

INGENIERÍA DEL SOFTWARE

PLAN DE PRUEBAS

PDP_2C_v3

MIEMBROS DEL GRUPO:

- María López García
- Marcos López Lamas
- David Macías Carbajo
- Abel Martínez Rodríguez

FECHA DE ENTREGA: 06/05/2021

CONTROL DE VERSIONES		
VERSION	FECHA	DESCRIPCIÓN DEL CAMBIO
1.0	22/04/2021	Primera versión del plan de pruebas solo teniendo en cuenta Pruebas de caja negra.
2.0	30/4/2021	Segunda versión con correcciones sobre las pruebas de caja negra.
3.0	06/05/2021	Añadidas pruebas de caja blanca.

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

ÍNDICE

DESCRIPCIÓN DE LA PRÁCTICA	1
DESCRIPCIÓN DEL GRUPO DE TRABAJO	1
DOCUMENTACIÓN DE LA PRÁCTICA	2
PLAN DE PRUEBAS	2
INTRODUCCIÓN	2
REFERENCIAS	2
ELEMENTOS DEL SOFTWARE A PROBAR	2
ACTIVIDADES, TÉCNICAS Y HERRAMIENTAS	3
o Actividades	3
o Técnicas	3
o Herramientas	3
CRITERIO DE PASO/FALLO	3
o Subsistema de gestión de incidencias	3
o Subsistema de gestión de procesos	3
o Subsistema de gestión de órdenes de trabajo	3
o Subsistema de análisis y estadísticas	4
DOCUMENTOS A ENTREGAR AL FINALIZAR LAS PRUEBAS	4
ACTIVIDADES DE EJECUCIÓN Y PREPARACIÓN DE LAS PRUEBAS	4
ESQUEMA DE TIEMPOS Y RIESGOS ASUMIDOS	4
RRHH	4
NECESIDADES FORMATIVAS	5
DISEÑO DE PRUEBAS	5
SUBSISTEMA DE GESTIÓN DE INCIDENCIAS	5
o Características a probar	5
o Detalles de plan de pruebas	5
o Identificador y procedimientos de prueba	5
o Criterios de paso/fallo	6
SUBSISTEMA DE GESTIÓN DE PROCESOS	7
o Características a probar	7
o Detalles de plan de pruebas	7
o Identificador y procedimientos de prueba	7
o Criterios de paso/fallo	8
SUBSISTEMA DE GESTIÓN DE ÓRDENES DE TRABAJO	9
o Características a probar	9
o Detalles de plan de pruebas	9
o Identificador y procedimientos de prueba	9
o Criterios de paso/fallo	11
SUBSISTEMA DE GESTIÓN DE ESTADÍSTICAS	12
o Características a probar	12
o Detalles de plan de pruebas	12
o Identificador y procedimientos de prueba	12
o Criterios de paso/fallo	13
ESPECIFICACIÓN DE PROCEDIMIENTOS	13
o Procedimiento para el subsistema de gestión de incidencias	13
o Procedimiento para el subsistema de Gestión de Procesos	14
o Procedimiento para el Subsistema de Gestión de OT	14

ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

○ Procedimiento para el Subsistema de Análisis y Estadísticas	15
ANEXOS	16
ANEXO 1. –ESPECIFICACIÓN DE PRUEBAS DE CAJA BLANCA Y CAJA NEGRA	16
PRUEBAS DE CAJA NEGRA	16
○ Subsistema de gestión de incidencias	16
○ Subsistema de gestión de procesos	32
○ Subsistema de gestión de OTs	46
○ Subsistema de análisis y estadísticas	68
PRUEBAS DE CAJA BLANCA	79
○ Actualizar Incidencia	79
○ Actualizar Proceso	84
○ Crear OT	91
○ Buscar orden de trabajo	97
ANEXO 2. –BIBLIOGRAFÍA Y MATERIAL UTILIZADO.....	104

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: <i>PDP_2C_v3.doc</i>	

Descripción de la práctica

Tras analizar el proyecto de otro grupo de prácticas generaremos en grupo las pruebas que se deben realizar a ese código para comprobar que su funcionalidad esté correcta.

Descripción del grupo de trabajo

El grupo que elaborará este documento es el 2C, formado por los alumnos:

- María López García
- Marcos López Lamas
- David Macías Carbajo
- Abel Martínez Rodríguez

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

DOCUMENTACIÓN DE LA PRÁCTICA

Plan de pruebas

Introducción

Este documento tiene como objetivo redactar la documentación de las pruebas realizadas para la prueba del software del grupo **2B** a partir de su código y del documento de análisis diseño y planificación **PrjGrp2B.pdf**. Para ello se crea la planificación general del plan pruebas, la especificación del diseño de las pruebas, la especificación de los casos de prueba y la especificación del procedimiento de prueba.

Referencias

Nombre del documento	Software de visualización (versión)	Descripción del documento
PrjGrp2B.pdf	Adobe	Pdf de documentación del grupo creador del código.

Elementos del software a probar

Subsistema de gestión de incidencias

UC-0001 Crear incidencia
UC-0002 Actualizar Incidencia
UC-0003 Búsqueda de Incidencias
UC-0004 Seleccionar filtro
UC-0005 Seleccionar incidencias sin asignar
UC-0006 Obtener Incidencias

Subsistema de gestión de procesos

UC-0007 Crear proceso
UC-0008 Actualizar proceso
UC-0009 Buscar proceso
UC-0011 Obtener proceso
UC-0012 Asignar proceso/OT
UC-0013 Asignar proceso/incidencia

Subsistema de gestión de órdenes de trabajo

UC-0014 Crear orden de trabajo
UC-0016 Asignar empresa/responsable
UC-0017 Gestionar recursos
UC-0018 Obtener OTs

Subsistema de análisis y estadísticas

UC-0019 Obtener estadísticas de incidencias

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

UC-0020 Obtener estadísticas de procesos

UC-0021 Obtener estadísticas de OT

Actividades, técnicas y herramientas

○ ***Actividades***

- Pruebas de unidad (JUnit): Se desarrollan pruebas de unidad a todos los métodos de las interfaces de los subsistemas.
- Pruebas de integración/sistema: Se desarrollan pruebas que comprueban que los diferentes subsistemas funcionan correctamente

○ ***Técnicas***

Se adoptará un enfoque sistemático para la realización de estas pruebas, empleando para esto las técnicas siguientes:

Particiones o clases de equivalencia: se dividirá el dominio de los valores de entradas en un número finito de clases, de modo que la prueba de un valor representativo de una de las clases permitirá suponer que el resultado será el mismo para el resto de valores de la clase.

Análisis de valores límite: generadas las clases de equivalencia anteriores, mediante la búsqueda de valores límite, profundizaremos en las situaciones que se hallen directamente arriba, abajo y en los márgenes de dichas clases.

Conjetura de errores: esta técnica la aplicaremos finalmente para crear casos de prueba enfocados a problemas y errores que, de manera subjetiva, se considere que pueden ser cometidos por los desarrolladores.

○ ***Herramientas***

Eclipse IDE
Framework Mockito
Framework JUnit

Criterio de paso/fallo

○ ***Subsistema de gestión de incidencias***

Para superar las pruebas del subsistema de gestión de incidencias será necesario realizar el 100% de los casos de prueba y superar el 80% de los casos de prueba principales.

○ ***Subsistema de gestión de procesos***

Para superar las pruebas del subsistema de gestión de procesos será necesario realizar el 100% de los casos de prueba y superar el 80% de los casos de prueba principales.

○ ***Subsistema de gestión de órdenes de trabajo***

Para superar las pruebas del subsistema de gestión de órdenes de trabajo será necesario realizar el 100% de los casos de prueba y superar el 80% de los casos de prueba principales.

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

○ *Subsistema de análisis y estadísticas*

Para superar las pruebas del subsistema de análisis y estadísticas será necesario realizar el 100% de los casos de prueba y superar el 80% de los casos de prueba principales.

Documentos a entregar al finalizar las pruebas

Nombre del documento	Software de visualización (versión)	Descripción del documento
PDP_v3.pdf	Adobe	Documento de plan de pruebas completo
Código del proyecto con las pruebas	IDE	-
Informe ejecución de pruebas	Adobe	-
Incidencias en las pruebas	Adobe	-
Informe ejecutivo de las pruebas	Adobe	-

Actividades de ejecución y preparación de las pruebas

- Transferencia del código realizado por el grupo 2B.
- Preparación del entorno de ejecución con el IDE Eclipse + los framework de java Mockito y JUnit.
- Construcción del código de las pruebas sobre el código.
- Ejecución del código de pruebas.
- Obtención de resultados de las pruebas.

Esquema de tiempos y riesgos asumidos

El principal riesgo que tendrán las pruebas viene dado por el corto plazo de tiempo que disponemos ya que no nos permitirá probar todos los casos de prueba de cada parámetro de las diferentes funciones, por lo tanto, asumimos que errores similares en diferentes métodos se habrán resuelto correctamente si pasa la prueba para un método.

En cuanto al reparto de tareas Marcos se encargará de las pruebas asociadas al subsistema de gestión de incidencias, David de las del subsistema de gestión de procesos, María de las del subsistema de gestión de OTs y Abel de las del subsistema de análisis y estadísticas.

RRHH

- *De qué recursos disponemos y responsabilidades*

Nombre responsable	Pruebas asignadas
María López García	Subsistema de gestión de ordenes de trabajo

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Marcos López Lamas	Subsistema de gestión de incidencias
David Macías Carbajo	Subsistema de gestión de procesos
Abel Martínez Rodríguez	Subsistema de gestión de estadísticas

Necesidades Formativas

- Conocimiento del estándar IEE 829
- Conocimiento básico de Java
- Conocimiento básico de JUnit

Diseño de pruebas

Subsistema de gestión de incidencias

- ***Características a probar***
 - UC-0001 Crear incidencia**
 - UC-0002 Actualizar Incidencia**
 - UC-0003 Búsqueda de Incidencias**
 - UC-0004 Seleccionar filtro**
 - UC-0005 Seleccionar incidencias sin asignar**
 - UC-0006 Obtener Incidencias**

- ***Detalles de plan de pruebas***

Para obtener los diferentes casos de prueba usamos las técnicas de clases de equivalencia, análisis de valores límites y por último conjetura de errores.

- ***Identificador y procedimientos de prueba***

PRU-0001 Pruebas sobre inicialización de incidencias

- CP-0001 Introducir todos los argumentos correctos.
- CP-0002 Introducir todos los argumentos a null.
- CP-0003 Introducir un identificador negativo
- CP-0004 Introducir un nombre de ciudadano con caracteres no alfabéticos.
- CP-0005 Introducir un nombre de ciudadano con más de 100 caracteres.
- CP-0006 Introducir un DNI con el formato incorrecto.
- CP-0007 Introducir un DNI con más de 9 caracteres.
- CP-0008 Introducir un teléfono sin prefijo.
- CP-0009 Introducir un teléfono con caracteres no numéricos.
- CP-0010 Introducir un teléfono con más de 15 caracteres.
- CP-0011 Introducir una descripción con más de 240 caracteres.
- CP-0012 Introducir una localización con más de 150 caracteres.
- CP-0013 Introducir un TipoIncidencia con la cadena "Iluminación".
- CP-0014 Introducir un TipoIncidencia con la cadena "Parques".
- CP-0015 Introducir un TipoIncidencia con la cadena "Asfalto".
- CP-0016 Introducir un TipoIncidencia con la cadena "Alcantarillado".
- CP-0017 Introducir un TipoIncidencia con la cadena "Tráfico".
- CP-0018 Introducir un TipoIncidencia con la cadena "Suministro".
- CP-0019 Introducir un TipoIncidencia con una cadena distinta a las válidas.

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

CP-0020 Introducir una Fecha Inicio con un formato no válido.

PRU-0002 Pruebas sobre creación de incidencias

CP-0021 Introducir incidencia con todos sus argumentos correctos.

CP-0022 Introducir incidencia que sea null.

CP-0023 Introducir un identificador repetido.

CP-0024 Introducir una Fecha Inicio anterior al día actual.

PRU-0003 Pruebas sobre actualización de incidencias

CP-0025 Introducir incidencia con todos sus argumentos correctos.

CP-0026 Introducir incidencia que sea null.

PRU-0004 Pruebas sobre búsqueda de incidencias

CP-0027 Introducir incidencia con todos sus argumentos correctos.

CP-0028 Introducir incidencia que sea null.

PRU-0005 Pruebas sobre obtener incidencias sin asignar

CP-0029 Llamar a obtener incidencias sin asignar con incidencias sin asignar almacenadas

CP-0030 Llamar a obtener incidencias sin asignar sin incidencias sin asignar almacenadas

○ Criterios de paso/fallo

PRU-0001 Pruebas sobre inicialización de incidencias

- Paso: Se devuelve una incidencia con todos los parámetros introducidos.
- Fallo: Se devuelve incidencia con los parámetros introducidos aun siendo estas combinaciones no permitidas.

PRU-0002 Pruebas sobre creación de incidencias

- Paso: Se devuelve la incidencia introducida.
- Fallo: Se devuelve la incidencia introducida con alguna combinación de parámetros errónea o repitiendo un identificador.

PRU-0003 Pruebas sobre actualización de incidencias

- Paso: Se devuelve una incidencia con los argumentos actualizados
- Fallo: Se devuelve una incidencia con los argumentos sin actualizar, mal actualizados o incompletos.

PRU-0004 Pruebas sobre búsqueda de incidencias

- Paso: Se devuelve una lista de incidencias coincidentes con los parámetros a buscar.
- Fallo: Se devuelve una lista de incidencias vacía o incompleta habiendo incidencias almacenadas que cumplen los filtros. Se devuelve una lista con incidencias que no cumplen los parámetros a buscar

PRU-0005 Pruebas sobre obtener incidencias sin asignar

- Paso: Se devuelve una lista de incidencias sin asignar

ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

- Fallo: Se devuelve una lista de incidencias sin asignar vacía o incompleta habiendo incidencias sin asignar almacenadas.

Subsistema de gestión de procesos

○ ***Características a probar***

Inicializar proceso

UC-0007 Crear proceso

UC-0008 Actualizar proceso

UC-0009 Buscar proceso

UC-0012 Asignar proceso/OT

UC-0013 Asignar proceso/incidencia

○ ***Detalles de plan de pruebas***

Para obtener los diferentes casos de prueba usamos las técnicas de clases de equivalencia, análisis de valores límites y por último conjetura de errores.

○ ***Identificador y procedimientos de prueba***

PRU-0006 Prueba sobre Inicializar proceso

- CP-0031 Introducir argumentos correctos
- CP-0032 Introducir argumentos null
- CP-0033 Introducir identificador erróneo (negativo)
- CP-0034 Introducir nombreProceso erróneo (vacío)
- CP-0035 Introducir coste erróneo (negativo)
- CP-0036 Introducir estado no existente
- CP-0037 Introducir incidencias null
- CP-0038 Introducir fechaInicio pasada.
- CP-0039 Introducir fechaInicio errónea.

PRU-0007 Prueba sobre Crear proceso

- CP-0040 Introducir proceso null
- CP-0041 Introducir proceso correcto
- CP-0042 Introducir identificador repetido

PRU-0008 Prueba sobre Actualizar proceso

Incluye los casos de prueba:

- CP-0043 Introducir proceso null
- CP-0044 Introducir proceso correcto

PRU-0009 Prueba sobre Buscar proceso

Incluye los casos de prueba:

- CP-0045 Introducir proceso null
- CP-0046 Introducir proceso correcto

PRU-0010 Prueba sobre Asignar proceso/OT

Incluye los casos de prueba:

- CP-0047 Introducir argumentos correctos.
- CP-0048 Introducir proceso null.

ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

CP-0049 Introducir órdenes de trabajo vacías.
CP-0050 Introducir órdenes de trabajo null.
CP-0051 Introducir órdenes de trabajo repetidas.

PRU-0011 Prueba sobre Asignar proceso/incidencia -> dependencia de inicializar proceso

Incluye los casos de prueba:

CP-0052 Introducir argumentos correctos.
CP-0053 Introducir proceso null.
CP-0054 Introducir incidencias vacías.
CP-0055 Introducir incidencias null.
CP-0056 Introducir incidencias repetidas.

○ Criterios de paso/fallo

PRU-0006 Prueba sobre Inicializar proceso

- Paso: Se crea y devuelve un Proceso con los atributos completos a excepción del atributo “Coste” que puede tener el valor null.
- Fallo: Se devuelve un objeto Proceso con los atributos incompletos o un se devuelve un tipo null.

PRU-0007 Prueba sobre Crear proceso

- Paso: Se crea el proceso en cuestión y se almacena de forma correcta en el sistema. Este proceso será único. Se devuelve el proceso creado con sus atributos correspondientes inicializados a valores correctos.
- Fallo: No se crea el proceso, o lo hace de manera incorrecta, esto es, no almacenándolo en el sistema o haciéndolo de forma incorrecta o devolviendo un objeto Proceso con sus atributos incorrectos. Se considera fallo si el proceso creado se encuentra repetido, aunque el almacenamiento y los valores de los atributos sean correctos.

PRU-0008 Prueba sobre Actualizar proceso

- Paso: Se actualiza el proceso correctamente en la memoria del sistema. Se devuelve el proceso actualizado con todos sus atributos con valores correctos.
- Fallo: No se actualiza el proceso o lo hace erróneamente. Se devuelve un objeto Proceso que no se corresponde con el proceso actualizado o contiene valores erróneos.

PRU-0009 Prueba sobre Buscar proceso

- Paso: Se devuelve una lista de Procesos correctos y únicos en base al filtro proporcionado, donde cada uno de los elementos de la lista, debe cumplir sus condiciones. En el caso de que no se contemplen procesos que cumplan el filtro se podrá devolver una lista vacía.
- Fallo: Se devuelve una lista vacía habiendo procesos que cumplen el filtro proporcionad. Los valores de los procesos devueltos son incorrectos.

PRU-0010 Prueba sobre Asignar proceso/OT

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

- Paso: Se asignan de forma correcta las órdenes de trabajo al proceso en cuestión, esto es que se actualiza el atributo “OrdenesTrabajo” del objeto Proceso de forma correcta, con la lista de OTs indicada. Además, se actualiza en la memoria del sistema.
- Fallo: No se asignan adecuadamente las órdenes de trabajo al proceso. No se devuelve el objeto Proceso con los valores de sus atributos correctos y actualizados. No se actualiza en la memoria del sistema.

PRU-0011 Prueba sobre Asignar proceso/incidencia

- Paso: Se asignan de forma correcta las incidencias al proceso en cuestión, esto es que se actualiza el atributo “Incidencias” del objeto Proceso de forma correcta, con la lista de incidencias indicada. Además, se actualiza en la memoria del sistema.
- Fallo: No se asignan adecuadamente las incidencias al proceso. No se devuelve el objeto Proceso con los valores de sus atributos correctos y actualizados. No se actualiza en la memoria del sistema.

Subsistema de gestión de órdenes de trabajo

○ ***Características a probar***

UC-0014 Crear orden de trabajo
UC-0016 Asignar empresa/responsable
UC-0017 Gestionar recursos
UC-0018 Obtener OTs
Inicializar Orden de Trabajo
Inicializar Presupuesto

○ ***Detalles de plan de pruebas***

Para obtener los diferentes casos de prueba usamos las técnicas de clases de equivalencia, análisis de valores límites y por último conjetura de errores.

○ ***Identificador y procedimientos de prueba***

Pruebas de caja negra:

PRU-0012 Inicializar Orden de Trabajo

Incluye los casos de prueba:

CP-0057 Introducir todos los argumentos correctos
 CP-0058 Introducir todos los argumentos nulos
 CP-0059 Introducir identificador negativo
 CP-0060 Introducir descripción con más de 500 caracteres
 CP-0061 Introducir responsable con caracteres no alfabéticos
 CP-0062 Introducir personal negativo
 CP-0063 Introducir fecha de inicio con formato incorrecto
 CP-0064 Introducir duración negativa
 CP-0065 Introducir estado “Pendiente de Asignación”
 CP-0066 Introducir estado con cadena distinta a las anteriores

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

PRU-0013 Inicializar presupuestos

Incluye los casos de prueba:

- CP-0067 Introducir todos los argumentos correctos
- CP-0068 Introducir todos los argumentos nulos
- CP-0069 Introducir identificador de OT inexistente
- CP-0070 Introducir identificador repetido
- CP-0071 Introducir identificador negativo
- CP-0072 Introducir empresa con más de 100 caracteres
- CP-0073 Introducir presupuesto negativo
- CP-0074 Introducir fecha de inicio con formato incorrecto
- CP-0075 Introducir fecha de inicio anterior a la actual
- CP-0076 Introducir duración negativa
- CP-0077 Introducir ArrayList material vacío
- CP-0078 Introducir personal negativo

PRU-0014 Crear Orden de Trabajo

Incluye los casos de prueba:

- CP-0079 Introducir una Orden de Trabajo nula
- CP-0080 Introducir una Orden de Trabajo con los argumentos obligatorios cubiertos
- CP-0081 Introducir una Orden de Trabajo con todos los argumentos correctos
- CP-0082 Introducir una Orden de Trabajo con identificador ya existente
- CP-0083 Introducir proceso
- CP-0084 Introducir fecha de inicio anterior a la actual

PRU-0015 Asignar Empresa

Incluye los casos de prueba:

- CP-0085 Introducir una OT y un Presupuesto con sus campos obligatorios cubiertos
- CP-0086 Introducir los dos argumentos nulos
- CP-0087 Introducir un presupuesto que no estaba almacenado
- CP-0088 Introducir una OT y un presupuesto sin que haya 3 presupuestos almacenados
- CP-0089 Introducir una orden de trabajo que no exista

PRU-0016 Gestionar recursos

Incluye los casos de prueba:

- CP-0090 Introducir una orden de trabajo que no exista en el sistema
- CP-0091 Introducir una orden de trabajo que exista, con campos modificados
- CP-0092 Introducir una orden de trabajo que exista, sin campos modificados

PRU-0017 Obtener OTs / Buscar

Incluye los casos de prueba:

ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

CP-0093 Obtener todas las OTs, sin filtrar (null)

CP-0094 Obtener todas las OTs, filtradas usando una OT con todos los campos a null

CP-0095 Obtener todas las OTs, filtradas usando una OT con algunos campos cubiertos

○ *Criterios de paso/fallo*

PRU-0012 Inicializar Orden de Trabajo

- Paso: Si se han introducido valores correctos como argumentos, se crea una Orden de Trabajo con los campos indicados inicializados y se devuelve esa OrdenTrabajo. Si se introducen valores incorrectos, no se permite crear la Orden de Trabajo y se devuelve null.
- Fallo: La prueba falla cuando se devuelve algo distinto a lo especificado en el Paso.

PRU-0013 Inicializar presupuestos

- Paso: Si se han introducido valores correctos como argumentos, se crea un Presupuesto con los campos indicados inicializados y se almacena en el sistema, devolviéndose ese Presupuesto. Si se introducen valores incorrectos o no se introducen los campos obligatorios, no se permite crear el Presupuesto y se devuelve null.
- Fallo: La prueba falla cuando se devuelve algo distinto a lo especificado en el Paso.

PRU-0014 Crear orden de trabajo

- Paso: Si se ha introducido una Orden de Trabajo correcta, se almacena esa Orden de Trabajo en el sistema y se devuelve esa misma OrdenTrabajo. Si se introduce una Orden de Trabajo incorrecta, no se permite almacenar la Orden de Trabajo y se devuelve null.
- Fallo: La prueba falla cuando se devuelve algo distinto a lo especificado en el Paso.

PRU-0015 Asignar Empresa

- Paso: Si se han introducido valores correctos como argumentos, la Orden de Trabajo introducida se ha actualizado con los campos de presupuesto necesarios y se devuelve esa nueva versión de la OrdenTrabajo. Si se introducen valores incorrectos, no se permite modificar la Orden de Trabajo, y se devolverá la antigua.
- Fallo: La prueba falla cuando se devuelve algo distinto a lo especificado en el Paso.

PRU-0016 Gestionar recursos

- Paso: Si se ha introducido una OrdenTrabajo con valores correctos como argumentos, se modifica la Orden de Trabajo con ese identificador con los campos indicados modificados y se devuelve esa OrdenTrabajo. Si se introducen valores incorrectos, no se permite modificar la Orden de Trabajo y se devuelve la antigua.

ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

- Fallo: La prueba falla cuando se devuelve algo distinto a lo especificado en el Paso.

PRU-0017 Obtener OTs

- Paso: Si se introduce null como argumento, se devuelve la lista de todas las Órdenes de Trabajo existentes en el sistema. Si se introduce un filtro, se devuelven Órdenes de Trabajo que cumplan las condiciones impuestas en los campos del filtro.
- Fallo: La prueba falla cuando se devuelve algo distinto a lo especificado en el Paso.

Subsistema de gestión de estadísticas

○ ***Características a probar***

UC-0019 Obtener estadísticas de incidencias

UC-0020 Obtener estadísticas de procesos

UC-0021 Obtener estadísticas de OT

○ ***Detalles de plan de pruebas***

Para obtener los diferentes casos de prueba usamos las técnicas de clases de equivalencia y conjetura de errores. En cuanto a resultados, además de contabilizar las pruebas pasadas sin fallos se contará el tiempo que se tardan en ejecutar las funcionalidades para comprobar si se cumple el requisito no funcional del sistema.

○ ***Identificador y procedimientos de prueba***

PRU-0018 Prueba sobre Obtener estadísticas de incidencias

Incluye los casos de prueba:

- CP-0096 Introducir todos los argumentos correctos
- CP-0097 Introducir un rango de fechas con formato incorrecto
- CP-0098 Introducir distribución semana
- CP-0099 Introducir distribución incorrecta
- CP-0100 Introducir filtro correcto

PRU-0019 Prueba sobre las estadísticas de procesos

Incluye los casos de prueba:

- CP-0101 Introducir todos los argumentos correctos
- CP-0102 Introducir un rango de fechas con formato incorrecto
- CP-0103 Introducir distribución semana
- CP-0104 Introducir distribución incorrecta
- CP-0105 Introducir filtro correcto

PRU-0020 Prueba sobre Obtener estadísticas de OTs

Incluye los casos de prueba:

- CP-0106 Introducir todos los argumentos correctos
- CP-0107 Introducir un rango de fechas con formato incorrecto
- CP-0108 Introducir distribución semana
- CP-0109 Introducir distribución incorrecta

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

CP-0110 Introducir filtro correcto

○ *Criterios de paso/fallo*

PRU-0018 Prueba sobre Obtener estadísticas de incidencias

- Paso: Introduciendo los parámetros adecuados se devuelve la clase Estadística correcta y si los parámetros son incorrectos se devuelve una clase null.
- Fallo: Cuando el método devuelva un valor diferente a los antes mencionado la prueba fallará.

PRU-0019 Prueba sobre Obtener estadísticas de procesos

- Paso: Introduciendo los parámetros adecuados se devuelve la clase Estadística correcta y si los parámetros son incorrectos se devuelve una clase null.
- Fallo: Cuando el método devuelva un valor diferente a los antes mencionado la prueba fallará.

PRU-0020 Prueba sobre Obtener estadísticas de OTs

- Paso: Introduciendo los parámetros adecuados se devuelve la clase Estadística correcta y si los parámetros son incorrectos se devuelve una clase null.
- Fallo: Cuando el método devuelva un valor diferente a los antes mencionado la prueba fallará.

Especificación de procedimientos

○ *Procedimiento para el subsistema de gestión de incidencias*

El objetivo de este procedimiento es comprobar que la incidencia se puede crear, actualizar y almacenar. Así como dotar al sistema la capacidad de búsqueda de incidencias mediante filtros e incidencias sin asignar. Para ello ejecutan los siguientes casos de prueba:

PRU-0001 Pruebas sobre inicialización de incidencias {CP-0001 – CP-00023}

PRU-0002 Pruebas sobre creación de incidencias {CP-00024 – CP-00025}

PRU-0003 Pruebas sobre actualización de incidencias {CP-00026 – CP-00027}

PRU-0004 Pruebas sobre búsqueda de incidencias {CP-00028 – CP-00029}

PRU-0005 Pruebas sobre obtener incidencias sin asignar {CP-00030 – CP-00031}

Las pruebas se ejecutan de forma independiente unas con otras y ejecutando sus casos uno a uno de manera secuencial siendo estos también independientes entre sí.

Para iniciar la ejecución se describen previamente las entradas que se van a inyectar en las funciones.

Los resultados se evaluarán contando el número de casos de prueba pasadas con éxito.

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: <i>PDP_2C_v3.doc</i>	

○ ***Procedimiento para el subsistema de Gestión de Procesos***

El objetivo de este procedimiento es comprobar que el proceso se puede crear, actualizar y almacenar. Así como dotar al sistema la capacidad de búsqueda de procesos mediante filtros. También se comprueba que el sistema asigna de forma correcta las incidencias y las órdenes de trabajo a un proceso. Para ello ejecutan los siguientes casos de prueba:

PRU-0006 Prueba sobre Inicializar proceso {CP-0032 - CP-0040}
PRU-0007 Prueba sobre Crear proceso {CP-0041 - CP-0042}
PRU-0008 Prueba sobre Actualizar proceso {CP-0043 - CP-0044}
PRU-0009 Prueba sobre Buscar proceso {CP-0045 - CP-0046}
PRU-0010 Prueba sobre Asignar proceso/OT {CP-0047 – CP-0051}
PRU-0011 Prueba sobre Asignar proceso/incidencia {CP-0052 - CP-0056}

Las pruebas se ejecutan de forma independiente unas con otras y ejecutando sus casos uno a uno de manera secuencial siendo estos también independientes entre sí.

Para iniciar la ejecución se describen previamente las entradas que se van a inyectar en las funciones.

Los resultados se evaluarán contando el número de casos de prueba pasadas con éxito.

○ ***Procedimiento para el Subsistema de Gestión de OT***

El objetivo de este procedimiento es comprobar que las OT se pueden crear, modificar y almacenar, además de los presupuestos. También se probarán las búsquedas con y sin filtrar. Para ello se agrupan ejecutan los casos de prueba que se listaron y especificaron antes, agrupados en 6 grupos, PRU-0012 Inicializar Orden de Trabajo, PRU-0013 Inicializar presupuestos, PRU-0014 Crear Orden de Trabajo, PRU-0015 Asignar Empresa, PRU-0016 Gestionar recursos y PRU-0017 Obtener OTs.

Estas pruebas se ejecutarán de forma consecutiva, y serán independientes unas de otras, pero algunas exigen ciertos prerrequisitos. Por ejemplo, si se va a crear una OT que incluye presupuestos, deben existir Presupuestos en el sistema. Sucederá lo mismo con Procesos.

Este procedimiento se encargará de comprobar que los métodos inicializar (OT), inicializar (Presupuesto), crear, gestionarRecursos, asignarEmpresa y buscar funcionan correctamente; es decir, que todos los casos de prueba asociados concluyen con el resultado esperado. En este procedimiento trataremos los casos de prueba del CP-57 al CP-74 en la primera prueba, CP-75 al CP-85 en la segunda prueba, del CP-86 al CP-89 en la tercera, del CP-90 al CP-94 en la cuarta, del CP-95 al CP-97 en la quinta y del CP-98 al CP-100 en la última prueba.

Para llevarlo a cabo, tendremos que llamar repetidamente al método probado, con los parámetros especificados en los casos de prueba. Para evitar cortes en la ejecución se introducirán los métodos a probar en sentencias try-catch para evitar que alguna

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

excepción termine el proceso antes de tiempo. Los resultados se evaluarán contando el número de casos de prueba pasadas con éxito con respecto a las fallidas.

○ *Procedimiento para el Subsistema de Análisis y Estadísticas*

Agrupar las pruebas **PRU-0018 Prueba sobre Obtener estadísticas de incidencias**, **PRU-0019 Prueba sobre Obtener estadísticas de procesos** y **PRU-0020 Prueba sobre Obtener estadísticas de OTs** para estas tres pruebas se sigue el mismo procedimiento variando simplemente la Clase sobre la que se trabaja siendo Incidencia, Proceso y OrdenTrabajo respectivamente.

Este procedimiento se encargará de comprobar que los métodos obtenerEstadísticasIncidencias, obtenerEstadísticasProcesos y obtenerEstadísticasOrdenesTrabajo funcionan correctamente, es decir todos los casos de prueba asociados concluyen con el resultado esperado. En este procedimiento trataremos los casos de prueba del CP-101 al CP-105 en la primera prueba, CP-106 al CP-110 en la segunda prueba y por último del CP-111 al CP-115 en la última prueba.

Para ejecutar este procedimiento necesitaremos tener previamente una incidencia, proceso u orden de trabajo registrada en el sistema y conocer sus valores para saber exactamente cuál será la salida de las estadísticas, esta Clase será la especificada en los prerequisites. Para asegurarnos de no tener resultados erróneos debemos vaciar la lista de incidencias, procesos u orden de trabajo antes de comenzar las pruebas.

Para llevarlo a cabo tendremos que llamar repetidamente al método probado, con los parámetros especificados en los casos de prueba. Además de esto tendremos que añadir una variable para cronometrar el tiempo que tarda en realizarse cada operación, que evaluaremos para saber si cumple el requisito no funcional del sistema.

Para evitar cortes en la ejecución se introducirán los métodos a probar en sentencias try-catch para evitar que alguna excepción termine el proceso antes de tiempo.

Los resultados se evaluarán contando el número de casos de prueba pasadas con éxito y comparando el valor de tiempo obtenido con el límite de 1 segundo que se exige en el requisito no funcional.

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

ANEXOS

Anexo 1. –Especificación de pruebas de caja blanca y caja negra

Pruebas de caja negra

- Subsistema de gestión de incidencias

PRU-0001 Pruebas sobre inicialización de incidencias

Métodos involucrados
public Incidencia inicializar (Integer identificador, String ciudadano, String DNI, String telefono, String descripcion, String localización, String tipoIncidencia, Proceso proceso, Date fechaInicio);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
identificador	Integer	Número entero > 0	Número entero <= 0
			null
nombreCiudadano	String	Longitud de la cadena <= 100	Longitud de la cadena > 100
			null
DNI	String	Longitud de la cadena = 9 con los 8 primeros caracteres como números y el último como letra	Longitud de la cadena <> 9
			8 primeros caracteres contienen caracteres no numéricos
			El último carácter no es una letra
			null
telefono	String	Longitud de la cadena = 15. Prefijo + número. Todos los caracteres son numéricos	Longitud de la cadena <> 15
			Caracteres no numéricos
			null
descripcion	String	Longitud de la cadena <= 240	Longitud de la cadena > 240
			null
localizacion	String	Longitud de la cadena <= 150	Longitud de la cadena > 150
			null
tipoIncidencia	String	Una de las siguientes cadenas: {"Iluminación", "Parques", "Asfalto", "Alcantarillado", "Tráfico", "Suministro", "Otra"}	Cadenas distintas a las anteriores
			null
proceso	Proceso	Objeto de tipo Proceso	
		null	
fechaInicio	Date	Objeto de tipo Date	null

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir argumentos correctos	CP-0001
Descripción: Introducir todos los argumentos de la función de manera correcta y comprobar que nos devuelva una incidencia con lo que hemos introducido.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = { identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021 }	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir argumentos a null	CP-0002
Descripción: Introducir todos los argumentos de la función a null y comprobar que devuelve un objeto incidencia a null.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = null ciudadano = null DNI = null telefono = null descripcion = null localizacion = null tipoIncidencia = null proceso = null fechaInicio = null	
Especificación de salidas esperadas: Incidencia incidencia = null	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir identificador negativo	CP-0003
Descripción: Introducir un identificador negativo para comprobar que no se puede crear una incidencia	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = -1 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir nombre ciudadano no alfabético	CP-0004
Descripción: Introducir un nombre de ciudadano que tenga caracteres no alfabéticos.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano1" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Probaremos el método java asociado a la funcionalidad
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "513199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021
Especificación de salidas esperadas: Incidencia incidencia = null
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir teléfono sin prefijo	CP-0008
Descripción: Introducir un teléfono que no tenga prefijo en su número	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir teléfono con caracteres no numéricos	CP-0009
Descripción: Introducir un número de teléfono que tenga caracteres no numéricos	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "3460a564255" descripcion = "descripción de prueba"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Iluminación" proceso = null fechaInicio = 22/04/2021 }
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir tipoIncidencia = "Parques"	CP-0014
Descripción: Introducir un tipoIncidencia con el nombre = "Parques"	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Parques" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = { identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Parques" proceso = null fechaInicio = 22/04/2021 }	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir tipoIncidencia = "Asfalto"	CP-0015
Descripción: Introducir un tipoIncidencia con el nombre = "Asfalto"	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Asfalto" proceso = null fechaInicio = 22/04/2021
Especificación de salidas esperadas: Incidencia incidencia = { identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Asfalto" proceso = null fechaInicio = 22/04/2021 }
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir tipoIncidencia = "Alcantarillado"	CP-0016
Descripción: Introducir un tipoIncidencia con el nombre = "Alcantarillado"	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Alcantarillado" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = { identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Alcantarillado" proceso = null fechaInicio = 22/04/2021 }	
Necesidades del entorno	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: <i>PDP_2C_v3.doc</i>	

El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.

Prerrequisitos: Ninguno

Introducir tipoIncidencia = “Tráfico”	CP-0017
Descripción: Introducir un tipoIncidencia con el nombre = “Tráfico”	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = “ciudadano” DNI = “53199271H” telefono = “34608564255” descripcion = “descripción de prueba” localizacion = “localización de prueba” tipoIncidencia = “Tráfico” proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = { identificador = 0 ciudadano = “ciudadano” DNI = “53199271H” telefono = “34608564255” descripcion = “descripción de prueba” localizacion = “localización de prueba” tipoIncidencia = “Iluminación” proceso = null fechaInicio = 22/04/2021 }	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir tipoIncidencia = “Suministro”	CP-0018
Descripción: Introducir un tipoIncidencia con el nombre = “Parques”	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = “ciudadano” DNI = “53199271H” telefono = “34608564255” descripcion = “descripción de prueba” localizacion = “localización de prueba” tipoIncidencia = “Suministro” proceso = null	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

fechaInicio = 22/04/2021
Especificación de salidas esperadas: Incidencia incidencia = { identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Suministro" proceso = null fechaInicio = 22/04/2021 }
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir tipoIncidencia no válido	CP-0019
Descripción: Introducir un tipoIncidencia con el nombre = "qwerty", que es no válido	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "qwerty" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir fechaInicio con formato no válido	CP-0020
Descripción: Introducir una fecha de inicio con un formato no válido	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

telefono = "34608564255"
descripcion = "descripción de prueba"
localizacion = "localización de prueba"
tipoIncidencia = "Iluminación"
proceso = null
fechaInicio = 22*04*2021
Especificación de salidas esperadas: Incidencia incidencia = null
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

PRU-0002 Pruebas sobre creación de incidencias

Métodos involucrados
public Incidencia crear (Incidencia incidencia);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
Incidencia	Incidencia	Incidencia con un identificador no repetido y fecha >= Hoy	Identificador repetido
			Fecha < Hoy
			null

Introducir una Incidencia válida	CP-0021
Descripción: Introducir una incidencia válida, es decir, que haya sido inicializada correctamente	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: Incidencia incidencia = { identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021 }	
Especificación de salidas esperadas: Incidencia incidencia = { identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021 }
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir un null	CP-0022
Descripción: Introducir un null como argumento	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: Incidencia incidencia = null	
Especificación de salidas esperadas: Incidencia incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir identificador repetido	CP-0023
Descripción: Introducir un identificador con un valor que ya haya sido almacenado por el sistema.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021	
Especificación de salidas esperadas: Incidencia incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Tener almacenada una incidencia con un identificador a 0	

Introducir fechaInicio < hoy	CP-0024
Descripción: Introducir una fecha de inicio con una fecha menor al día actual	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Elementos software a probar: Probaremos el método java asociado a la funcionalidad
Especificación de entradas: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Iluminación" proceso = null fechaInicio = 21/04/2021
Especificación de salidas esperadas: Incidencia incidencia = null
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

PRU-0003 Pruebas sobre actualización de incidencias

Métodos involucrados
public Incidencia actualizar (Incidencia incidencia);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
incidencia	Incidencia	Un objeto de tipo incidencia	null

Introducir una Incidencia válida	CP-0025
Descripción: Introducir una incidencia válida, es decir, que haya sido inicializada correctamente	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: Incidencia incidencia = { identificador = Indiferente, no se va a modificar ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021 }	
Especificación de salidas esperadas: Incidencia incidencia = { identificador = Valor inmutable ciudadano = "ciudadano" }	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Otra" proceso = null fechaInicio = 22/04/2021 }
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir un null	CP-0026
Descripción: Introducir un null como argumento	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: Incidencia incidencia = null	
Especificación de salidas esperadas: Incidencia incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

PRU-0004 Pruebas sobre búsqueda de incidencias

Métodos involucrados
public ArrayList<Incidencia> buscar (Incidencia filtro);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
filtro	Incidencia	Un objeto de tipo incidencia	null

Introducir una Incidencia válida	CP-0027
Descripción: Introducir una incidencia válida, es decir, que haya sido inicializada correctamente	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: Incidencia incidencia = { ciudadano = "ciudadano" }	
Especificación de salidas esperadas: ArrayList<Incidencia> incidencia = {incidencia1, incidencia2, ...}	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Prerrequisitos: Tener 1 o más incidencias para buscar
--

Introducir un null	CP-0028
Descripción: Introducir un null como argumento	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas: Icendencia incidencia = null	
Especificación de salidas esperadas: ArrayList<Icendencia> incidenciasSinFiltrar = { incidencia1, incidencia2, ... } (Todas las incidencias sin filtrar)	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

PRU-0005 Pruebas sobre obtener incidencias sin asignar

Métodos involucrados
public ArrayList<Icendencia> obtenerIcendenciaSinAsignar() ;

Llamar a obtenerIcenciasSinAsignar() con incidencias sin asignar almacenadas	CP-0029
Descripción: Introducir una incidencia válida, es decir, que haya sido inicializada correctamente	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas:	
Especificación de salidas esperadas: ArrayList<Icendencia> incidencia = { incidencia1, incidencia2, ... }	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Tener 1 o más incidencias sin asignar almacenadas	

Llamar a obtenerIcenciasSinAsignar() con incidencias sin asignar no almacenadas	CP-0030
Descripción: Introducir una incidencia válida, es decir, que haya sido inicializada correctamente	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad	
Especificación de entradas:	
Especificación de salidas esperadas: ArrayList<Icendencia> incidencia = null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

- Subsistema de gestión de procesos

PRU-0006 Prueba sobre Inicializar proceso

Métodos involucrados
public Proceso inicializar (Integer identificador, String nombreProceso, String descripcion, Double coste, Double estimado, Sstring estado, String responsable, ArrayList<Incidencia> incidencias, ArrayList<OrdenTrabajo> ordenesTrabajo, Date fechaInicio);

Entrada	Tipo de dato	Válido	No válido
identificador	Integer	Número entero positivo	Número entero negativo
			null
nombreProceso	String	Longitud de la cadena <= 100	Longitud de la cadena > 100
			null
descripcion	String	Longitud de la cadena <= 500	Longitud de la cadena > 500
			null
coste	Double	Decimal positivo	Decimal negativo
		null	
estimado	Double	Decimal positivo	Decimal negativo
			null
estado	String	Presente en la lista: {"Pendiente", "En ejecución", "Finalizado"}	Otra cadena no presente en la lista en su totalidad.
			null
responsable	String	Longitud de la cadena <= 100	Longitud de la cadena > 100
			null
incidencias	ArrayList<Incidencia>	Lista de incidencias correctas, inicializadas previamente	
		null	
ordenesTrabajo	ArrayList<ordenesTrabajo>	Lista de órdenes de trabajo previamente inicializadas	
		null	
fechaInicio	Date	Fecha de inicio posterior al día de hoy, inclusive	Fecha de inicio anterior al día de hoy
			Fecha de inicio con mal formato (28//2020)
			null

Introducir argumentos correctos	CP-0031
Descripción:	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir los argumentos del método con valores correctos y comprobar que el objeto devuelto se corresponde con estos.
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar
Especificación de entradas: Proceso inicializar = { identificador = 0 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = 15.0 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 }
Especificación de salidas esperadas: La salida esperada será devolver un objeto de la clase Proceso construido con los argumentos introducidos.
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos:

Introducir argumentos null	CP-0032
Descripción: Introducir los argumentos del método con valores null y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso inicializar= { identificador = null nombreProceso = null descripcion = null coste = null estimado = null estado = “Pendiente” responsable = null servicio = null incidencias = null ordenesTrabajo = null fechaInicio = null }	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos:	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir identificador erróneo	CP-0033
Descripción: Introducir los argumentos del método con valores correctos a excepción del identificador y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso inicializar= { identificador = -75 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = 15.0 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 }	
Especificación de salidas esperadas: La salida esperada será devolver un objeto de tipo null indicando que ha sido imposible inicializar el objeto.	
Necesidades del entorno La salida esperada será una excepción con el código de error “Campos no válidos”	
Prerrequisitos:	

Introducir nombreProceso erróneo	CP-0034
Descripción: Introducir los argumentos del método con valores correctos a excepción del nombreProceso y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso inicializar= { identificador = 5 nombreProceso = “” descripcion = “proceso primero” coste = 15.0 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 }	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.

Prerrequisitos:

Introducir coste erróneo	CP-0035
Descripción: Introducir los argumentos del método con valores correctos a excepción del coste y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso inicializar= { identificador = 5 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = -55.5 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 }	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos:	

Introducir estado no existente	CP-0036
Descripción: Introducir los argumentos del método con valores correctos a excepción del estado y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso inicializar= { identificador = 5 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = 50.0 estimado = 0 estado = “Error” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 }	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos:

Introducir incidencias null	CP-0037
Descripción: Introducir los argumentos del método con valores correctos a excepción de las incidencias y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso inicializar= { identificador = 5 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = 50.0 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 }	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos:	

Introducir fechaInicio pasada	CP-0038
Descripción: Introducir los argumentos del método con valores correctos a excepción de la fechaInicio y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso inicializar= { identificador = 5 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = 50.0 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = null }	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

<pre>ordenesTrabajo = ArrayList<OrdenTrabajo> incidencia = { orden1, orden2 } fechaInicio = 22/06/2019 }</pre>
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos:

Introducir fechaInicio errónea	CP-0039
Descripción: Introducir los argumentos del método con valores correctos a excepción de la fechaInicio y comprobar que el objeto devuelto se corresponde con estos.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: <pre>Proceso inicializar= { identificador = 5 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = 50.0 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = null ordenesTrabajo = ArrayList<OrdenTrabajo> incidencia = { orden1, orden2 } fechaInicio = 22//2 }</pre>	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos:	

PRU-0007 Prueba sobre Crear proceso

Métodos involucrados
public Proceso crear (Proceso proceso);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
proceso	Proceso	Proceso con id inexistente en el sistema y previamente inicializado	Proceso con id existente en el sistema
			null

Introducir proceso null	CP-0040
Descripción: Introducir como argumento de la función un objeto null	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Elementos software a probar: Probaremos el método “crear”, correspondiente a la funcionalidad Crear Proceso
Especificación de entradas: La entrada del método será un valor null
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos:

Introducir proceso correcto	CP-0041
Descripción: Introducir como argumento de la función un objeto de la clase Proceso inicializado previamente	
Elementos software a probar: Probaremos el método “crear”, correspondiente a la funcionalidad Crear Proceso	
Especificación de entradas: La entrada del método será un proceso de la clase Proceso correctamente inicializado.	
Especificación de salidas esperadas: La salida esperada será devolver un objeto Proceso similar al de la entrada, indicando la correcta ejecución del método.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto	

Introducir identificador repetido	CP-0042
Descripción: Llamar dos veces al método, la primera con todos el argumento correcto, la segunda, con el mismo identificador que la anterior.	
Elementos software a probar: Probaremos el método “inicializar”, correspondiente a la funcionalidad Inicializar	
Especificación de entradas: Proceso crear = { identificador = 5 nombreProceso = “proceso-1” descripcion = “proceso primero” coste = 15.0 estimado = 0 estado = “Pendiente” responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 } Proceso crear2 = { identificador = 5 nombreProceso = “proceso-2” descripcion = “proceso segundo”	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

coste = 15.0 estimado = 0 estado = "Pendiente" responsable = "Paco Meralgo" servicio = "Asfalto" incidencias = ArrayList<Incidencia> incidencia = { incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> ordenesTrabajo = { orden1, orden2} fechaInicio = 22/06/2021 }
Especificación de salidas esperadas: La salida esperada será un objeto Proceso creado correctamente en base a los atributos pasados en la primera llamada y una excepción con el código de error "Conflicto" en la segunda llamada al método.
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos:

PRU-0008 Prueba sobre Actualizar proceso

Métodos involucrados
public Proceso actualizar (Proceso proceso);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
proceso	Proceso	Proceso con id existente en el sistema y previamente inicializado	Proceso con id inexistente en el sistema
			null

Introducir proceso null	CP-0043
Descripción: Introducir como argumento de la función un objeto null.	
Elementos software a probar: Probaremos el método "actualizar", correspondiente a la funcionalidad Actualizar Proceso	
Especificación de entradas: La entrada del método será un objeto null.	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error "Campos no válidos"	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos:	

Introducir proceso correcto	CP-0044
Descripción: Introducir como argumento un objeto Proceso generado correctamente.	
Elementos software a probar: Probaremos el método "actualizar", correspondiente a la funcionalidad Actualizar Proceso	
Especificación de entradas: Las entrada del método será un objeto Proceso, previamente generado de forma correcta.	
Especificación de salidas esperadas:	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

La salida será un objeto de la clase Proceso, el cual deberá mostrar las actualizaciones realizadas sobre el.
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos:

PRU-0009 Prueba sobre Buscar proceso

Métodos involucrados
public ArrayList<Proceso> buscar (Proceso filtro);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
proceso	Proceso	Proceso con los campos que se desean buscar	
		null	

Introducir proceso null	CP-0045
Descripción: Introducir como argumento de la función un objeto null.	
Elementos software a probar: Probaremos el método “buscar”, correspondiente a la funcionalidad Buscar Proceso	
Especificación de entradas: La entrada del método será un objeto null.	
Especificación de salidas esperadas: La salida esperada será una lista con todos los procesos existentes.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos:	

Introducir proceso correcto	CP-0046
Descripción: Introducir como argumento un objeto Proceso generado correctamente.	
Elementos software a probar: Probaremos el método “buscar”, correspondiente a la funcionalidad Buscar Proceso	
Especificación de entradas: La entrada del método será un objeto Proceso, previamente generado de forma correcta.	
Especificación de salidas esperadas: La salida será un ArrayList<Proceso> que contendrá ninguno o varios procesos que se corresponden con los filtros proporcionados.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto	

PRU-0010 Prueba sobre Asignar proceso/OT

Métodos involucrados
public Proceso asignarOrdentrabajo (Proceso proceso, ArrayList<OrdenTrabajo> ordenesTrabajo);

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
proceso	Proceso	Proceso con id existente en el sistema y previamente inicializado	Proceso con id inexistente en el sistema
			null
ordenesTrabajo	ArrayList <ordenTrabajo>	Lista de órdenes de trabajo existentes en el sistema	Lista de órdenes de trabajo inexistentes en el sistema
			Lista de órdenes de trabajo ya asociadas a otro proceso
			null

Introducir argumentos correctos	CP-0047
Descripción: Introducir los argumentos del método de forma correcta, siendo el Proceso previamente creado correctamente, así como una lista de órdenes de trabajo correctamente creadas.	
Elementos software a probar: Probaremos el método “asignarOrdenTrabajo”, correspondiente a la funcionalidad Asignar proceso/OT.	
Especificación de entradas: Las entradas para los argumentos proceso y ordenesTrabajo serán un proceso creado con anterioridad correctamente y una lista de órdenes de trabajo también generadas correctamente.	
Especificación de salidas esperadas: La salida esperada será el proceso pasado como argumento actualizado con sus órdenes de trabajo asignadas correctamente a este. Esta información será almacenada en el sistema.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto, varios objetos de la clase OrdenTrabajo previamente inicializados y correctos.	

Introducir proceso null	CP-0048
Descripción: Introducir el argumento de ordenesTrabajo de forma correcta, mientras que el proceso establecer el valor null como argumento.	
Elementos software a probar: Probaremos el método “actasignarOrdenTrabajo”, correspondiente a la funcionalidad Asignar proceso/OT.	
Especificación de entradas: Las entradas para el argumento proceso será null mientras que para las ordenes de trabajo se pasará una lista de ordenes de trabajo generadas de forma correcta con anterioridad.	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: varios objetos de la clase OrdenTrabajo previamente inicializados y correctos.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir órdenes de trabajo vacías	CP-0049
Descripción: Introducir el argumento de ordenes de trabajo con una lista vacía, mientras que para el argumento proceso establecer un valor correcto.	
Elementos software a probar: Probaremos el método “actasignarOrdenTrabajo”, correspondiente a la funcionalidad Asignar proceso/OT.	
Especificación de entradas: Las entradas para los argumentos serán, un valor correcto para proceso y una lista de ordenes de trabajo vacía.	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto.	

Introducir órdenes de trabajo null	CP-0050
Descripción: Introducir el argumento de ordenes de trabajo con un objeto null, mientras que para el argumento proceso establecer un valor correcto.	
Elementos software a probar: Probaremos el método “actasignarOrdenTrabajo”, correspondiente a la funcionalidad Asignar proceso/OT.	
Especificación de entradas: Las entradas para los argumentos serán, un valor correcto para proceso y un valor null como lista de ordenes de trabajo.	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto.	

Introducir órdenes de trabajo repetidas	CP-0051
Descripción: Llamar dos veces al método con la misma lista y el mismo proceso, intentando asignarle dos veces las órdenes de trabajo.	
Elementos software a probar: Probaremos el método “actasignarOrdenTrabajo”, correspondiente a la funcionalidad Asignar proceso/OT.	
Especificación de entradas: Las entradas para los argumentos serán, un valor correcto para proceso y una lista de ordenes de trabajo correcta, en la segunda llamada al método, serán los mismos argumentos.	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Conflicto”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto, varios objetos de la clase OrdenTrabajo previamente inicializados y correctos.

PRU-0011 Prueba sobre Asignar proceso/incidencia

Métodos involucrados
public Proceso asignarIncidencia (Proceso proceso, ArrayList<Incidencia> incidencias);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
proceso	Proceso	Proceso con id existente en el sistema y previamente inicializado	Proceso con id inexistente en el sistema
			null
incidencias	ArrayList<Incidencia>	Lista de incidencias existentes en el sistema	Lista de incidencias inexistentes en el sistema
			Lista de incidencias ya asociadas a otro proceso
			null

Introducir argumentos correctos	CP-0052
Descripción: Introducir los argumentos del método de forma correcta, siendo el Proceso previamente creado correctamente, así como una lista de incidencias correctamente creadas.	
Elementos software a probar: Probaremos el método “asignarIncidencia”, correspondiente a la funcionalidad Asignar proceso/incidencia.	
Especificación de entradas: Las entradas para los argumentos proceso e incidencias serán un proceso creado con anterioridad correctamente y una lista de incidencias también generadas correctamente.	
Especificación de salidas esperadas: La salida esperada será el proceso pasado como argumento actualizado con sus incidencias asignadas correctamente a este. Esta información será almacenada en el sistema.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto, varios objetos de la clase Incidencia previamente inicializados y correctos.	

Introducir proceso null	CP-0053
Descripción: Introducir el argumento de incidencias de forma correcta, mientras que el proceso establecer el valor null como argumento.	
Elementos software a probar: Probaremos el método “asignarIncidencia”, correspondiente a la funcionalidad Asignar proceso/incidencia.	
Especificación de entradas: Las entradas para el argumento proceso será null mientras que para las incidencias se pasará una lista de incidencias generadas de forma correcta con anterioridad.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: varios objetos de la clase Incidencia previamente inicializados y correctos.

Introducir incidencias vacías	CP-0054
Descripción: Introducir el argumento de incidencias con una lista vacía, mientras que para el argumento proceso establecer un valor correcto.	
Elementos software a probar: Probaremos el método “asignarIncidencia”, correspondiente a la funcionalidad Asignar proceso/incidencia.	
Especificación de entradas: Las entradas para los argumentos serán, un valor correcto para proceso y una lista de incidencias vacía.	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto.	

Introducir incidencias null	CP-0055
Descripción: Introducir el argumento de incidencias con un objeto null, mientras que para el argumento proceso establecer un valor correcto.	
Elementos software a probar: Probaremos el método “asignarIncidencia”, correspondiente a la funcionalidad Asignar proceso/incidencia.	
Especificación de entradas: Las entradas para los argumentos serán, un valor correcto para proceso y un valor null como lista de incidencias.	
Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Campos no válidos”	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto.	

Introducir incidencias repetidas	CP-0056
Descripción: Llamar dos veces al método con la misma lista y el mismo proceso, intentando asignarle dos veces las incidencias.	
Elementos software a probar: Probaremos el método “asignarIncidencia”, correspondiente a la funcionalidad Asignar proceso/incidencia.	
Especificación de entradas: Las entradas para los argumentos serán, un valor correcto para proceso y una lista de incidencias correcta, en la segunda llamada al método, serán los mismos argumentos.	

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Especificación de salidas esperadas: La salida esperada será una excepción con el código de error “Conflicto”
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: un objeto de la clase Proceso previamente inicializado y correcto, varios objetos de la clase Incidencia previamente inicializados y correctos.

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

- *Subsistema de gestión de OTs*

PRU-0012 Inicializar Orden de Trabajo

Métodos involucrados
public OrdenTrabajo inicializar (Integer identificador, String descripcion, ArrayList<String> material, ArrayList<Presupuesto> presupuestos, Double coste, String responsable, Integer personal, Date fechaInicio, Integer duracion, String estado, Proceso proceso);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
identificador	Integer	Número entero positivo	Número entero negativo
		null	
descripcion	String	Longitud de la cadena <= 500	Longitud de la cadena > 500
		null	
material	ArrayList<String>	Lista de materiales, cada uno de longitud <= 20	Lista de materiales con longitud > 20
		null	
presupuestos	ArrayList<Presupuesto>	Lista de presupuestos	
		null	
coste	Double	Decimal positivo	Decimal negativo
		null	
responsable	String	String alfabético con longitud <= 100 caracteres	String con caracteres no alfabéticos
		null	String con más de 100 caracteres
personal	Integer	Número entero positivo	Número entero negativo
		null	
fechaInicio	Date	Fecha	Fecha de inicio anterior al día de hoy
		null	Fecha de inicio con mal formato (28//2020)
duracion	Integer	Número entero positivo	Número entero negativo
		null	
estado	String	Presente en la lista: {"Pendiente de asignación", "En curso", "Finalizado"}	String no presente en la lista: {"Pendiente de asignación", "En curso", "Finalizado"}
		null	
proceso	Proceso	Proceso inicializado	
		null	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir todos los argumentos correctos	CP-0057
Descripción: Introduciremos todos los argumentos correctos del método.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será devolver la clase Orden Trabajo con los valores: OrdenTrabajo nuevaOT = { Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = {presupuesto1, presupuesto2} Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.	

Introducir todos los argumentos nulos	CP-0058
Descripción: Introduciremos todos los argumentos nulos en el método y comprobaremos si devuelve un objeto OT nulo.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: null Descripción: null Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: null Proceso: null	

ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

<p>Especificación de salidas esperadas: La salida esperada será lanzar una CustomException con código 1.</p>
<p>Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.</p>
<p>Prerrequisitos: Ninguno.</p>

Introducir identificador negativo	CP-0059
Descripción: Introduciremos todos los argumentos correctos del método excepto el identificador, y este será un número negativo.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: -12345 Descripción: “Se cambiarán las aceras de la Avenida Rosalía de Castro” Material: {“Adoquines”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 2000 Responsable: “Laura” Personal: 50 FechaInicio: 21/12/2021 Duración: 100 Estado: “Pendiente de asignación” Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será lanzar una CustomException con código 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.	

Introducir descripción con más de 500 caracteres	CP-0060
Descripción: Introduciremos todos los argumentos correctos del método, pero la descripción será de más de 500 caracteres.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: 456789 Descripción: “aa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...” (500 caracteres) Material: {“Adoquines”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 2000 Responsable: “Laura” Personal: 50 FechaInicio: 21/12/2021 Duración: 100 Estado: “Pendiente de asignación” <u>Proceso: proceso1</u>	
Especificación de salidas esperadas: La salida esperada será lanzar una CustomException con código 1.	
Necesidades del entorno	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

Introducir responsable con caracteres no alfabéticos	CP-0061
Descripción: Introduciremos todos los argumentos correctos del método pero responsable tendrá caracteres no alfabéticos.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: “Pepe1” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será lanzar una CustomException con código 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.	

Introducir personal negativo	CP-0062
Descripción: Introduciremos todos los argumentos correctos del método pero personal será negativo.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: “Pepe” Personal: -10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será lanzar una CustomException con código 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

Introducir fecha de inicio con formato incorrecto	CP-0063
Descripción: Introduciremos todos los argumentos correctos del método pero la fecha tendrá formato incorrecto.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31//2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será devolver la clase Orden Trabajo con los valores: null	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.	

Introducir duración negativa	CP-0064
Descripción: Introduciremos todos los argumentos correctos del método pero con la duración negativa.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: -80 Estado: "Pendiente de asignación" Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será lanzar una CustomException con código 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.	

Introducir estado "Pendiente de Asignación"	CP-0065
---	---------

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: <i>PDP_2C_v3.doc</i>	

Descripción: Introduciremos todos los argumentos correctos del método con estado “Pendiente de Asignación”.
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar
Especificación de entradas: Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: “Pepe” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1
Especificación de salidas esperadas: La salida esperada será devolver la clase Orden Trabajo con los valores: OrdenTrabajo nueva = {Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: “Pepe” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1 }
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

Introducir estado con cadena no válida	CP-0066
Descripción: Introduciremos todos los argumentos correctos del método con estado “Esperando”.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: “Pepe” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Esperando” Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será lanzar una CustomException con código 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

PRU-0013 Prueba sobre Inicializar presupuestos.

Métodos involucrados
public Presupuesto inicializar (Integer identificador, String empresa, Double presupuesto, Date fechaInicio, Integer duracion, ArrayList<String> material, Integer personal, Integer idOT));

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
identificador	Integer	Número entero positivo	Número entero negativo
			null
empresa	String	Longitud de la cadena <= 100	Longitud de la cadena > 100
			null
presupuesto	Double	Número decimal positivo	Número decimal negativo
			null
fechaInicio	Date	Fecha válida	Fecha de inicio anterior al día de hoy
			Fecha con formato incorrecto
			null
duracion	Integer	String <= 100 caracteres	String con caracteres no numéricos
			null
material	ArrayList<String>	Número entero positivo	Entero negativo
			null
personal	Integer	Número entero positivo	Número negativo
			null
idOT	Integer	Entero positivo que identifica una orden de trabajo registrada	Entero negativo
		null	Identificador de OT inexistente

Introducir todos los argumentos correctos	CP-0067
Descripción: Introduciremos todos los argumentos correctos del método.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 123 Empresa: "Eléctricas Pepe" Presupuesto: 5000 FechaInicio: 31/12/2021 Duración: 80 Material: {"Metal", "Bombillas"} Personal: 10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será devolver la clase Presupuesto con los valores: Presupuesto nuevoPres = {Identificador: 123 Empresa: "Eléctricas Pepe" Presupuesto: 5000	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

FechaInicio: 31/12/2021 Duración: 80 Material: {"Metal", "Bombillas"} Personal: 10 IdOT: 12345}
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir todos los argumentos nulos	CP-0068
Descripción: Introduciremos todos los argumentos nulos en el método.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: null Empresa: null Presupuesto: null FechaInicio: null Duración: null Material: null Personal: null IdOT: null	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir identificador de OT inexistente	CP-0069
Descripción: Introduciremos todos los argumentos correctos del método excepto el identificador de OT, y este debe coincidir con el de una OT que no exista en el sistema.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: 123 Descripción: "Se cambiarán las aceras de la Avenida Rosalía de Castro" Material: {"Adoquines"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 2000 Responsable: "Laura" Personal: 50 FechaInicio: 21/12/2021 Duración: 100 Estado: "Pendiente de asignación" Proceso: proceso1 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 4 (No encontrado).	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado. Habrá una OT con ese identificador ya en el sistema.

Introducir identificador repetido	CP-0070
Descripción: Introduciremos todos los argumentos correctos en el método con el identificador repetido de otro Presupuesto existente en el sistema.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 123 Empresa: “Eléctricas Pepe” Presupuesto: 5000 FechaInicio: 31/12/2021 Duración: 80 Material: {“Metal”, “Bombillas”} Personal: 10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 2.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Habrá un Presupuesto con ese identificador ya en el sistema.	

Introducir identificador negativo	CP-0071
Descripción: Introduciremos todos los argumentos correctos en el método con el identificador negativo.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: -123 Empresa: “Eléctricas Pepe” Presupuesto: 5000 FechaInicio: 31/12/2021 Duración: 80 Material: {“Metal”, “Bombillas”} Personal: 10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir empresa con más de 100 caracteres	CP-0072
Descripción: Introduciremos todos los argumentos correctos en el método pero la empresa con más de 100 caracteres	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: -123 Empresa: “Eléctricas Pepeee” Presupuesto: 5000	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

FechaInicio: 31/12/2021 Duración: 80 Material: {"Metal", "Bombillas"} Personal: 10 IdOT: 12345
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir presupuesto negativo	CP-0073
Descripción: Introduciremos todos los argumentos correctos en el método pero el presupuesto negativo.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: -123 Empresa: "Eléctricas Pepe" Presupuesto: -5000 FechaInicio: 31/12/2021 Duración: 80 Material: {"Metal", "Bombillas"} Personal: 10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir fecha de inicio con formato incorrecto	CP-0074
Descripción: Introduciremos todos los argumentos correctos en el método pero la fecha tiene formato incorrecto.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 123 Empresa: "Eléctricas Pepe" Presupuesto: 5000 FechaInicio: 31//2021 Duración: 80 Material: {"Metal", "Bombillas"} Personal: 10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción en la creación de la fecha.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir fecha de inicio anterior a la actual	CP-0075
---	---------

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Descripción: Introduciremos todos los argumentos correctos en el método pero la fecha es anterior a la actual.
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .
Especificación de entradas: Identificador: 123 Empresa: “Eléctricas Pepe” Presupuesto: 5000 FechaInicio: 31/12/2020 Duración: 80 Material: {“Metal”, “Bombillas”} Personal: 10 IdOT: 12345
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir duración negativa	CP-0076
Descripción: Introduciremos todos los argumentos correctos en el método pero la duración es negativa.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 123 Empresa: “Eléctricas Pepe” Presupuesto: 5000 FechaInicio: 31/12/2021 Duración: -80 Material: {“Metal”, “Bombillas”} Personal: 10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir ArrayList material vacío	CP-0077
Descripción: Introduciremos todos los argumentos correctos en el método pero material es un ArrayList vacío.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 123 Empresa: “Eléctricas Pepe” Presupuesto: 5000 FechaInicio: 31/12/2021 Duración: 80 Material: new ArrayList<>() Personal: 10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será devolver la clase Presupuesto con los valores: Presupuesto nuevoPres = {Identificador: 12345 Empresa: “Eléctricas Pepe” Presupuesto: 5000	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

FechaInicio: 31/12/2021 Duración: 80 Material: new ArrayList<>() Personal: 10 IdOT: 12345}
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir personal negativo	CP-0078
Descripción: Introduciremos todos los argumentos correctos en el método pero personal es negativo.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar .	
Especificación de entradas: Identificador: 123 Empresa: “Eléctricas Pepe” Presupuesto: 5000 FechaInicio: 31/12/2021 Duración: 80 Material: {“Metal”, “Bombillas”} Personal: -10 IdOT: 12345	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

PRU-0014 Prueba sobre Crear Orden de Trabajo.

Métodos involucrados
public OrdenTrabajo crear (OrdenTrabajo ordenTrabajo);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
ordenTrabajo	OrdenTrabajo	Orden de Trabajo con campos válidos	null

Introducir una Orden de Trabajo nula	CP-0079
Descripción: Introduciremos una OT nula	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0014 Crear Orden de Trabajo .	
Especificación de entradas: OrdenTrabajo ot = null	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguna	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir los argumentos obligatorios cubiertos	CP-0080
Descripción: Introduciremos todos los argumentos obligatorios del método correctamente, y el resto a null.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0014 Crear Orden de Trabajo.	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: "Pendiente de asignación" Proceso: null}	
Especificación de salidas esperadas: La salida esperada será devolver la clase OrdenTrabajo con los valores: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: "Pendiente de asignación" Proceso: null}	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguno	

Introducir todos los argumentos correctos	CP-0081
Descripción: Introduciremos todos los argumentos correctos del método.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0014 Crear Orden de Trabajo.	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: null Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1}	
Especificación de salidas esperadas: La salida esperada será devolver la clase OrdenTrabajo con los valores: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = new ArrayList<>() Coste: 1000 Responsable: "Pepe"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

Introducir OT con identificador ya existente	CP-0082
Descripción: Introduciremos todos los argumentos correctos del método pero el identificador corresponde a una OT ya existente.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0014 Crear Orden de Trabajo.	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: null Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 2.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado. Además, habrá una OT con ese identificador ya en el sistema.	

Introducir proceso	CP-0083
Descripción: Introduciremos todos los argumentos correctos del método con un proceso (no se puede vincular a proceso hasta después de crearla.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0014 Crear Orden de Trabajo.	
Especificación de entradas: Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: null Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso404	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 3.	
Necesidades del entorno	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.

Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

Introducir fecha de inicio anterior a la actual	CP-0084
Descripción: Introduciremos todos los argumentos correctos del método pero la fecha será anterior a la actual.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de Inicializar	
Especificación de entradas: Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: null Coste: 1000 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2020 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.	

PRU-0015 Prueba sobre Asignar Empresa.

Métodos involucrados
public OrdenTrabajo asignarEmpresa (OrdenTrabajo ordentrabajo, Presupuesto presupuesto);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
ordenTrabajo	OrdenTrabajo	Orden de Trabajo con campos válidos	null
presupuesto	Presupuesto	Presupuesto con campos válidos	null

Introducir OT y Presupuesto con campos obligatorios	CP-0085
Descripción: Introduciremos todos los argumentos obligatorios correctos del método.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0016 Asignar Empresa/Responsable .	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: "Pendiente de asignación" Proceso: null} Presupuesto nuevo = {Identificador: 12347 Empresa: "Eléctricas Pepe" Presupuesto: 10000 FechaInicio: 29/5/2021 Duración: 80 Material: {"Metal", "Bombillas"} Personal: 50 IdOT: 12345}
Especificación de salidas esperadas: La salida esperada será devolver la clase OrdenTrabajo con los valores: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList pres = [{Identificador: 12347 Empresa: "Eléctricas Pepe" Presupuesto: 10000 FechaInicio: 29/5/2021 Duración: 80 Material: {"Metal", "Bombillas"} Personal: 80 IdOT: 12345}] Coste: null Responsable: "Eléctricas Pepe" Personal: 80 FechaInicio: 29/5/2021 Duración: 80 Estado: "Asignado" Proceso: proceso1}
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguno

Introducir OT y Presupuesto nulas	CP-0086
Descripción: Introduciremos todos los argumentos nulos.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0016 Asignar Empresa/Responsable .	
Especificación de entradas: OrdenTrabajo ot = null Presupuesto nuevo = null	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 1.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Ninguna	

Introducir un presupuesto que no estaba almacenado	CP-0087
--	---------

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Descripción: Introduciremos todos los argumentos correctos pero el identificador de Presupuesto no corresponde a ninguno registrado.
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0016 Asignar Empresa/Responsable.
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: "Pendiente de asignación" Proceso: null} Presupuesto nuevo = {Identificador: 12347 Empresa: "Eléctricas Pepe" Presupuesto: 10000 FechaInicio: 29/07/2021 Duración: 140 Material: {"Metal", "Bombillas"} Personal: 50 IdOT: 12345}
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 4 (presupuesto no registrado).
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: El identificador de Presupuesto no corresponde a ninguno existente.

Introducir un presupuesto y OT sin que haya 3 almacenados	CP-0088
Descripción: Introduciremos todos los argumentos obligatorios correctos pero no hay 3 presupuestos almacenados para esa OT	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0016 Asignar Empresa/Responsable.	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: "Pendiente de asignación" Proceso: null} Presupuesto nuevo = {Identificador: 12347 Empresa: "Eléctricas Pepe" Presupuesto: 10000 FechaInicio: 29/07/2021 Duración: 140 Material: {"Metal", "Bombillas"} Personal: 50 IdOT: 12345}	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 3.
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: No hay 3 presupuestos almacenados para esa OT.

Introducir una OT que no exista en el sistema	CP-0089
Descripción: Introduciremos todos los argumentos obligatorios correctos pero el identificador de la OT no se corresponde con ninguna del sistema.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0016 Asignar Empresa/Responsable.	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: "Pendiente de asignación" Proceso: null} Presupuesto nuevo = {Identificador: 12347 Empresa: "Eléctricas Pepe" Presupuesto: null FechaInicio: null Duración: null Material: {"Metal", "Bombillas"} Personal: null IdOT: 12345}	
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 4.	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: el identificador de la OT no se corresponde con ninguna del sistema.	

PRU-0016 Prueba sobre Gestionar Recursos.

Métodos involucrados
public OrdenTrabajo gestionarRecursos (OrdenTrabajo ordenTrabajo);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
ordenTrabajo	OrdenTrabajo	Orden de trabajo con campos válidos	null

Introducir una OT que no exista en el sistema	CP-0090
---	---------

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Descripción: Introduciremos todos los argumentos obligatorios correctos pero el identificador de la OT no se corresponde con ninguna del sistema.
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0017 Gestionar Recursos.
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: "Pendiente de asignación" Proceso: proceso1 }
Especificación de salidas esperadas: La salida esperada será que se lance una excepción CustomException con código 4.
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: el identificador de la OT no se corresponde con ninguna del sistema.

Introducir una OT que exista, con campos modificados	CP-0091
Descripción: Introduciremos todos los argumentos correctos con campos modificados con respecto a la original (se ha modificado campo materiales)	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0017 Gestionar Recursos.	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }	
Especificación de salidas esperadas: La salida esperada será devolver la clase OrdenTrabajo con los valores: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }	
Siendo la OrdenTrabajo original la siguiente: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1 }	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

Introducir una OT que exista, sin campos modificados	CP-0092
Descripción: Introduciremos todos los argumentos correctos con campos iguales con respecto a la original.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0017 Gestionar Recursos .	
Especificación de entradas: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }	
Especificación de salidas esperadas: La salida esperada será devolver la clase OrdenTrabajo con los valores: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }	
Siendo la OrdenTrabajo original la siguiente: OrdenTrabajo ot = {Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.

PRU-0017 Prueba sobre Obtener OTs / Buscar.

Métodos involucrados
public OrdenTrabajo buscar (OrdenTrabajo filtro);

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
filtro	OrdenTrabajo	OrdenTrabajo con campos cubiertos	
		OrdenTrabajo nula para buscar sin filtros	

Introducir una OT null para obtener todas las OT	CP-0093
Descripción: Introduciremos como argumento null para obtener todas las OT (si es que hay).	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0018 Obtener OTs .	
Especificación de entradas: null	
Especificación de salidas esperadas: La salida esperada será un ArrayList de la clase OrdenTrabajo con los valores, todas las OTs disponibles: ArrayList<OrdenTrabajo> ot = [{Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }, {Identificador: 12346 Descripción: "Se cambiarán las aceras de la Avenida Rosalía de Castro" Material: {"Adoquines"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto3 } Coste: 5000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 100 Estado: "Pendiente de asignación" Proceso: proceso1 }]	
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Se necesita que haya al menos un presupuesto creado y al menos un Proceso creado.	

Introducir una OT con todos sus campos a null	CP-0094
Descripción: Introduciremos como argumento null una OT con todos los campos a null.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0018 Obtener OTs.
Especificación de entradas: ot = [{Identificador: 12345 Descripción: null Material: null Presupuestos: null Coste: null Responsable: null Personal: null FechaInicio: null Duración: null Estado: null Proceso: null}]
Especificación de salidas esperadas: La salida esperada será un ArrayList de la clase OrdenTrabajo con los valores, todas las OTs disponibles: ArrayList<OrdenTrabajo> ot = [{Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1 }, {Identificador: 12346 Descripción: "Se cambiarán las aceras de la Avenida Rosalía de Castro" Material: {"Adoquines"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto3 } Coste: 5000 Responsable: null Personal: 10 FechaInicio: 31/12/2021 Duración: 100 Estado: "Pendiente de asignación" Proceso: proceso1 }]
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Ninguna.

Introducir una OT con algunos campos cubiertos	CP-0095
Descripción: Introduciremos como argumento null una OT con algunos campos cubiertos para devolver la búsqueda filtrada.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad de UC-0018 Obtener OTs.	
Especificación de entradas: OrdenTrabajo ot = {Identificador: null Descripción: null Material: null Presupuestos: null Coste: 1000 Responsable: null Personal: null FechaInicio: null Duración: 80 Estado: null	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Proceso: proceso1 }
Especificación de salidas esperadas: La salida esperada será devolver la clase OrdenTrabajo con los valores: ot = {Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 1000 Responsable: “Eléctricas Pepe” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1 }
Necesidades del entorno: El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Tiene que haber un proceso creado.

- *Subsistema de análisis y estadísticas*

PRU-0018 Prueba sobre Obtener estadísticas de incidencias.

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
Rango	String	Rango de fechas con formato correcto “1/1/2000-1/1/3000”	Rango de fechas con formato incorrecto “1/1/2000-1/1-3000” o “1-1-2000/1-1-3000”
			null
Filtro	Incidencia	Incidencia con campos adecuados para filtrar	
		Incidencia nula para no filtrar nada	
Distribucion	String	Valor “día”	Otro valor diferente
		Valor “semana”	null

Métodos involucrados
public Incidencia obtenerEstadisticasIncidencias (String rango, Incidencia filtro, String distribución);

Introducir todos los argumentos correctos	CP-0096
Descripción: Introduciremos todos los argumentos correctos del método y el filtro a null para que se cuenten todos los elementos.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0019 Obtener estadísticas de incidencias .	
Especificación de entradas: rango=”1/1/2000-1/1/3000” filtro = null	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

distribucion = "dia"
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/dias.
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Que exista una incidencia en el sistema tal que: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Asfalto" Proceso.coste = 20 fechaInicio = 22/04/2021

Introducir un rango de fechas con formato incorrecto	CP-0097
Descripción: Introduciremos un rango de fechas en formato incorrecto en la función.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0019 Obtener estadísticas de incidencias.	
Especificación de entradas: rango="1/1/2000-1/1/-3000" filtro = null distribucion = "dia"	
Especificación de salidas esperadas: Será una excepción con código de error 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista una incidencia en el sistema tal que: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

tipoIncidencia = "Asfalto"
 Proceso.coste = 20
 fechaInicio = 22/04/2021

Introducir una distribución semana	CP-0098
Descripción: Introduciremos el valor "semana" para el parámetro distribución.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0019 Obtener estadísticas de incidencias.	
Especificación de entradas: rango="1/1/2000-1/1/3000" filtro = null distribucion = "semana"	
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/semanas.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista una incidencia en el sistema tal que: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Asfalto" Proceso.coste = 20 fechaInicio = 22/04/2021	

Introducir una distribución incorrecta	CP-0099
Descripción: Introduciremos un valor incorrecto para el parámetro distribución.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0019 Obtener estadísticas de incidencias.	
Especificación de entradas: rango="1/1/2000-1/1/3000"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

filtro = null distribucion = "pepe"
Especificación de salidas esperadas: Será una excepción con código de error 1.
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Que exista una incidencia en el sistema tal que: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Asfalto" Proceso.coste = 20 fechaInicio = 22/04/2021

Introducir un filtro correcto	CP-0100
Descripción: Introduciremos una Incidencia con valores correctos en sus campos para filtrar.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0019 Obtener estadísticas de incidencias .	
Especificación de entradas: rango="1/1/2000-1/1/3000" Filtro.dni=" 53199271H" distribucion = "dia"	
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con las características correctas habiendo aplicado el filtro.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista una incidencia en el sistema tal que: identificador = 0 ciudadano = "ciudadano" DNI = "53199271H" telefono = "34608564255" descripcion = "descripción de prueba" localizacion = "localización de prueba" tipoIncidencia = "Asfalto"	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Proceso.coste = 20
fechaInicio = 22/04/2021

PRU-0019 Prueba sobre Obtener estadísticas de procesos.

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
Rango	String	Rango de fechas con formato correcto "1/1/2000-1/1/3000"	Rango de fechas con formato incorrecto "1/1/2000-1/1/-3000" o "1-1-2000/1-1-3000"
			null
Filtro	Procesos	Proceso con campos adecuados para filtrar	
		Proceso nulo para no filtrar nada	
Distribucion	String	Valor "día"	Otro valor diferente
		Valor "semana"	null

Métodos involucrados
public Incidencia obtenerEstadisticasProcesos (String rango, Proceso filtro, String distribución);

Introducir todos los argumentos correctos	CP-0101
Descripción: Introduciremos todos los argumentos correctos del método y el filtro a null para que se cuenten todos los elementos.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0020 Obtener estadísticas de procesos.	
Especificación de entradas: rango="1/1/2000-1/1/3000" filtro = null distribucion = "día"	
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/días.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista un proceso en el sistema tal que: identificador = 0 descripcion = "proceso-1" coste = 20.0 estimado = 0 responsable = "Paco Meralgo" servicio = "Asfalto" incidencias = ArrayList<Incidencia> incidencia = {incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> incidencia = {orden1, orden2}	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

fechaInicio = 22/06/2021

Introducir un rango de fechas con formato incorrecto	CP-0102
Descripción: Introduciremos un rango de fechas en formato incorrecto en la función.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0020 Obtener estadísticas de procesos.	
Especificación de entradas: rango="1/1/2000-1/1/-3000" filtro = null distribucion = "dia"	
Especificación de salidas esperadas: Será una excepción con código de error 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista un proceso en el sistema tal que: identificador = 0 descripcion = "proceso-1" coste = 20.0 estimado = 0 responsable = "Paco Meralgo" servicio = "Asfalto" incidencias = ArrayList<Incidencia> incidencia = {incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> incidencia = {orden1, orden2} fechaInicio = 22/06/2021	

Introducir una distribución semana	CP-0103
Descripción: Introduciremos el valor "semana" para el parámetro distribución.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0020 Obtener estadísticas de procesos.	
Especificación de entradas: rango="1/1/2000-1/1/3000" filtro = null distribucion = "semana"	
Especificación de salidas esperadas:	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/días.
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Que exista un proceso en el sistema tal que: identificador = 0 descripcion = “proceso-1” coste = 20.0 estimado = 0 responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = {incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> incidencia = {orden1, orden2} fechaInicio = 22/06/2021

Introducir una distribución incorrecta	CP-0104
Descripción: Introduciremos un valor incorrecto para el parámetro distribución.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0020 Obtener estadísticas de procesos.	
Especificación de entradas: rango=”1/1/2000-1/1/3000” filtro = null distribucion = “pepe”	
Especificación de salidas esperadas: Será una excepción con código de error 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista un proceso en el sistema tal que: identificador = 0 descripcion = “proceso-1” coste = 20.0 estimado = 0 responsable = “Paco Meralgo” servicio = “Asfalto” incidencias = ArrayList<Incidencia> incidencia = {incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> incidencia = {orden1, orden2} fechaInicio = 22/06/2021	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir un filtro correcto	CP-0105
Descripción: Introduciremos un Proceso con valores correctos en sus campos para filtrar.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0020 Obtener estadísticas de procesos.	
Especificación de entradas: rango="1/1/2000-1/1/3000" filtro.responsable="Paco Meralgo" distribucion = "dia"	
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/dias.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista un proceso en el sistema tal que: identificador = 0 descripcion = "proceso-1" coste = 20.0 estimado = 0 responsable = "Paco Meralgo" servicio = "Asfalto" incidencias = ArrayList<Incidencia> incidencia = {incidencia1, incidencia2} ordenesTrabajo = ArrayList<OrdenTrabajo> incidencia = {orden1, orden2} fechaInicio = 22/06/2021	

PRU-0020 Prueba sobre Obtener estadísticas de OTs.

Clases de equivalencia			
Entrada	Tipo de dato	Válido	No válido
Rango	String	Rango de fechas con formato correcto "1/1/2000-1/1/3000"	Rango de fechas con formato incorrecto "1/1/2000-1/1/-3000" o "1-1-2000/1-1-3000"
			null
Filtro	OT	OT con campos adecuados para filtrar	
		OT nula para no filtrar nada	
Distribucion	String	Valor "día"	Otro valor diferente
		Valor "semana"	null

Métodos involucrados
public Incidencia obtenerEstadisticasOrdenesTrabajo (String rango, OrdenTrabajo filtro, String distribución);

Introducir todos los argumentos correctos	CP-0106
Descripción: Introduciremos todos los argumentos correctos del método y el filtro a null para que se cuenten todos los elementos.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0021 Obtener estadísticas de OTs.	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Especificación de entradas: rango="1/1/2000-1/1/3000" filtro = null distribucion = "dia"
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/dias.
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Que exista una OTs en el sistema tal que: Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 20 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1

Introducir un rango de fechas con formato incorrecto	CP-0107
Descripción: Introduciremos un rango de fechas en formato incorrecto en la función.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0021 Obtener estadísticas de OTs.	
Especificación de entradas: rango="1/1/2000-1/1/-3000" filtro = null distribucion = "dia"	
Especificación de salidas esperadas: Será una excepción con código de error 1.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista una OTs en el sistema tal que: Identificador: 12345 Descripción: "Se cambiarán las farolas de la Avenida Rosalía de Castro" Material: {"Metal", "Bombillas"} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 20 Responsable: "Pepe" Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: "Pendiente de asignación" Proceso: proceso1	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Introducir una distribución semana	CP-0108
Descripción: Introduciremos el valor “semana” para el parámetro distribución.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0021 Obtener estadísticas de OTs.	
Especificación de entradas: rango=”1/1/2000-1/1/3000” filtro = null distribucion = “semana”	
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/dias.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista una OTs en el sistema tal que: Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 20 Responsable: “Pepe” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1	

Introducir una distribución incorrecta	CP-0109
Descripción: Introduciremos un valor incorrecto para el parámetro distribución.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0021 Obtener estadísticas de OTs.	
Especificación de entradas: rango=”1/1/2000-1/1/3000” filtro = null distribucion = “pepe”	
Especificación de salidas esperadas: Será una excepción con código de error 1.	

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.
Prerrequisitos: Que exista una OTs en el sistema tal que: Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 20 Responsable: “Pepe” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1

Introducir un filtro correcto	CP-0110
Descripción: Introduciremos una OTs con valores correctos en sus campos para filtrar.	
Elementos software a probar: Probaremos el método java asociado a la funcionalidad del UC-0021 Obtener estadísticas de OTs.	
Especificación de entradas: rango=”1/1/2000-1/1/3000” filtro.responsable=”responsableExistenteEnLaLista” distribucion = “dia”	
Especificación de salidas esperadas: La salida esperada será devolver la clase Estadística con los valores numeroTotal=1 y coste=20/dias.	
Necesidades del entorno El código asociado al proyecto en un ordenador con Eclipse, Junit y Mockito. Todo esto a disposición de la persona responsable de la prueba.	
Prerrequisitos: Que exista una OTs en el sistema tal que: Identificador: 12345 Descripción: “Se cambiarán las farolas de la Avenida Rosalía de Castro” Material: {“Metal”, “Bombillas”} Presupuestos: ArrayList<Presupuesto> pres = { presupuesto1, presupuesto2} Coste: 20 Responsable: “Pepe” Personal: 10 FechaInicio: 31/12/2021 Duración: 80 Estado: “Pendiente de asignación” Proceso: proceso1	

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

Pruebas de caja blanca

Para este apartado se realizan las pruebas de caja blanca en 4 métodos del sistema con una complejidad alta. Se elaboran los grafos asociados al código y se aplica el método del camino básico para encontrar el número mínimo de caminos independientes, cabe destacar que en algunos casos se han tenido que modificar los cambios para satisfacer decisiones posteriores debido a restricciones específicas del método. En cuanto a los casos de prueba se aplica el criterio de cobertura de decisión. Mediante este criterio cada uno de los posibles caminos del grafo se ejecuta al menos una vez. Para reducir el número de caminos generados por los bucles se utiliza el concepto de camino de prueba (**test path**), en el cual se atraviese como máximo una vez el interior de cada bucle.

El número de caminos que se deben ejecutar como mínimo para cubrir los posibles caminos del grafo es igual a su complejidad. Para el cálculo de la complejidad se aplica cualquiera de las siguientes formulas.

Nº de aristas = a

Nº de nodos = n

Regiones = r

Complejidad = V(G)

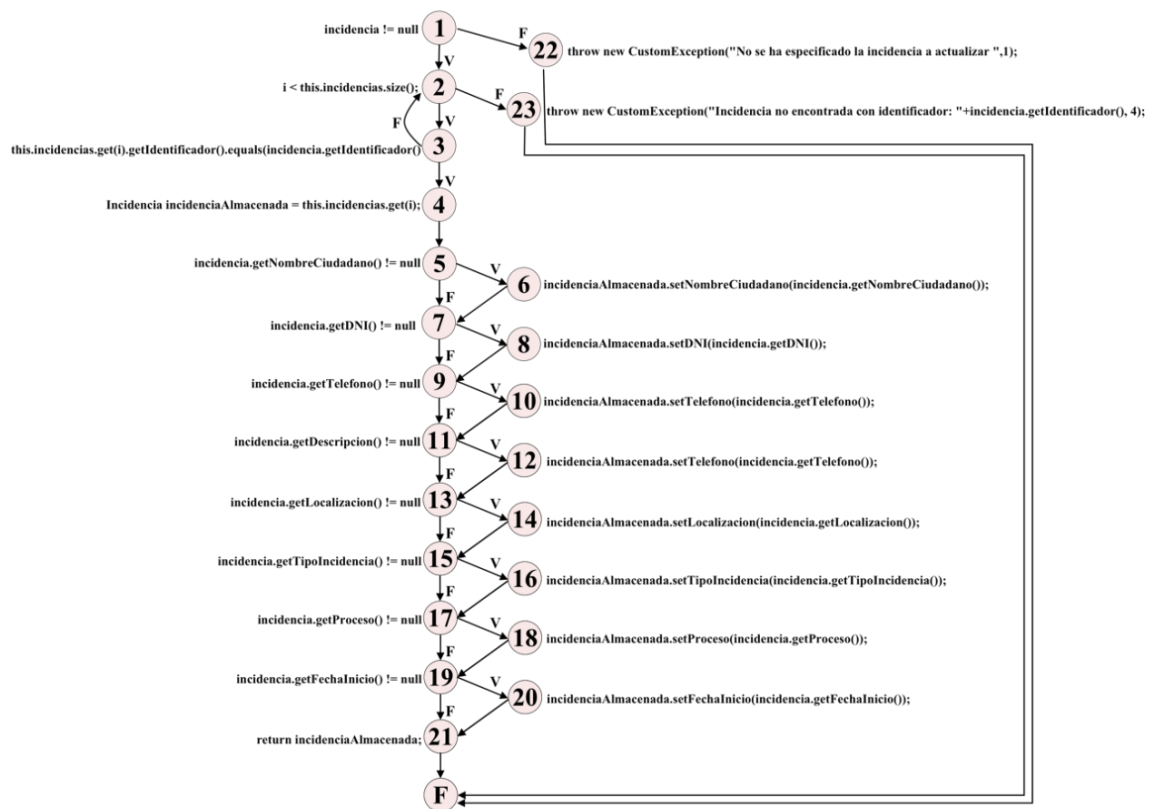
$$V(G) = a - n + 2$$

$$V(G) = r$$

$$V(G) = r + 1 \text{ (si el grafo no es fuertemente conexo)}$$

- ***Actualizar Incidencia***

```
public Incidencia actualizar(Incidencia incidencia) throws CustomException{
    if (incidencia != null) {
        for (int i = 0; i < this.incidentes.size(); i++) {
            if (this.incidentes.get(i).getIdificador().equals(incidencia.getIdificador())) {
                Incidencia incidenciaAlmacenada = this.incidentes.get(i);
                if (incidencia.getNombreCiudadano() != null)
                    incidenciaAlmacenada.setNombreCiudadano(incidencia.getNombreCiudadano());
                if (incidencia.getDNI() != null)
                    incidenciaAlmacenada.setDNI(incidencia.getDNI());
                if (incidencia.getTelefono() != null)
                    incidenciaAlmacenada.setTelefono(incidencia.getTelefono());
                if (incidencia.getDescripcion() != null)
                    incidenciaAlmacenada.setDescripcion(incidencia.getDescripcion());
                if (incidencia.getLocalizacion() != null)
                    incidenciaAlmacenada.setLocalizacion(incidencia.getLocalizacion());
                if (incidencia.getTipoIncidencia() != null)
                    incidenciaAlmacenada.setTipoIncidencia(incidencia.getTipoIncidencia());
                if (incidencia.getProceso() != null)
                    incidenciaAlmacenada.setProceso(incidencia.getProceso());
                if (incidencia.getFechaInicio() != null)
                    incidenciaAlmacenada.setFechaInicio(incidencia.getFechaInicio());
                return incidenciaAlmacenada;
            }
        }
        throw new CustomException("Incidencia no encontrada con identificador: "+incidencia.getIdificador(), 4);
    }
    throw new CustomException("No se ha especificado la incidencia a actualizar ",1);
}
```



ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Cobertura de decisión:

Nº de aristas (a) = 34

Nº de nodos (n) = 24

Complejidad = $a - n + 2 = 34 - 24 + 2 = 12$

1. {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, F}

incidencia $\langle \rangle$ null = V

$i < \text{this.incidencia.size()} = V$

$\text{this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador())} = V$

Incidencia = {

identificador = indiferente,

nombreCiudadano = Jhon Doe,

DNI = 54101293H,

telefono = 34608219325,

descripcion = qwerty,

localizacion = qwerty,

tipoIncidencia = Otra,

proceso = Objeto proceso,

fechaInicio = 4/05/2021,

}

2. {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, F}

incidencia $\langle \rangle$ null = V

$i < \text{this.incidencia.size()} = V$

$\text{this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador())} = V$

Incidencia = {

identificador = indiferente,

nombreCiudadano = Jhon Doe,

DNI = 54101293H,

telefono = 34608219325,

descripcion = qwerty,

localizacion = qwerty,

tipoIncidencia = Otra,

proceso = Objeto proceso,

fechaInicio = null,

}

3. {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 21, F}

incidencia $\langle \rangle$ null = V

$i < \text{this.incidencia.size()} = V$

$\text{this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador())} = V$

Incidencia = {

identificador = indiferente,

nombreCiudadano = Jhon Doe,

DNI = 54101293H,

telefono = 34608219325,

descripcion = qwerty,

localizacion = qwerty,

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

        tipoIncidencia = Otra,
        proceso = null,
        fechaInicio = null,
    }
4.  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 19, 21, F}
    incidencia <> null = V
    i < this.incidencia.size() = V
    this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador()) = V
    Incidencia = {
        identificador = indiferente,
        nombreCiudadano = Jhon Doe,
        DNI = 54101293H,
        telefono = 34608219325,
        descripcion = qwerty,
        localizacion = qwerty,
        tipoIncidencia = null,
        proceso = null,
        fechaInicio = null,
    }
5.  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 17, 19, 21, F}
    incidencia <> null = V
    i < this.incidencia.size() = V
    this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador()) = V
    Incidencia = {
        identificador = indiferente,
        nombreCiudadano = Jhon Doe,
        DNI = 54101293H,
        telefono = 34608219325,
        descripcion = qwerty,
        localizacion = null,
        tipoIncidencia = null,
        proceso = null,
        fechaInicio = null,
    }
6.  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 17, 19, 21, F}
    incidencia <> null = V
    i < this.incidencia.size() = V
    this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador()) = V
    Incidencia = {
        identificador = indiferente,
        nombreCiudadano = Jhon Doe,
        DNI = 54101293H,
        telefono = 34608219325,
        descripcion = null,
        localizacion = null,
        tipoIncidencia = null,
        proceso = null,
        fechaInicio = null,
    }

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

7. {1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 15, 17, 19, 21, F}

incidencia <> null = V

i < this.incidencia.size() = V

this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador()) = V

```

Incidencia = {
    identificador = indiferente,
    nombreCiudadano = Jhon Doe,
    DNI = 54101293H,
    telefono = null,
    descripcion = null,
    localizacion = null,
    tipoIncidencia = null,
    proceso = null,
    fechaInicio = null,
}

```

8. {1, 2, 3, 4, 5, 6, 7, 9, 11, 13, 15, 17, 19, 21, F}

incidencia <> null = V

i < this.incidencia.size() = V

this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador()) = V

```

Incidencia = {
    identificador = indiferente,
    nombreCiudadano = Jhon Doe,
    DNI = null,
    telefono = null,
    descripcion = null,
    localizacion = null,
    tipoIncidencia = null,
    proceso = null,
    fechaInicio = null,
}

```

9. {1, 2, 3, 4, 5, 7, 9, 11, 13, 15, 17, 19, 21, F}

incidencia <> null = V

i < this.incidencia.size() = V

this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador()) = V

```

Incidencia = {
    identificador = indiferente,
    nombreCiudadano = null,
    DNI = null,
    telefono = null,
    descripcion = null,
    localizacion = null,
    tipoIncidencia = null,
    proceso = null,
    fechaInicio = null,
}

```

10. {1, 2, 3, 2, 23, F}

incidencia <> null = V

i < this.incidencia.size() = V

this.incidencias.get(i).getIdificador.equals(incidencia.getIdificador()) = F

ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

11. {1, 2, 23, F}

incidencia <> null = V

i < this.incidencia.size() = F

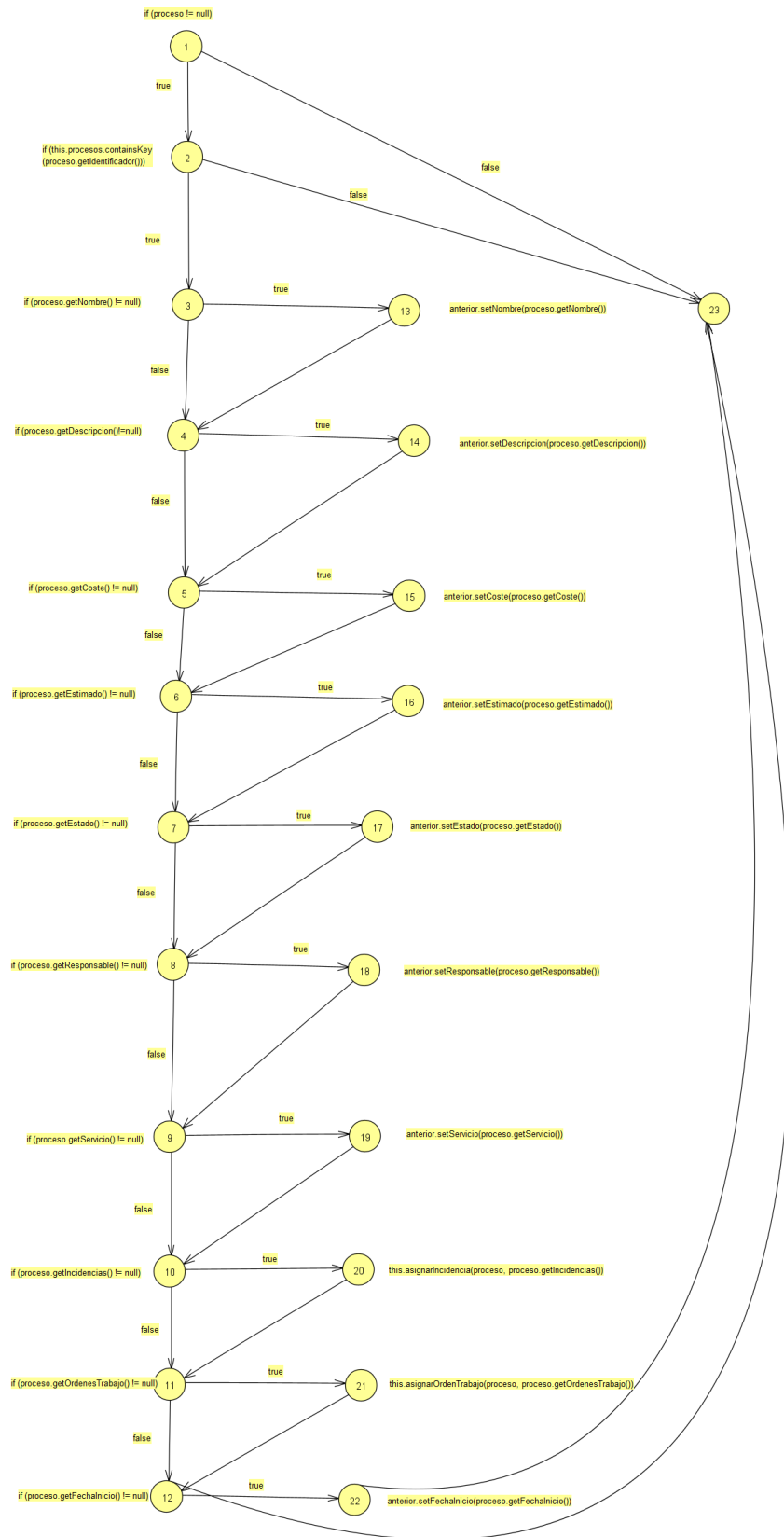
12. {1, 22, F}

incidencia <> null = F

○ *Actualizar Proceso*

```
public Proceso actualizar(Proceso proceso) throws CustomException{
    Proceso anterior=new Proceso();
    if(proceso!=null) {
        if(this.procesos.containsKey(proceso.getIdentificador())) {
            anterior=this.procesos.get(proceso.getIdentificador());
            if(proceso.getNombre()!=null)
                anterior.setNombre(proceso.getNombre());
            if(proceso.getDescripcion()!=null)
                anterior.setDescripcion(proceso.getDescripcion());
            if(proceso.getCoste()!=null)
                anterior.setCoste(proceso.getCoste());
            if(proceso.getEstimado()!=null)
                anterior.setEstimado(proceso.getEstimado());
            if(proceso.getEstado()!=null)
                anterior.setEstado(proceso.getEstado());
            if(proceso.getResponsable()!=null)
                anterior.setResponsable(proceso.getResponsable());
            if(proceso.getServicio()!=null)
                anterior.setServicio(proceso.getServicio());
            if(proceso.getIncidencias()!=null)
                this.asignarIncidencia(proceso, proceso.getIncidencias()); //Llamada a m todo asignar
            if(proceso.getOrdenesTrabajo()!=null)
                this.asignarOrdenTrabajo(proceso, proceso.getOrdenesTrabajo()); //Llamada a m todo asignar
            if(proceso.getFechaInicio()!=null)
                anterior.setFechaInicio(proceso.getFechaInicio());
        }
    }

    return anterior;
}
```



(Click para ampliar)

ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

Nº de aristas (a) = 34

Nº de nodos (n) = 23

Complejidad = $a - n + 2 = 34 - 23 + 2 = 13$

Cobertura de decisión (13 a probar):

1. {1, 2, 3, 13 4, 14, 5, 15, 6, 16, 7, 17, 8, 18, 9, 19, 10, 20, 11, 21, 12, 22, 23}

```
Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = 0;
    estado = "Pendiente";
    responsable = Paco Meralgo;
    servicio = "Asfalto";
    fechaInicio = 22/06/2021;
    ordenesTrabajo = ArrayList<OrdenTrabajo> {orden1, orden2};
    incidencias = ArrayList<Incidencia> {incidencia1, incidencia2};
}
```

2. {1, 2, 3, 13, 4, 14, 5, 15, 6, 16, 7, 17, 8, 18, 9, 19, 10, 20, 11, 21, 12, 23}

```
Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = 0;
    estado = "Pendiente";
    responsable = Paco Meralgo;
    servicio = "Asfalto";
    fechaInicio = null;
    ordenesTrabajo = ArrayList<OrdenTrabajo> {orden1, orden2};
    incidencias = ArrayList<Incidencia> {incidencia1, incidencia2};
}
```

3. {1, 2, 3, 13, 4, 14, 5, 15, 6, 16, 7, 17, 8, 18, 9, 19, 10, 20, 11, 12, 23}

```
Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = 0;
    estado = "Pendiente";
    responsable = Paco Meralgo;
    servicio = "Asfalto";
}
```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

        fechaInicio = null;
        ordenesTrabajo = null;
        incidencias = ArrayList<Incidencia> {incidencia1, incidencia2};
    }

```

4. {1, 2, 3, 13, 4, 14, 5, 15, 6, 16, 7, 17, 8, 18, 9, 19, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = 0;
    estado = "Pendiente";
    responsable = Paco Meralgo;
    servicio = "Asfalto";
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}

```

5. {1, 2, 3, 13, 4, 14, 5, 15, 6, 16, 7, 17, 8, 18, 9, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = 0;
    estado = "Pendiente";
    responsable = Paco Meralgo;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}

```

6. {1, 2, 3, 13, 4, 14, 5, 15, 6, 16, 7, 17, 8, 9, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = 0;
    estado = "Pendiente";
    responsable = null;
    servicio = null;
    fechaInicio = null;
}

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

ordenesTrabajo = null;
incidencias = null;
}

```

7. {1, 2, 3, 13, 4, 14, 5, 15, 6, 16, 7, 8, 9, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = 0;
    estado = null;
    responsable = null;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}

```

8. {1, 2, 3, 13, 4, 14, 5, 15, 6, 7, 8, 9, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = 15.0;
    estimado = null;
    estado = null;
    responsable = null;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}

```

9. {1, 2, 3, 13, 4, 14, 5, 6, 7, 8, 9, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = "proceso primero";
    coste = null;
    estimado = null;
    estado = null;
    responsable = null;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
}

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

        incidencias = null;
    }

```

10. {1, 2, 3, 13, 4, 5, 6, 7, 8, 9, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = "proceso-1";
    descripcion = null;
    coste = null;
    estimado = null;
    estado = null;
    responsable = null;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}

```

11. {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 23}

```

Proceso = {
    identificador = 0;
    nombre = null;
    descripcion = null.;
    coste = null;
    estimado = null;
    estado = null;
    responsable = null;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}

```

12. {1, 2, 23}

```

Proceso = {
    identificador = -1;
    nombre = null;
    descripcion = null.;
    coste = null;
    estimado = null;
    estado = null;
    responsable = null;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

}

13. {1, 23}

```
Proceso = {
    identificador = null;
    nombre = null;
    descripcion = null.;
    coste = null;
    estimado = null;
    estado = null;
    responsable = null;
    servicio = null;
    fechaInicio = null;
    ordenesTrabajo = null;
    incidencias = null;
}
```

○ **Crear OT**

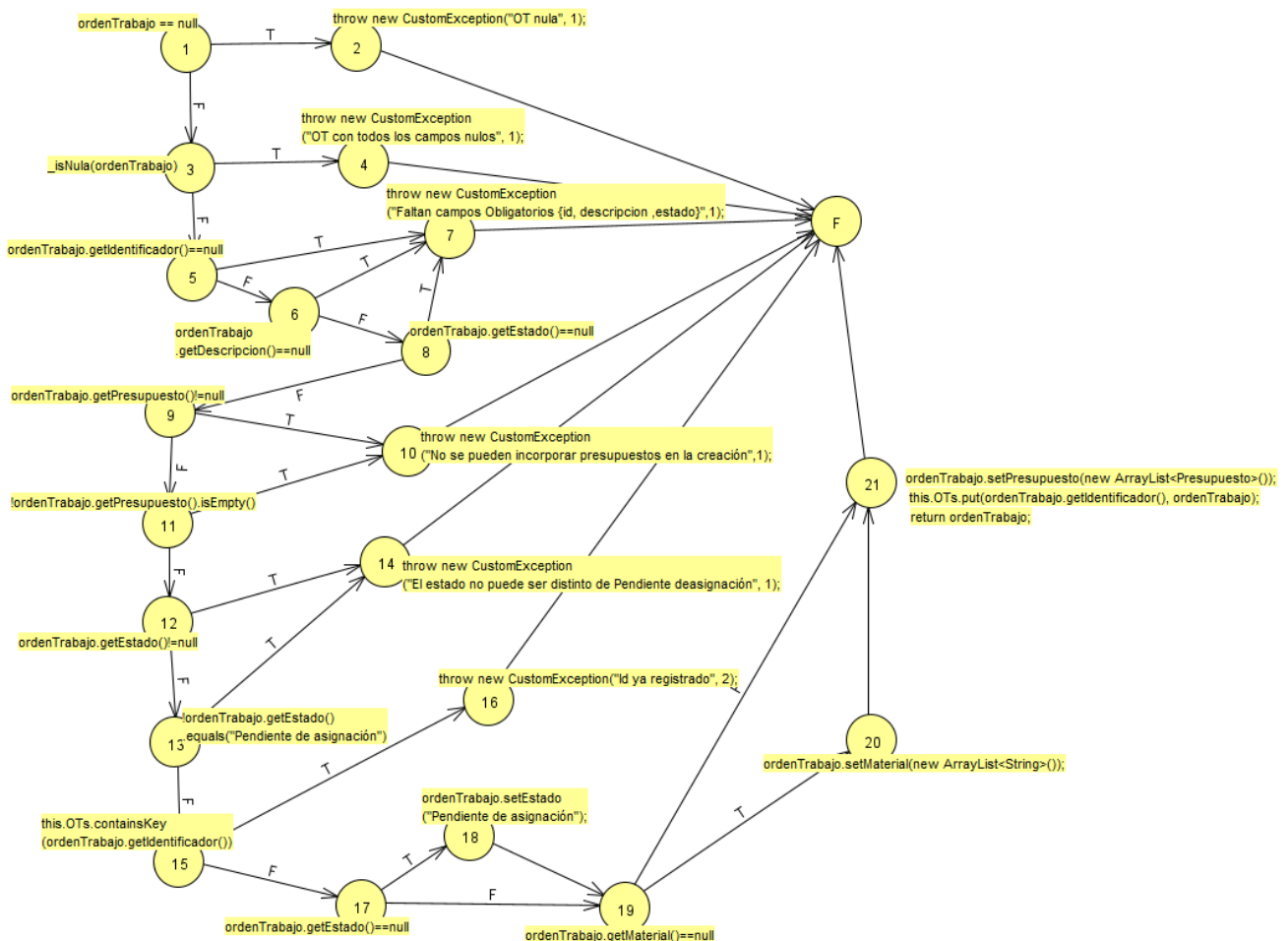
```

public OrdenTrabajo crear(OrdenTrabajo ordenTrabajo) throws CustomException{
    if(ordenTrabajo == null) {
        throw new CustomException("OT nula", 1);
    }else { //OT no nula
        if(!_isNula(ordenTrabajo)) { //todos los campos nulos
            throw new CustomException("OT con todos los campos nulos", 1);
        }else if(ordenTrabajo.getIdentificador()==null || ordenTrabajo.getDescripcion()==null || ordenTrabajo.getEstado()==null) { //Obligatios
            throw new CustomException("Faltan campos Obligatorios {id, descripcion ,estado}",1);
        }else if(ordenTrabajo.getPresupuesto()!=null && !ordenTrabajo.getPresupuesto().isEmpty()) {
            throw new CustomException("No se pueden incorporar presupuestos en la creación",1);
        }else if(ordenTrabajo.getEstado()!=null && !ordenTrabajo.getEstado().equals("Pendiente de asignación")) {
            throw new CustomException("El estado no puede ser distinto de Pendiente de asignación", 1);
        }else if(this.OTs.containsKey(ordenTrabajo.getIdentificador())) //Conflicto
            throw new CustomException("Id ya registrado", 2);
        }
        if(ordenTrabajo.getEstado()==null) {
            ordenTrabajo.setEstado("Pendiente de asignación");
        }

        //Inicializamos arrays
        if(ordenTrabajo.getMaterial()==null)
            ordenTrabajo.setMaterial(new ArrayList<String>());
        ordenTrabajo.setPresupuesto(new ArrayList<Presupuesto>()); //Por defecto es null

        //éxito
        this.OTs.put(ordenTrabajo.getIdentificador(), ordenTrabajo);
        return ordenTrabajo;
    }
}

```



ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Cobertura de decisión (13 a probar)

Nº de aristas (a) = 33

Nº de nodos (n) = 22

Complejidad = $a - n + 2 = 33 - 22 + 2 = 13$

1. {1, 3, 5, 6, 8, 9, 11, 12, 13, 15, 17, 18, 19, 20, 21, F}

```
OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= null
    Presupuesto= null
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado="Pendiente de asignación"
    Proceso = proceso
}
```

2. {1, 3, 5, 6, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, F}

```
OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= materiales1
    Presupuesto= null
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado="Pendiente de asignación"
    Proceso = proceso
}
```

3. {1, 3, 5, 6, 8, 9, 11, 12, 13, 15, 17, 19, 21, F}

```
OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= materiales1
    Presupuesto= null
    Coste = 5
```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

Responsable="Paco meralgo"
Personal=5
FechaInicio= "21/04/2021"
Duracion=2
Estado="Pendiente de asignación"
Proceso = proceso
}

```

4. {1, 3, 5, 6, 8, 9, 11, 12, 13, 15, 16, F}

El id de OT ya existe en el ArrayList.

```

OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= materiales1
    Presupuesto= null
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado="Pendiente de asignación"
    Proceso = proceso
}

```

5. {1, 3, 5, 6, 8, 9, 11, 12, 13, 14, F}

```

OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= materiales1
    Presupuesto= null
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado="Asignado"
    Proceso = proceso
}

```

6. {1, 3, 5, 6, 8, 9, 11, 12, 15, 16, F}

```

OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= materiales1
    Presupuesto= null
}

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

Coste = 5
Responsable="Paco meralgo"
Personal=5
FechaInicio= "21/04/2021"
Duracion=2
Estado=null
Proceso = proceso
}

```

7. {1, 3, 5, 6, 8, 9, 11, 10, F}

```

OrdenTrabajo = {
  Id=231456
  Descripcion="Reparar farola"
  Material= materiales1
  Presupuesto= presupuesto1
  Coste = 5
  Responsable="Paco meralgo"
  Personal=5
  FechaInicio= "21/04/2021"
  Duracion=2
  Estado=null
  Proceso = proceso
}

```

8. {1, 3, 5, 6, 8, 9, 12, 13, 14, F}

```

OrdenTrabajo = {
  Id=231456
  Descripcion="Reparar farola"
  Material= materiales1
  Presupuesto= null
  Coste = 5
  Responsable="Paco meralgo"
  Personal=5
  FechaInicio= "21/04/2021"
  Duracion=2
  Estado= "Asignado"
  Proceso = proceso
}

```

9. {1, 3, 5, 6, 8, 7, F}

```

OrdenTrabajo = {
  Id=231456
  Descripcion="Reparar farola"
  Material= materiales1
  Presupuesto= null
  Coste = 5

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado= null
    Proceso = proceso
}

```

10. {1, 3, 5, 6, 7, F}

```

OrdenTrabajo = {
    Id=231456
    Descripcion=null
    Material= materiales1
    Presupuesto= null
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado= null
    Proceso = proceso
}

```

11. {1, 3, 5, 7, F}

```

OrdenTrabajo = {
    Id=null
    Descripcion=null
    Material= materiales1
    Presupuesto= null
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado= null
    Proceso = proceso
}

```

12. {1, 3, 4, F}

```

OrdenTrabajo = {
    Id=null
    Descripcion=null
    Material= null
    Presupuesto= null
    Coste = null
    Responsable= null
}

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: <i>PDP_2C_v3.doc</i>	

```

Personal= null
FechaInicio= null
Duracion= null
Estado= null
Proceso = null
}

```

13. {1, 2, F}

OrdenTrabajo = null

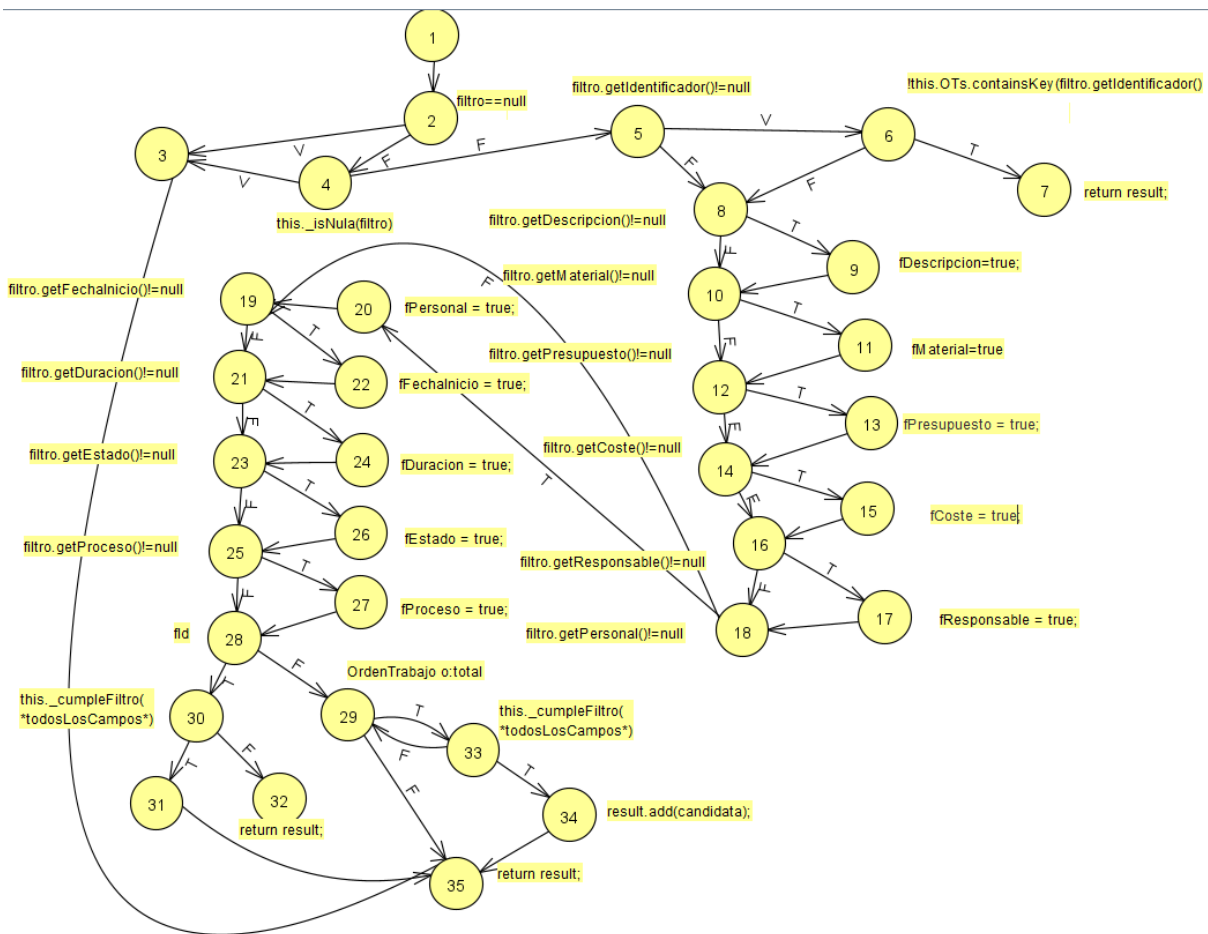
ENSO GrEI	Plan de pruebas	06/05/2021
	Doc.: PDP_2C_v3.doc	

○ *Buscar orden de trabajo*

```

public ArrayList<OrdenTrabajo> buscar(OrdenTrabajo filtro){
    ArrayList<OrdenTrabajo> result = new ArrayList<>();
    if(filtro==null || this._isNula(filtro)) {
        result = new ArrayList<>(this.OTs.values());
    }else { //filtramos
        Boolean fId = false, fDescripcion = false, fMaterial = false, fPresupuesto = false,
            fCoste = false, fResponsable = false, fPersonal = false, fFechaInicio = false,
            fDuracion = false, fEstado = false, fProceso = false;
        if(filtro.getIdentificador()!=null) {
            fId = true;
            if(!this.OTs.containsKey(filtro.getIdentificador())) {
                return result; //vacio
            }
        }
        if(filtro.getDescripcion()!=null)
            fDescripcion = true;
        if(filtro.getMaterial()!=null)
            fMaterial = true;
        if(filtro.getPresupuesto()!=null)
            fPresupuesto = false;
        if(filtro.getCoste()!=null)
            fCoste = true;
        if(filtro.getResponsable()!=null)
            fResponsable = true;
        if(filtro.getPersonal()!=null)
            fPersonal = true;
        if(filtro.getFechaInicio()!=null)
            fFechaInicio = true;
        if(filtro.getDuracion()!=null)
            fDuracion = true;
        if(filtro.getEstado()!=null)
            fEstado = true;
        if(filtro.getProceso()!=null)
            fProceso = true;
        //Aplicamos filtro
        if(fId) { //Solo puede haber una
            OrdenTrabajo candidata = this.OTs.get(filtro.getIdentificador());
            if(this._cumpleFiltro(fDescripcion, fMaterial, fPresupuesto, fCoste,
                fResponsable, fPersonal, fFechaInicio, fDuracion, fEstado,
                fProceso, candidata, filtro))
                result.add(candidata);
            else
                return result;
        }else {
            Collection<OrdenTrabajo> total = this.OTs.values();
            for(OrdenTrabajo o:total) {
                if(this._cumpleFiltro(fDescripcion, fMaterial, fPresupuesto, fCoste,
                    fResponsable, fPersonal, fFechaInicio, fDuracion, fEstado,
                    fProceso, o, filtro))
                    result.add(o);
            }
        }
    }
    return result;
}

```



Cobertura de caminos: 17 a probar

Nº de aristas (a) = 50

Nº de nodos (n) = 35

Complejidad = $a - n + 2 = 34 - 24 + 2 = 17$

1. {1,2,4,5,8,9,10,11,12,13,14,15,16,17,18,20,19,22,21,24,23,26,25,27,28,29,33,34,35}

```

OrdenTrabajo = {
  Id=null
  Descripcion="Reparar luz"
  Material= materiales
  Presupuesto= presupuestos
  Coste = 5
  Responsable="Paco meralgo"
  Personal=5
  FechaInicio= "21/04/2021"
  Duracion=2
  Estado="Asignado"

```

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Proceso = proceso
 }
 Todos los valores no nulos corresponden con los de una OT del sistema

2. {1,2,4,5,8,9,10,11,12,13,14,15,16,17,18,20,19,22,21,24,23,26,25,27,28,29,33,29,35}

```
OrdenTrabajo{
    Id=null
    Descripcion="Reparar farola"
    Material= materiales
    Presupuesto= presupuestos
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado="Terminado"
    Proceso = proceso
}
```

No existe ninguna OT con estas características.

3. {1,2,4,5,6,8,9,10,11,12,13,14,15,16,17,18,20,19,22,21,24,23,26,25,27,28,30,31,35}

```
OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= materiales
    Presupuesto= presupuestos
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
    Duracion=2
    Estado="Asignado"
    Proceso = proceso
}
```

Todos los valores no nulos corresponden con los de una OT del sistema

4. {1,2,4,5,8,9,10,11,12,13,14,15,16,17,18,20,19,22,21,24,23,26,25,27,28,30,32}

```
OrdenTrabajo = {
    Id=231456
    Descripcion="Reparar farola"
    Material= materiales
    Presupuesto= presupuestos
    Coste = 5
    Responsable="Paco meralgo"
    Personal=5
    FechaInicio= "21/04/2021"
```


ENSO GrEI	<i>Plan de pruebas</i>	06/05/2021
	Doc.: PDP_2C_v3.doc	

Duracion=2
Estado="Terminado"
Proceso = proceso

}

Los valores no coinciden con ninguna OT del sistema

5. {1,2,4,5,8,9,10,11,12,13,14,15,16,17,18,20,19,22,21,24,23,26,25,28,30,32}

OrdenTrabajo = {
Id= 231456
Descripcion="Reparar farola"
Material= materiales
Presupuesto= presupuestos
Coste = 5
Responsable="Paco meralgo"
Personal=5
FechaInicio= "21/04/2021"
Duracion=2
Estado="Terminado"
Proceso = null

}

Los valores no coinciden con ninguna OT del sistema

6. {1,2,4,5,6,8,9,10,11,12,13,14,15,16,17,18,20,19,22,21,24,23,25,28,30,32}

OrdenTrabajo = {
Id= 231456
Descripcion="Reparar farola"
Material= materiales
Presupuesto= presupuestos
Coste = 5
Responsable="Paco meralgo"
Personal=5
FechaInicio= "21/04/2021"
Duracion=2
Estado=null
Proceso = null

}

Los valores no coinciden con ninguna OT del sistema

7. {1,2,4,5,6,8,9,10,11,12,13,14,15,16,17,18,20,19,22,21,23,25,28,30,32}

OrdenTrabajo = {
Id= 231456
Descripcion="Reparar farola"
Material= materiales
Presupuesto= presupuestos
Coste = 5
Responsable="Paco meralgo"

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Personal=5
 FechaInicio= "21/04/2021"
 Duracion=null
 Estado=null
 Proceso = null

}

Los valores salvo el id, no coinciden con ninguna OT del sistema

8. {1,2,4,5,6,8,9,10,11,12,13,14,15,16,17,18,20,19,21,23,25,28,30,32}

OrdenTrabajo = {
 Id= 231456
 Descripcion="Reparar farola"
 Material= materiales
 Presupuesto= presupuestos
 Coste = 5
 Responsable="Paco meralgo"
 Personal=5
 FechaInicio= null
 Duracion=null
 Estado=null
 Proceso = null

}

Los valores salvo el id, no coinciden con ninguna OT del sistema

9. {1,2,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19,21,23,25,28,30,32}

OrdenTrabajo = {
 Id= 231456
 Descripcion="Reparar farola"
 Material= materiales
 Presupuesto= presupuestos
 Coste = 5
 Responsable="Paco meralgo"
 Personal=null
 FechaInicio= null
 Duracion=null
 Estado=null
 Proceso = null

}

Los valores salvo el id, no coinciden con ninguna OT del sistema

10. {1,2,4,5,6,8,9,10,11,12,13,14,15,16,18,19,21,23,25,28,30,32}

OrdenTrabajo = {
 Id= 231456
 Descripcion="Reparar farola"
 Material= materiales
 Presupuesto= presupuestos

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Coste = 5
 Responsable=null
 Personal=null
 FechaInicio= null
 Duracion=null
 Estado=null
 Proceso = null

}

Los valores salvo el id, no coinciden con ninguna OT del sistema

11. {1,2,4,5,6,8,9,10,11,12,13,14,16,18,19,21,23,25,28,30,32}

OrdenTrabajo = {
 Id= 231456
 Descripcion="Reparar farola"
 Material= materiales
 Presupuesto= presupuestos
 Coste = null
 Responsable=null
 Personal=null
 FechaInicio= null
 Duracion=null
 Estado=null
 Proceso = null

}

Los valores salvo el id, no coinciden con ninguna OT del sistema

12. {1,2,4,5,6,8,9,10,11,12,14,16,18,19,21,23,25,28,30,32}

OrdenTrabajo = {
 Id= 231456
 Descripcion="Reparar farola"
 Material= materiales
 Presupuesto= null
 Coste = null
 Responsable=null
 Personal=null
 FechaInicio= null
 Duracion=null
 Estado=null
 Proceso = null

}

Los valores salvo el id, no coinciden con ninguna OT del sistema

13. {1,2,4,5,6,8,9,10,12,14,16,18,19,21,23,25,28,30,32}

OrdenTrabajo = {
 Id= 231456
 Descripcion="Reparar farola"

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

Material= null
Presupuesto= null
Coste = null
Responsable=null
Personal=null
FechaInicio= null
Duracion=null
Estado=null
Proceso = null

}

Los valores salvo el id, no coinciden con ninguna OT del sistema

14. {1,2,4,5,6,8,10,12,14,16,18,19,21,23,25,28,30,31,35}

OrdenTrabajo = {

Id= 231456
Descripcion=null
Material= null
Presupuesto= null
Coste = null
Responsable=null
Personal=null
FechaInicio= null
Duracion=null
Estado=null
Proceso = null

}

Los valores no coinciden con ninguna OT del sistema

15. {1,2,4,5,6,7}

OrdenTrabajo = {

Id= 12
Descripcion= null
Material= null
Presupuesto= null
Coste = null
Responsable=null
Personal=null
FechaInicio= null
Duracion=null
Estado=null
Proceso = null

}

El id no coincide con una OT del sistema

16. {1,2,4,3,35}

OrdenTrabajo = {

Id= null

ENSO GrEI	<i>Plan de pruebas</i>	<i>06/05/2021</i>
	Doc.: PDP_2C_v3.doc	

```

        Descripcion= null
        Material= null
        Presupuesto= null
        Coste = null
        Responsable=null
        Personal=null
        FechaInicio= null
        Duracion=null
        Estado=null
        Proceso = null
    }

```

Los valores no coinciden con ninguna OT del sistema

17. {1,2,3,35}

Introducir como argumento un null directamente

Anexo 2. –Bibliografía y material utilizado

- Enxenaaría do Software. **Sample Test Process.pdf** [Online] Campus Virtual USC. (Se ha utilizado para realizar toda la práctica).
- Enxenaaría do Software. **ieee829mtp.pdf** [Online] Campus Virtual USC. (Se ha utilizado para realizar toda la práctica).
- Enxenaaría do Software. **04_pdp.pdf**[Online] Campus Virtual USC. (Se ha utilizado para realizar toda la práctica).

