle sobre evolutivos ver anexo de gestión de la configuración) son:

- Inserción de nuevos requisitos
- Eliminación de requisitos
- Modificación de requisitos

En este apartado vamos a explicar en detalle como enlazar requisitos y casos de pruebas de las diferentes versiones manteniendo su trazabilidad.

Nuestro ámbito estará acotado a las tres posibles situaciones indicadas anteriormente y su correspondencia en ALM será la siguiente:

- Duplicar en ALM los requisitos que cambien
- Duplicar en ALM los requisitos en los que los casos de prueba asociados cambien.
- Duplicar en ALM los requisitos que están directamente relacionados (relación de primer nivel) con requisitos que cambian (trace to y trace from).
- Asociar en ALM los requisitos que no cambien, que no cambien su plan de pruebas y que no cambien los requisitos que están relacionados con la nueva versión.

Con estas pautas en ALM y ayudándonos de los filtros mantendremos la trazabilidad entre las diferentes versiones.

3.2 Ejemplo

Para detallar los pasos a realizar en los diferentes módulos nos basaremos en un proyecto de ejemplo didáctico que utilizaremos durante toda la guía.

Ejemplo:

Proyecto con dos versiones **[SUBS1-]1.0 y [SUBS1-]1.1**, siendo **[SUBS1-]1.1** un evolutivo que añade funcionalidad (RF-04 Convertir) y modifica casos de prueba de un requisito que no cambia funcionalidad (RF-02 Ver detalles del vehículo).

La estructura del proyecto en la versión **[SUBS-]1.0** (inicial) la mostramos en la siguiente figura:

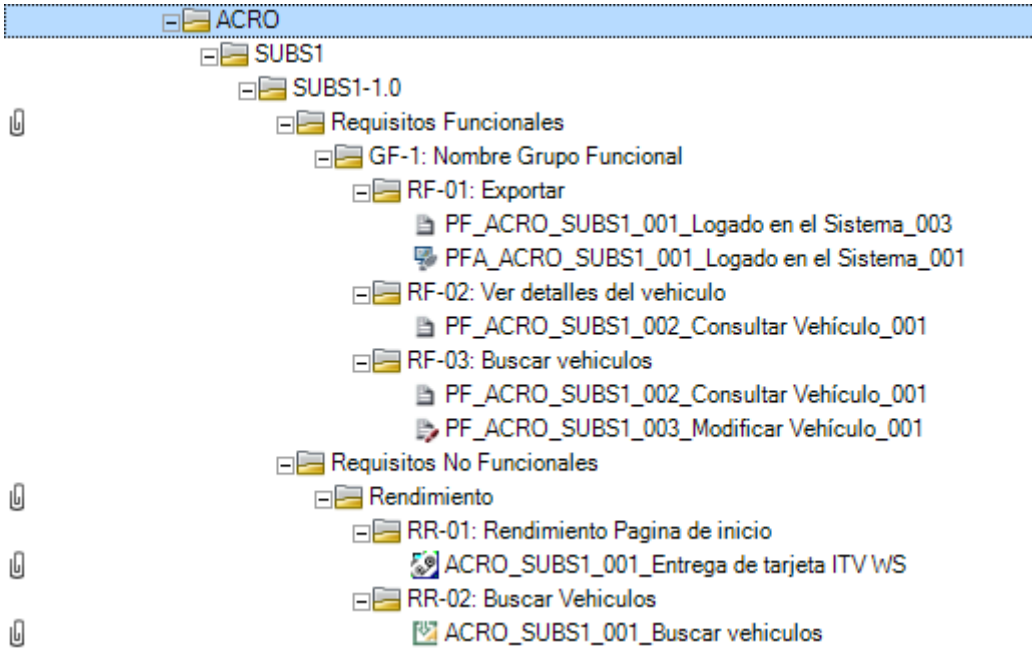


Ilustración 6. Estructura Test Plan SUBS1-1.0

3.2.1 Módulo Management

Dentro del módulo **Management** deberemos crear la nueva versión y ciclo. En este ejemplo, será la versión [SUBS-]1.1 y la revisión [SUBS-]1.1.0.

La nueva versión [SUBS-]1.1 deberá estar asociada obligatoriamente a un ciclo [SUBS-]1.1.0.

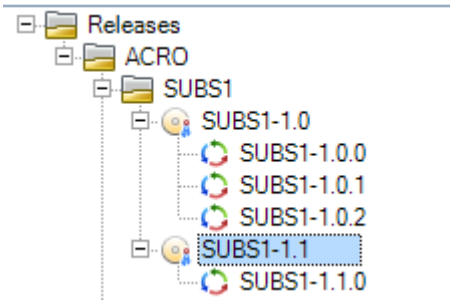


Ilustración 7. Estructura Releases [SUBS-]1.1

3.2.2 Módulo Requirements

En el módulo **Requirements** deberemos crear la correspondiente estructura con la nueva versión **[SUBS-]1.1**, tal y como se indica en la figura siguiente.

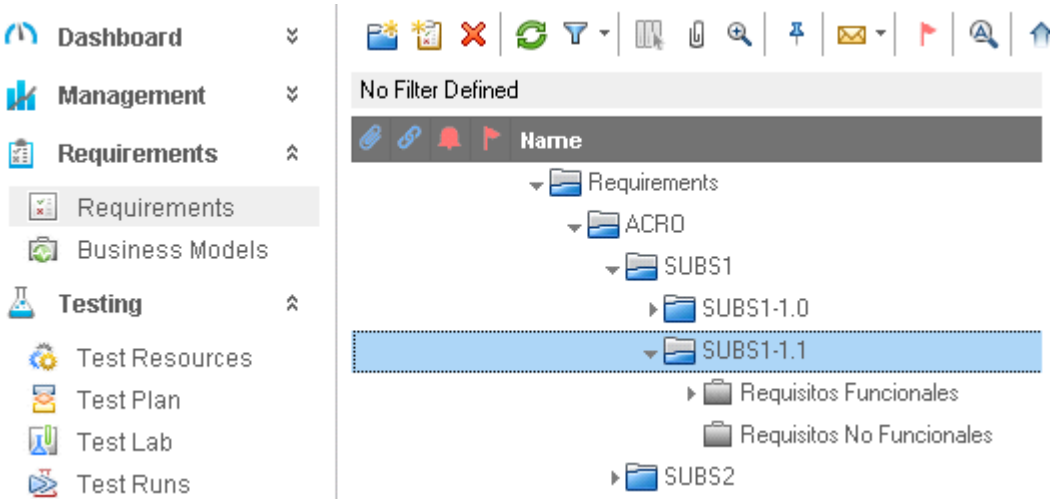


Ilustración 8. Estructura Requirements [SUBS-]1.1

En nuestro ejemplo deberemos crear:

- Un nuevo requisito funcional (RF-04 Convertir) en la versión [SUBS-]1.1
- El requisito (RF-02 Exportar) tanto si es para el caso en que cambie dicho requisito funcionalmente, o porque los casos de prueba que cubren dicho requisito en la versión [SUBS-]1.1 cambien respecto a los que tenían en la [SUBS-]1.0

- El requisito funcional (RF-01 Exportar) por estar relacionado con el requisito funcional (RF-02 Ver detalles del vehiculo).

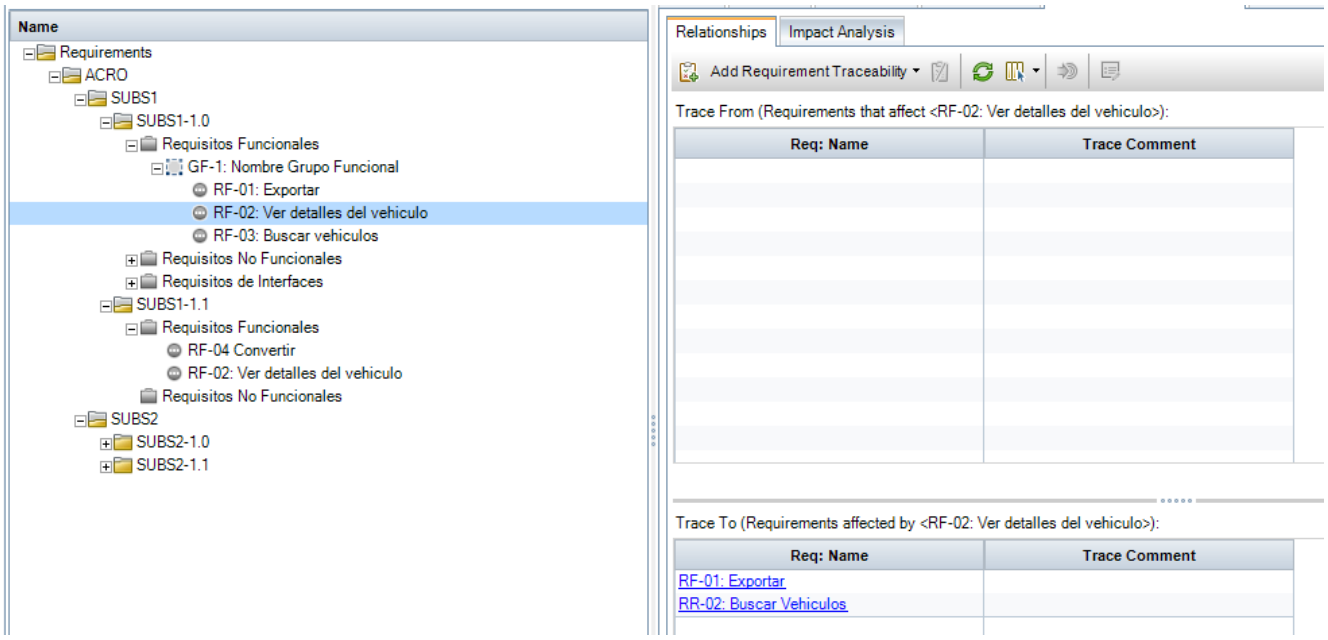


Ilustración 9. Estructura Requirements [SUBS-]1.0

Por lo tanto, la estructura nos quedaría de la siguiente manera:

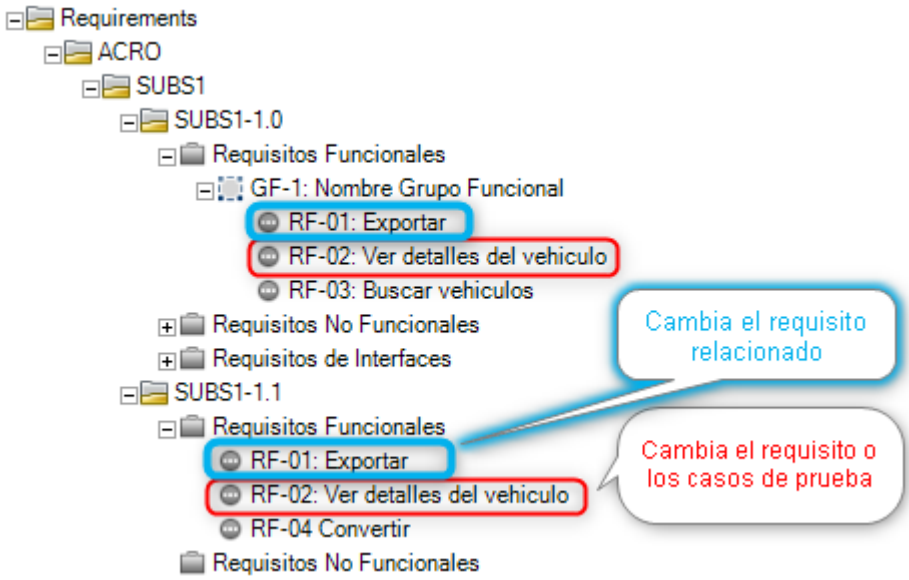


Ilustración 10. Estructura Requirements [SUBS-]1.1

Y cada requisito se debería asociar con su release correspondiente.

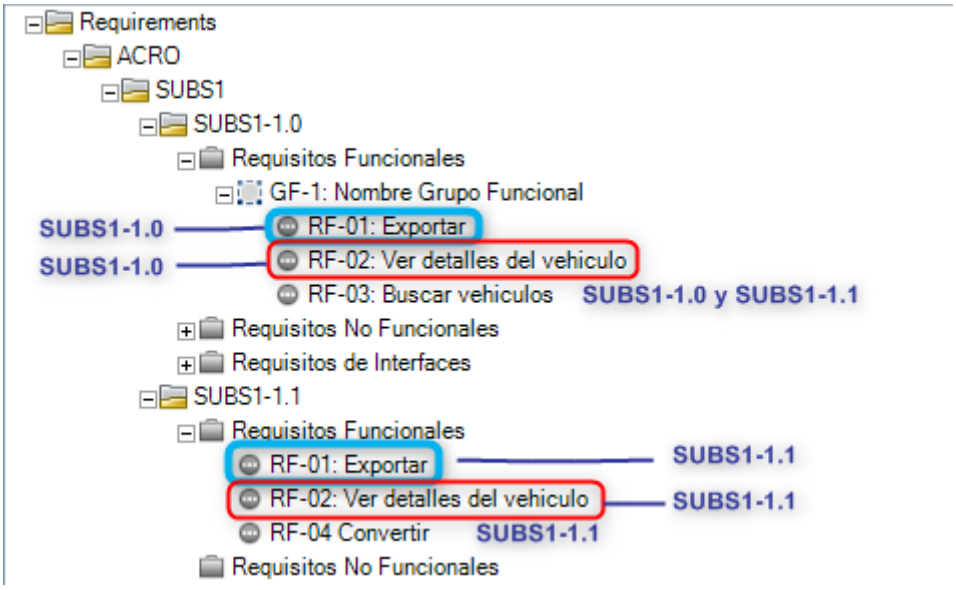


Ilustración 11. Relación de requisitos [SUBS-]1.0 y [SUBS-]-1.1

Cuando un requisito cambia porque está relacionado con otro requisito que cambia en la pestaña *"Description"* de ALM se debe indicar en una nota en la que se haga mención a dicha circunstancia.

Ejemplo:

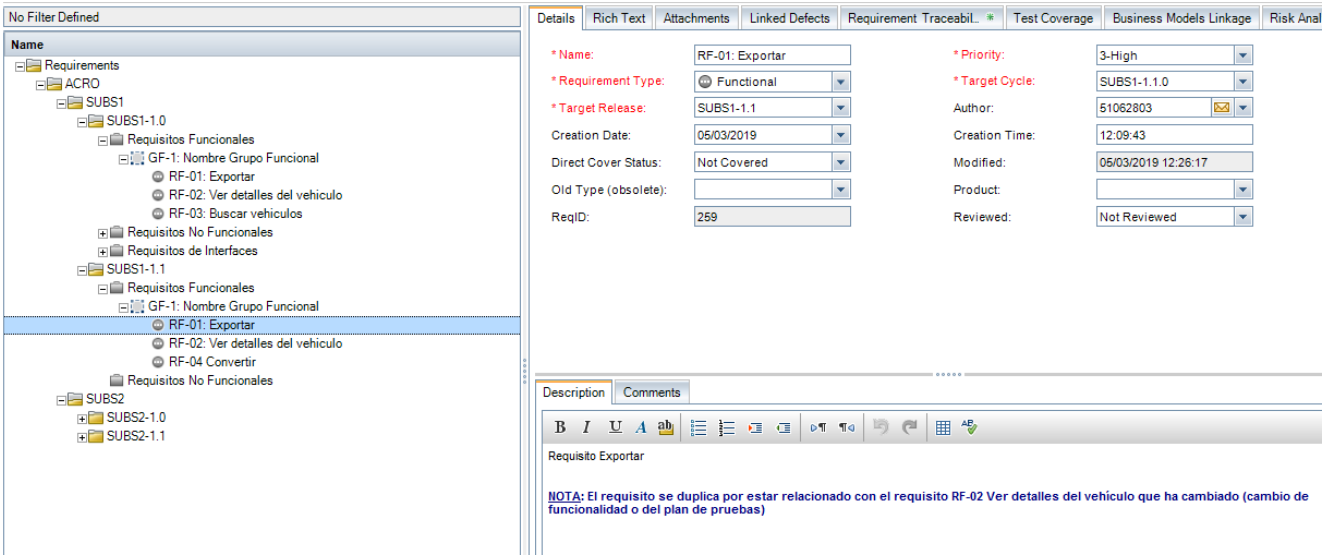


Ilustración 12. Descripción de requisitos que cambian

3.2.3 Módulo Test Plan

En el módulo Test Plan en la versión [**SUBS-]1.0** tendríamos la siguiente estructura.

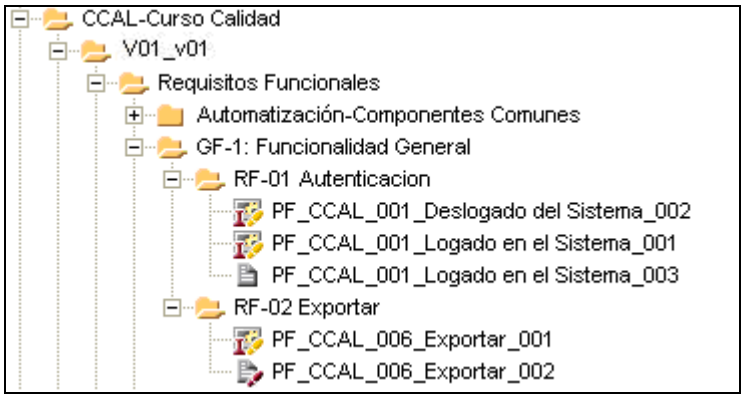


Ilustración 13. Estructura Test Plan [SUBS-]1.0

Generaríamos la estructura de carpetas para la [SUBS-]1.1 y añadiríamos la estructura de carpetas de los requisitos afectados y los casos de prueba nuevos y los que cambian.

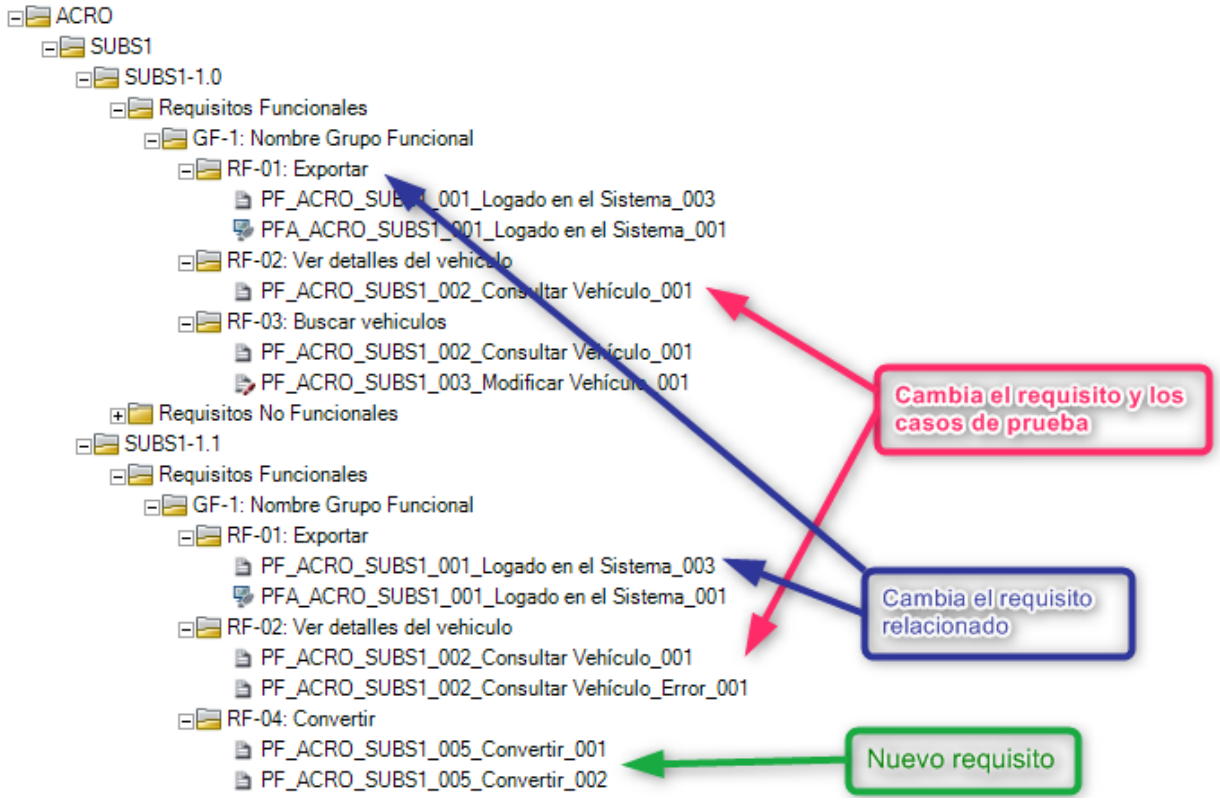


Ilustración 14. Estructura Test Plan V01_v02

Ahora debemos enlazar cada caso de prueba con su requisito correspondiente teniendo en cuenta si es de la versión anterior [SUBS-]1.0 o de la nueva [SUBS-]1.1.

3.3 Comprobación

Para comprobar que nuestro plan de pruebas está correctamente generado y enlazado con los casos de prueba que les aplica podemos hacer la siguiente comprobación.

Realizar un filtro a nivel del campo "**Target Release**" y "**Target Cycle**".

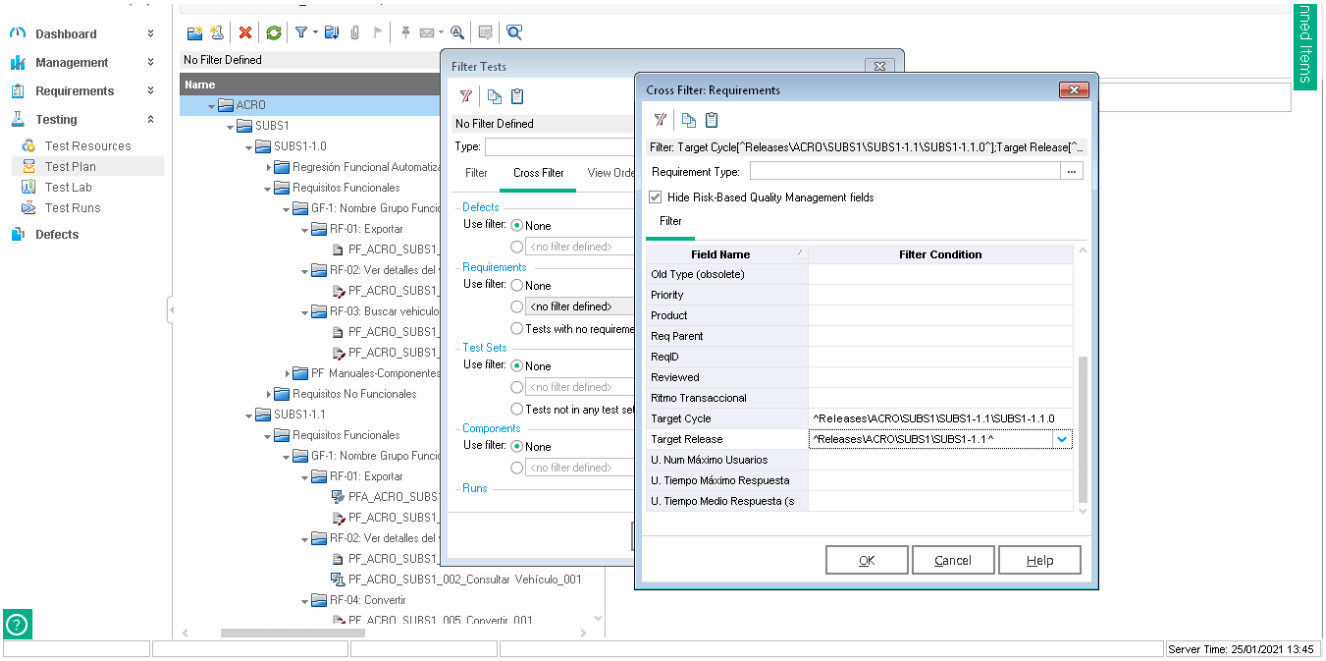


Ilustración 15. Filtro Test Plan

Al aplicar el filtro, nos deberían quedar sólo los casos de prueba que aplican a la release [SUBS-]1.1.