

Ingeniería de Software

Proceso de Pruebas

Entrega 2

Crístofer Canosa Domínguez
Silvia Rodríguez Alcaraz
Orquídea Seijas Salinas
Samuel Soutullo Sobral

FECHA DE ENTREGA: 02/04/2017

CONTROL DE VERSIONES		
Edición	Fecha	Descripción del cambio
1.0	02/04/2017	Diseño y casos de prueba de caja negra

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

ÍNDICE

1	PLAN DE PRUEBAS	1
1.1	IDENTIFICADOR ÚNICO DEL DOCUMENTO	1
1.2	INTRODUCCIÓN	1
1.2.1	DOCUMENTOS RELACIONADOS	1
1.3	ELEMENTOS SOFTWARE A PROBAR	1
1.4	CARACTERÍSTICAS A PROBAR.....	1
1.5	CARACTERÍSTICAS QUE NO SE PRUEBAN.....	1
1.6	ENFOQUE GENERAL DE LA PRUEBA	2
1.7	CRITERIOS DE PASO/FALLO PARA CADA ELEMENTO	2
1.8	CRITERIOS DE SUSPENSIÓN Y REQUISITOS DE REANUDACIÓN	2
1.9	DOCUMENTOS A ENTREGAR	3
1.10	ACTIVIDADES DE PREPARACIÓN Y EJECUCIÓN DE PRUEBAS.....	3
1.11	NECESIDADES DE ENTORNO	3
1.12	RESPONSABILIDADES EN LA ORGANIZACIÓN Y REALIZACIÓN DE LAS PRUEBAS.....	4
1.13	NECESIDADES DE PERSONAL Y DE FORMACIÓN	4
1.14	ESQUEMA DE TIEMPOS	5
1.15	RIESGOS ASUMIDOS POR EL PLAN Y PLANES DE CONTINGENCIAS PARA CADA RIESGO	5
1.16	APROBACIONES Y FIRMAS CON NOMBRE Y PUESTO DESEMPEÑADO	5
2	DISEÑO DE PRUEBAS	6
2.1	INTERFAZIMPORTACIÓN.....	6
2.1.1	PRUEBA I-01.....	6
2.1.2	PRUEBA I-02.....	6
2.1.3	PRUEBA I-03.....	8
2.2	INTERFAZDAO.....	9
2.2.1	PRUEBA D-01	9
2.2.2	PRUEBA D-02	11
2.2.3	PRUEBA D-03	12
2.2.4	PRUEBA D-04	13
2.2.5	PRUEBA D-05	14
2.3	INTERFAZESTADISTICA	14

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

2.3.1	PRUEBA D-06	14
2.4	INTERFAZCONTROLADOR	15
2.4.1	NOTA.....	15
2.4.2	PRUEBA C-01	15
2.4.3	PRUEBA C-02	16
2.4.4	PRUEBA C-03	17
2.4.5	PRUEBA C-04	17
2.4.6	PRUEBA C-05	18
3	CASOS DE PRUEBA	18
3.1	INTERFAZIMPORTACION.....	18
3.1.1	CASO DE PRUEBA I-01-P-01	18
3.1.2	CASO DE PRUEBA I-01-P-02	19
3.1.3	CASO DE PRUEBA I-01-P-03	19
3.1.4	CASO DE PRUEBA I-01-P-04	19
3.1.5	CASO DE PRUEBA I-02-P-01	20
3.1.6	CASO DE PRUEBA I-02-P-02	20
3.1.7	CASO DE PRUEBA I-02-P-03	20
3.1.8	CASO DE PRUEBA I-02-P-04	21
3.1.9	CASO DE PRUEBA I-02-P-05	21
3.1.10	CASO DE PRUEBA I-02-P-06	22
3.1.11	CASO DE PRUEBA I-02-P-07	22
3.1.12	CASO DE PRUEBA I-02-P-08	22
3.1.13	CASO DE PRUEBA I-02-P-09	23
3.1.14	CASO DE PRUEBA I-02-P-10	23
3.1.15	CASO DE PRUEBA I-03-P-01	24
3.1.16	CASO DE PRUEBA I-03-P-02	24
3.1.17	CASO DE PRUEBA I-03-P-03	24
3.2	INTERFAZDAO.....	25
3.2.1	CASO DE PRUEBA D-01-P-01.....	25
3.2.2	CASO DE PRUEBA D-01-P-02.....	25
3.2.3	CASO DE PRUEBA D-01-P-03.....	25
3.2.4	CASO DE PRUEBA D-01-P-04.....	26
3.2.5	CASO DE PRUEBA D-01-P-05.....	26
3.2.6	CASO DE PRUEBA D-01-P-06.....	26
3.2.7	CASO DE PRUEBA D-01-P-07.....	27
3.2.8	CASO DE PRUEBA D-02-P-01.....	27
3.2.9	CASO DE PRUEBA D-02-P-02.....	28
3.2.10	CASO DE PRUEBA D-02-P-03.....	28
3.2.11	CASO DE PRUEBA D-02-P-04.....	28
3.2.12	CASO DE PRUEBA D-02-P-05.....	29
3.2.13	CASO DE PRUEBA D-02-P-06.....	29
3.2.14	CASO DE PRUEBA D-03-P-01.....	29
3.2.15	CASO DE PRUEBA D-03-P-02.....	30
3.2.16	CASO DE PRUEBA D-03-P-03.....	30
3.2.17	CASO DE PRUEBA D-04-P-01.....	30
3.2.18	CASO DE PRUEBA D-04-P-02.....	31
3.2.19	CASO DE PRUEBA D-04-P-03.....	31
3.2.20	CASO DE PRUEBA D-04-P-04.....	31

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

3.2.21	CASO DE PRUEBA D-05-P-01.....	32
3.2.22	CASO DE PRUEBA D-05-P-02.....	32
3.2.23	CASO DE PRUEBA D-05-P-03.....	33
3.2.24	CASO DE PRUEBA D-05-P-04.....	33
3.3	INTERFAZESTADISTICA	33
3.3.1	NOTA.....	33
3.3.2	CASO DE PRUEBA E-01-P-01.....	34
3.3.3	CASO DE PRUEBA E-01-P-02.....	34
3.3.4	CASO DE PRUEBA E-01-P-03.....	34
3.3.5	CASO DE PRUEBA E-01-P-04.....	35
3.4	INTERFAZCONTROLADOR	35
3.4.1	CASO DE PRUEBA C-01-P-01.....	35
3.4.2	CASO DE PRUEBA C-01-P-02.....	35
3.4.3	CASO DE PRUEBA C-01-P-03.....	36
3.4.4	CASO DE PRUEBA C-01-P-04.....	36
3.4.5	CASO DE PRUEBA C-01-P-05.....	36
3.4.6	CASO DE PRUEBA C-02-P-01.....	37
3.4.7	CASO DE PRUEBA C-02-P-02.....	37
3.4.8	CASO DE PRUEBA C-03-P-01.....	37
3.4.9	CASO DE PRUEBA C-03-P-02.....	38
3.4.10	CASO DE PRUEBA C-03-P-03.....	38
3.4.11	CASO DE PRUEBA C-04-P-01.....	38
3.4.12	CASO DE PRUEBA C-04-P-02.....	39
3.4.13	CASO DE PRUEBA C-05-P-01.....	39
3.4.14	CASO DE PRUEBA C-05-P-02.....	40
4	PROCEDIMIENTOS DE PRUEBA	40
5	ANEXO: PLANTILLAS	40
5.1	PLANTILLA DE PLAN DE PRUEBAS	40
5.2	PLANTILLA DE DISEÑO DE PRUEBAS.....	41
5.3	PLANTILLA DE CASOS DE PRUEBA	41

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

1 Plan de pruebas

1.1 Identificador único del documento

PDP_v1

1.2 Introducción

Este documento se corresponde con el plan de pruebas correspondiente al proyecto de ventas de la USC. El software se basa en un portal web mediante el cual los usuarios pueden solicitar pedidos, siendo tramitados posteriormente por la universidad.

El objetivo de este plan es marcar las pautas y definir la estrategia que se seguirá para certificar el software. En este caso, los casos de prueba y procedimientos tan sólo están generados mediante métodos de caja negra contra la especificación de las interfaces del software.

Por tanto, los elementos sobre los que se aplicarán las pruebas son las interfaces correspondientes a los módulos de importación, control, estadística y acceso a base de datos (DAO). Dado que las pruebas son de caja negra, se centrarán en el estudio de las especificaciones: testear las entradas para comprobar si se obtienen las salidas esperadas. Para la selección de casos de prueba se utilizó un enfoque sistemático basado en el método de partición o clases de equivalencia.

1.2.1 Documentos relacionados

Nombre	Descripción
02_PlantAnaDisPlan_v1.pdf	Documento de análisis y diseño del software sobre el cual
Std_IEEE_829.pdf	Estándar 829.

1.3 Elementos software a probar

Como ya se ha mencionado en la introducción, los elementos a probar son los módulos de importación, estadístico, control y DAO a través de pruebas de caja negra sobre sus interfaces.

1.4 Características a probar

Dado que son pruebas funcionales, las características de las interfaces a probar son las entradas pertinentes a sus métodos. Se pretende comprobar, mediante una serie de casos de prueba planteados mediante partición, que para una entrada dada se obtiene la salida esperada.

1.5 Características que no se prueban

Partiendo de que se utilizan pruebas de caja negra, todas las características a nivel estructural del software (caja blanca) no se probarán en esta versión del software. Además, puesto que se ha planteado centrar la búsqueda de errores en las interfaces, no se realizarán pruebas centradas en las funciones incorrectas o ausentes, estructuras de datos o acceso a BBDD, rendimiento o errores de inicialización y terminación.

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

1.6 Enfoque general de la prueba

Dada la importancia de los formatos para los distintos atributos expuestos en los requisitos, las pruebas se centraron en asegurar su correcta interpretación. Para ello se utilizó principalmente el método de partición para crear casos de prueba.

Como segunda técnica se usó el análisis por valores límite y, con menor frecuencia, la conjetura de errores.

Teniendo en cuenta que las pruebas se centran en generación de clases del modelo, muchos de los métodos se consideraron similares entre sí, asumiendo que si uno de ellos es correcto sería apropiado pensar que el resto también lo serán. Sin embargo, atendiendo al teorema de Pareto, en el caso de que uno de estos métodos falle, será conveniente ampliar los casos de prueba a todos sus métodos similares.

1.7 Criterios de paso/fallo para cada elemento

La aprobación o rechazo de un elemento sobre el que se han aplicado pruebas se basará en un criterio basado en gravedad de errores. El elemento podrá ser aprobado si se han realizado el 100% de pruebas sobre él y ha obtenido un 90% de éxito. El 10% se correspondería a errores medios y leves, pero nunca graves. Un elemento que contenga errores graves, aunque sólo sea uno, no podrá ser aprobado. A continuación, se especifica qué se entiende por error leve, medio y grave:

- **Error leve:** el elemento muestra un comportamiento correcto, pero ligeramente variable en situaciones similares. También se entienden como errores leves pequeñas desviaciones con respecto a los estándares a seguir.
- **Error medio:** es un error medio el hecho de que los métodos del elemento a probar, pese a funcionar, realicen sus operaciones de manera incorrecta o no de la forma previamente planteada. En otras palabras, que el elemento no cumpla los requisitos que se le habían impuesto pese a su aparente buen funcionamiento.
- **Error grave:** el elemento no cumple el objetivo para el que fue diseñado. También se considera error grave cuando la salida de un método del elemento provoca una situación inesperada y no controlada.

1.8 Criterios de suspensión y requisitos de reanudación

En ocasiones cuando el número o tipo de errores llega a un punto en el que el seguimiento de la prueba no tiene valor, es mejor barajar el hecho de frenar el proceso de pruebas. A continuación, se indican las situaciones en las que se frenaría dicho proceso con el fin de no desperdiciar recursos:

- **Existencia de un error grave:** una vez se encuentra un error grave en un elemento a probar dentro de un caso de prueba, lo mejor es frenar dicha prueba dado que, como ya se ha mencionado, no es posible que vaya a aprobar. En este caso, el elemento deberá ser estudiado y depurado con el fin de remediar dicho error y continuar con las pruebas más adelante.
No dejando pasar errores críticos se evita el confundir errores fantasma provocados por este error con meros defectos.
- **Ningún elemento está preparado para ser probado:** en este caso, realizar pruebas sobre elementos que todavía se encuentran en un estado prematuro sólo conseguirá desperdiciar recursos y no obtener ningún tipo de información

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

esclarecedora. Lo mejor en este caso, es dejar pasar un período de tiempo donde se complementen los elementos y, en este punto, realizar sobre ellos las pruebas.

- **Se supera un 20% de errores al probar un elemento:** aunque sean errores leves o medios, un elemento con un 20% de errores acabará provocando más fallos en el conjunto de la aplicación en algún momento. De alcanzar y superar este límite, este elemento debe ser revisado para poder volver a realizar pruebas sobre él en un futuro.
- **No disponibilidad del software apropiado para realizar las pruebas:** si las pruebas han sido planteadas para realizar de una forma concreta es mejor esperar a tener todas las herramientas disponibles. Sustituir estas herramientas por otras puede provocar una pérdida de tiempo posterior o el rediseño de alguna prueba.

1.9 Documentos a entregar

Puesto que en este punto tan sólo se ha planteado la aplicación de pruebas de caja negra, los documentos a entregar tras la realización de las pruebas serían:

- Informe de pruebas de caja negra.
- Código de los elementos a probar
- Código de las pruebas que se realizaron

1.10 Actividades de preparación y ejecución de pruebas

Para el correcto desarrollo de las pruebas se necesita contar con equipos que contengan el software que se utilizará para su realización. En este caso, Mockito y JUnit, además del correspondiente IDE de trabajo que compartirán todos los programadores de pruebas a la hora de codificar: Eclipse. Es obvio que, además, todo el personal implicado en la ejecución de las pruebas tendrá acceso al código de las implementado, el código de los elementos a probar y el plan de pruebas a seguir.

1.11 Necesidades de entorno

Para la realización de las pruebas será necesario tener una base de datos funcional con los datos necesarios para cada uno de los casos de prueba. Dado que los casos serán idempotentes siempre que no se indique lo contrario, serán ellos los encargados de inicializar la base de datos a sus valores necesarios para la correcta ejecución.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

1.12 Responsabilidades en la organización y realización de las pruebas

Para la realización de las pruebas se precisará un equipo de pruebas que podría estar formado por personas que cumplan los siguientes roles (entiéndase que sólo existiría un jefe de proyecto):

Rol	Responsabilidad
Jefe del proyecto	Responsable de estudiar las condiciones de finalización del proceso de pruebas. Será tarea suya concluir si el producto probado está listo o no. También es su deber coordinar al resto del equipo y delegar sobre él.
Analista	Persona encargada de resolver las incidencias, analizar los resultados y generar los informes correspondientes para comunicar los resultados técnicos al jefe de proyecto.
Programador de pruebas	Responsable de la generación del plan de pruebas.

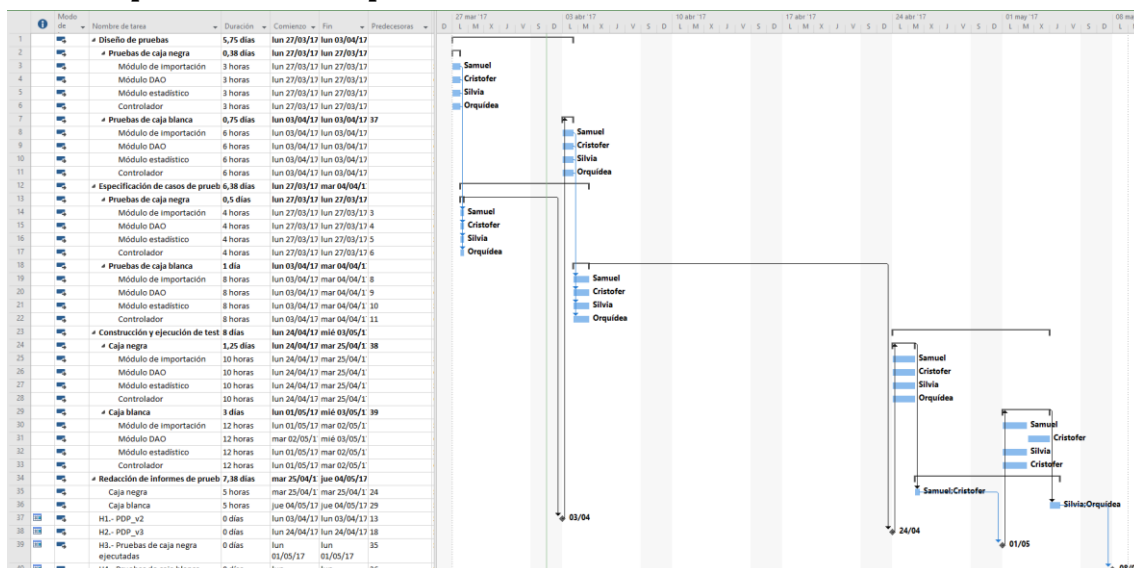
1.13 Necesidades de personal y de formación

Por una parte, el personal debe poseer los conocimientos técnicos informáticos suficientes como para poder implementar las pruebas que se van a realizar y comprender el código a probar, además de saber manejar las herramientas planteadas para el desarrollo de las pruebas.

Por otra parte, es necesaria una figura que lidere el equipo de pruebas y coordine a todos sus integrantes, con capacidad de liderazgo. Sus conocimientos estarían más orientados a la empresa pese a que sería apropiado que tuviese implicación dentro de las TICs.

Finalmente, sería necesaria una persona centrada en el análisis e interpretación de los resultados obtenidos en las pruebas, que comunicase el equipo de *testing* con el jefe de proyecto. Sería conveniente que esta persona tuviese conocimientos en el área de la ingeniería y, como requisito recomendable, experiencia previa.

1.14 Esquema de tiempos



1.15 Riesgos asumidos por el plan y planes de contingencias para cada riesgo

Riesgo	Descripción	Contingencia
Fuerte limitación temporal para el planteamiento de las pruebas.	Intervalo de tiempo escaso para diseñar el plan de pruebas.	Generación de pruebas de caja negra sin excesivo nivel de detalle para cumplir las fechas de entrega.
Cambios en los requisitos o diseños originales.	Incoherencias y falta de información en el documento de análisis y diseño.	Diálogo intergrupar para solucionar posibles malas interpretaciones.
Recursos con escaso tiempo para asignar a las pruebas.	Recursos con poca disponibilidad.	Redistribución de la carga entre recursos más disponibles.

1.16 Aprobaciones y firmas con nombre y puesto desempeñado

La aprobación para definir el proceso como completo y permitir que avance al siguiente nivel es una tarea cooperativa entre dos departamentos. Mediante una reunión formal en la que se encuentren el jefe de desarrollo y el analista a cargo del proyecto se decidirá si el software está preparado. Se precisa de la aprobación de ambos ya que es necesario contar con un aprobado a nivel técnico y otro más enfocado a un nivel empresarial. De esta forma, mediante el debate y el estudio de los informes de pruebas se considera que el software pasa a la siguiente fase si se aprueba por unanimidad.

La firma del analista y el jefe de desarrollo supondrían la aprobación del software y, por tanto, el pase a la siguiente fase del proyecto.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

2 Diseño de pruebas

2.1 InterfazImportación

2.1.1 Prueba I-01

2.1.1.1 Objetivo

Comprobar que un archivo con líneas incompletas, con demasiados campos, o en blanco no provoca un comportamiento inesperado en el programa.

2.1.1.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R1:** incluir una línea con campos de menos (todos ellos con valores válidos), otra línea con campos de más (los que no se hayan incluido a mayores deben tener valores válidos) y una línea con el número de campos correctos (todos ellos con valores válidos).
- **R3:** incluir una línea en blanco y otra completa y correcta.

Dado que las clases de equivalencia se obtienen en base al contenido del archivo, se aplicarán a los tres métodos de la interfaz (*importarUsuarios*, *importarProducto* e *importarCompra*).

2.1.1.3 Técnicas de caja blanca

2.1.1.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada como consecuencia del formato inesperado de la línea del fichero. Cuando un archivo contenga líneas incorrectas se debe cancelar la importación informando al usuario del error. Por otra parte, un archivo con líneas correctas debe ser correctamente procesado.

2.1.1.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
Línea	String[]	R1	{1} Número de campos correcto.	{2} Número de campos inferior al correcto. {3} Número de campos superior al correcto.
		R3	{4} Línea correcta.	{5} Línea en blanco.

2.1.2 Prueba I-02

2.1.2.1 Objetivo

Comprobar que un archivo que contenga un campo con un formato o valor incorrecto no provoca un comportamiento inesperado en el programa.

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

2.1.2.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R1:** para todos los campos con longitud máxima delimitada, se probará introduciendo el campo vacío, el campo con una longitud correcta, y el campo con una longitud superior a la correcta.
- **R3:** para cada uno de los campos sujetos a restricción de formato, se probará un caso con formato correcto y otro con formato incorrecto (por ejemplo: fecha con formato adecuado, unidades, cantidad y precio deben ser numéricos, etc.).
- **R3:** para cada uno de los campos identificadores, se probará un caso con un identificador ya presente en la base de datos, y otro no presente en la base de datos. En otras palabras, se probará a importar un elemento ya presente en la base de datos y otro que no lo esté.
- **R3:** para los campos de referencia a ítem y usuario en las líneas de venta, se probará un caso en el que el identificador del ítem o usuario referenciado se encuentre previamente en la base de datos y otro caso en el que no.

Dado que las anteriores clases de equivalencia se obtienen en base al contenido del archivo, se aplicarán a los tres métodos de la interfaz (*importarUsuarios*, *importarProducto* e *importarCompra*).

Dada su relación directa con el tipo de línea en cuestión, las reglas que se listan a continuación sí se aplican de manera diferente a cada método:

- ***importarUsuarios***
 - **R3:** un caso en el que el primer campo sea *U* y otro en el que sea distinto de *U*.
- ***importarProducto***
 - **R3:** un caso en el que el primer campo sea *I* y otro en el que sea distinto de *I*.
- ***importarCompra***
 - **R3:** un caso en el que el primer campo sea *V* y otro en el que sea distinto de *V*.
 - **R3:** para los campos de referencia a ítem y usuario en las líneas de venta, se probará un caso en el que el identificador del ítem o usuario referenciado se encuentre previamente en la base de datos y otro caso en el que no.

Aplicando las técnicas de análisis por valores límite se obtiene lo siguiente:

- **R1:** para todos los campos con longitud máxima *m*, se probará introduciendo valores con longitud 0, 1, *m* y *m* + 1.

Finalmente, aplicando la técnica de conjetura de errores, se obtiene lo siguiente:

- Se debe comprobar si la existencia de varias líneas de venta con el mismo identificador provoca que se añadan varios ítems a la misma venta. Si esto no sucede, el comportamiento es incorrecto.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

2.1.2.3 Técnicas de caja blanca

2.1.2.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada como consecuencia del formato o valor inesperado del campo. Cuando un archivo contenga uno o más campos incorrectos se debe cancelar la importación informando al usuario del error. Por otra parte, un archivo con todos los campos correctos debe ser correctamente procesado.

2.1.2.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
Nombre	String	R1	{1} Longitud entre 1 y 255 caracteres.	{2} Longitud de cero caracteres.
Apellidos				{3} Longitud superior a 255 caracteres.
Categoría				
Descripción	String	R1	{4} Longitud entre 1 y 1024 caracteres.	{5} Longitud de cero caracteres. {6} Longitud superior a 1024 caracteres.
Fecha	Date	R3	{7} Fecha con formato dd/mm/aaaa	{8} Fecha formato distinto a dd/mm/aaaa
Unidades	Integer	R3	{8} Formato de número entero.	{9} Formato distinto a número entero.
Cantidad				
PrecioUnidad	Float	R3	{10} Formato de número con dígitos decimales.	{11} Formato distinto de número con dígitos decimales.
IdxUser	String	R3	{12} Identificador no presente previamente en la base de datos.	{13} Identificador previamente presente en la base de datos.
ItemRef				
VRef		R3	{14} Primer campo con el valor adecuado para el método llamado	{15} Primer campo con un valor incorrecto para el método llamado
IdxUser	String	R3	{16} Identificador referenciado previamente en base de datos.	{17} Identificador referenciado no previamente en base de datos.
ItemRef				

2.1.3 Prueba I-03

2.1.3.1 Objetivo

Comprobar que la importación se realiza correctamente cuando se pasan argumentos válidos y que no se genera ninguna excepción sin controlar cuando se pasan argumentos inválidos.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

2.1.3.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R3:** se probará, para los tres métodos de importación, un caso con un *path* con formato válido y otro caso con formato inválido.
- **R3:** se probará, para los tres métodos de importación, un caso con un *path* que referencia a un archivo existente, y otro caso que referencia a un archivo inexistente.

2.1.3.3 Técnicas de caja blanca

2.1.3.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada como consecuencia de que los métodos reciban argumentos inválidos. Tanto si el fichero no existe como si la ruta tiene un formato incorrecto, se debe informar al usuario mediante un error. Por otra parte, si se pasa como argumento la ruta de un fichero de importación válido, la importación se debe realizar correctamente.

2.1.3.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
Path	String	R3	{1} El <i>path</i> tiene un formato válido.	{2} El <i>path</i> no tiene un formato válido.
		R3	{3} El <i>path</i> referencia a un fichero existente.	{4} El <i>path</i> referencia a un fichero inexistente.

2.2 InterfazDAO

2.2.1 Prueba D-01

2.2.1.1 Objetivo

Comprobar que las inserciones o modificaciones de usuarios se reflejan correctamente en la base de datos.

Además, se busca que, en caso de error, este sea correctamente gestionado.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

2.2.1.2 Técnicas de caja negra

2.2.1.2.1 Generación de clases de equivalencia

Centrándose en los métodos de inserción y modificación de usuarios y haciendo uso de las reglas especificadas, se definen las siguientes clases:

- **Insertar usuario**
 - **R3:** Probar un usuario con ID válido y otro sin ID.
 - **R5:** Los Id de usuario de forma N-XXXXXX-000.
 - R1: Cadena de letras con 5 y 7 caracteres (incorrecto) y una correcta. Clase equivalente para la cadena numérica.
 - R3: Eliminar uno de los campos del ID.
 - R3: Que N sea U u otro caracter.
- **Modificar usuario**
 - Se supone que la introducción del usuario modificado en la base de datos será equivalente a la usada en el método anterior y, por tanto, ya está probada.
 - **R3:** Insertar datos modificados de un usuario existente (caso correcto) e insertar datos de un usuario que no existe.

2.2.1.3 Técnicas de caja blanca

2.2.1.4 Criterios de paso/fallo

Los usuarios introducidos o modificados usando datos correctos deben verse reflejados en la base de datos.

En los casos de incluir usuarios de forma errónea debe notificarse del error y en ningún caso reflejarse en la base de datos.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

2.2.1.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
insertUser	Usuario	R3	{1} Id válido.	{2} Id inválido
		R5,R1	{3} Cadena de 6 letras en el Id	{4} Cadena de 5 letras
				{5} Cadena de 7 letras
		R5,R1	{6} Cadena de 3 dígitos en el ID	{7} Cadena de 2 dígitos
				{8} Cadena de 4 dígitos
		R5,R3	{9} Uno de los campos del ID existe.	{10} Uno de los campos del ID no existe.
		R5,R3	{10} El valor N del ID es U	{11} Otro valor.
modUsuario	Usuario	R3	{12} Usuario existente modificado.	{13} Usuario correcto no existente.

2.2.2 Prueba D-02

2.2.2.1 Objetivo

Comprobar que las inserciones o modificaciones de items y ventas se reflejan correctamente en la base de datos.

Además, se busca que, en caso de error, este sea correctamente gestionado.

Dada su similitud con las pruebas para usuarios no se hará una prueba exhaustiva excepto en el caso de que se encuentren fallos en las pruebas anteriores.

2.2.2.2 Técnicas de caja negra

2.2.2.2.1 Generación de clases de equivalencia

Centrándose en los métodos de inserción y modificación de items y ventas y haciendo uso de las reglas especificadas, se definen las siguientes clases:

- **Insertar item / venta / Modificar item**
 - **R3:** Id válido para la inserción y uno inválido.
 - **R3:** Id existente y no existente para la modificación.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

2.2.2.3 Técnicas de caja blanca

2.2.2.4 Criterios de paso/fallo

Los ítems o ventas introducidos o modificados usando datos correctos deben verse reflejados en la base de datos.

En los casos de incluir ítems o ventas de forma errónea debe notificarse del error y en ningún caso reflejarse en la base de datos.

2.2.2.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
insertItem	Item	R3	{1} Id válido.	{2} Id inválido
insertVenta	Venta	R3	{3} Id válido.	{4} Id inválido
updateItem	Item	R3	{7} Id de ítem existente	{6} Id de ítem no existente

2.2.3 Prueba D-03

2.2.3.1 Objetivo

Comprobar que la gestión del login de usuario es correcta.

2.2.3.2 Técnicas de caja negra

2.2.3.2.1 Generación de clases de equivalencia

Solo se proporciona el método isLogged para la gestión del estado de la sesión de usuario.

- **isLogged**
 - **R3:** Id de un usuario existente y un ID de un usuario no existente.
 - **R5:** En base al tipo de ID
 - **R3:** Introducir un ID de un usuario, un ID válido diferente (ítem o venta) y un ID inválido.

2.2.3.3 Técnicas de caja blanca

2.2.3.4 Criterios de paso/fallo

Cuando se introduzca un usuario correcto debe indicarse adecuadamente su estado de sesión. En caso de introducir datos erróneos debe notificarse sobre ello.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

2.2.3.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
isLogged	String	R3	{1} Id de usuario existente	{2} Id de usuario no existente
		R5, R3	{3} Id válido	{4} Id inválido

2.2.4 Prueba D-04

2.2.4.1 Objetivo

Comprobar la correcta validación de un pedido. Para la correcta validación de un pedido este se confirma como compra en la base de datos.

En base a la definición de la interfaz no tenemos forma de seleccionar el pedido a validar ni el usuario que realiza la validación (para comprobar la gestión de las autorizaciones). Por todo esto solo podremos realizar pruebas en función del estado introducido.

2.2.4.2 Técnicas de caja negra

2.2.4.2.1 Generación de clases de equivalencia

Solo el metodo validarPedido gestiona esta acción.

- **validarPedido**
 - **R3:** Cadena válida y cadena vacía.
 - **R4:** Marcar pedido como aceptado, rechazado o una cadena inválida.

2.2.4.3 Técnicas de caja blanca

2.2.4.4 Criterios de paso/fallo

La prueba será satisfactoria en caso de que el pedido pase al estado declarado siempre que el usuario tenga los privilegios para ello.

En caso de que un usuario no esté autorizado deberá indicarse dicho error.

Ninguna entrada incorrecta debe reflejarse en la base de datos.

2.2.4.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
validarPedido	String	R3	{1} Cadena válida	{2} Cadena vacía
		R4	{3} Pedido aceptado	{4} Cadena inválida
			{5} Pedido rechazado	

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

2.2.5 Prueba D-05

2.2.5.1 Objetivo

Comprobar que el historial de compra de los usuarios devuelto se corresponde con el de la base de datos.

2.2.5.2 Técnicas de caja negra

2.2.5.2.1 Generación de clases de equivalencia

Se comprobará el método getHistorialUsuario que devuelve una lista de compras.

- **getHistorialUsuario**
 - **R3:** Id válido de usuario e Id inválido.
 - **R3:** Id de usuario existente e inexistente.

2.2.5.3 Técnicas de caja blanca

2.2.5.4 Criterios de paso/fallo

Siempre que se introduzca un usuario correcto se devuelve un historial correspondiente a dicho usuario. En caso de introducir un usuario incorrecto o cuyo historial de compras es vacío se debe informar de dichos errores.

2.2.5.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
getHistorialUsuario	String	R3	{1} Id válido	{2} Id inválido
		R3	{3} Id de usuario existente	{4} Id de usuario no existente

2.3 Interfaz Estadística

2.3.1 Prueba D-06

2.3.1.1 Objetivo

Comprobar que para entradas incorrectas en los métodos de la interfaz no se producen situaciones no controladas o inesperadas.

2.3.1.2 Técnicas de caja negra

2.3.1.2.1 Generación de clases de equivalencia

Se ha aplicado la siguiente regla a todos los métodos de la interfaz ya que todos reciben la misma entrada: un entero en el rango [1,3] para referenciar respectivamente a días, semanas y meses.

- **getValoresBrutos / getMedias / getHistos / getPorcentajes**
 - **R4:** Introducir cada uno de los modos correctos {1,2,3} y uno incorrecto.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

2.3.1.3 Técnicas de caja blanca

2.3.1.4 Criterios de paso/fallo

Dada una entrada no esperada en un método, ésta no puede provocar un comportamiento no controlado. De ocurrir esto, la consulta realizada se omitirá, devolviendo un mensaje de error indicando que el método no se ha utilizado de manera apropiada.

2.3.1.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
getValoresBrutos / getMedias / getHistos / getPorcentajes	int	R4	{1} Modo 1	{2} N° fuera del rango [1,3]
			{3} Modo 2	
			{4} Modo 3	

2.4 InterfazControlador

2.4.1 Nota

El método *aceptarPedido* es similar a *realizarPedido*, por lo que no se realizarán pruebas exhaustivas sobre este método a menos que se encuentren errores en las pruebas realizadas.

Los métodos *consultarHistorial*, *añadirUsuario*, *modificarUsuario*, *mostrarMedias*, *mostrarHistos*, *mostrarPorcentajes* y *mostrarVentasBruto* cuentan con métodos análogos en otras interfaces. Por tanto, no se realizarán pruebas exhaustivas sobre ellos ya que se considera que estas han sido correctamente realizadas.

2.4.2 Prueba C-01

2.4.2.1 Objetivo

Comprobar que la búsqueda de productos funciona correctamente.

2.4.2.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R3:** introducir una cadena con el identificador de un producto existente y una de uno que no esté registrado pero que cumpla el formato especificado.
- **R5:** los identificadores de productos son de la forma N-XXXXXX-000.
 - **R1:** cadena de letras con 5 y 7 caracteres (incorrecto) y una correcta. Clase equivalente para la cadena numérica.
 - **R3:** eliminar uno de los campos del identificador.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

2.4.2.3 Técnicas de caja blanca

2.4.2.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada. Cuando el argumento sea correcto la búsqueda se efectuará apropiadamente.

2.4.2.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
Producto	String	R3	{1} Identificador existente.	{2} Identificador no existente.
		R5, R1	{3} Cadena de 6 letras para la parte alfabética del ID.	{4} Cadena de 5 letras para la parte alfabética del ID
				{5} Cadena de 7 letras para la parte alfabética del ID
		R5, R1	{6} 3 dígitos para la parte numérica del ID.	{7} 2 dígitos para la parte numérica del ID.
				{8} 4 dígitos para la parte numérica del ID.
		R5, R3	{9} Todos los campos del ID están presentes.	{10} No todos los campos del ID están presentes.

2.4.3 Prueba C-02

2.4.3.1 Objetivo

Comprobar que la inserción de ventas funciona correctamente.

2.4.3.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R3:** la clase *Compra* cuenta únicamente con un atributo de fecha, se introducirá una compra con una fecha válida y una con una fecha inválida para el sistema de compras.

2.4.3.3 Técnicas de caja blanca

2.4.3.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada. Cuando el argumento sea correcto la inserción se efectuará apropiadamente.

2.4.3.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
Fecha	Date	R3	{1} Fecha válida	{2} Fecha no válida

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

2.4.4 Prueba C-03

2.4.4.1 Objetivo

Comprobar que la realización de pedidos funciona correctamente.

2.4.4.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R4:** marcar pedido como aceptado, rechazado o una cadena inválida.

2.4.4.3 Técnicas de caja blanca

2.4.4.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada. Cuando el argumento sea correcto la realización del pedido se efectuará apropiadamente.

2.4.4.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
Estado	String	R4	{1} Estado aceptado	{3} Estado inválido
			{2} Estado rechazado	

2.4.5 Prueba C-04

2.4.5.1 Objetivo

Validar si la comprobación de la existencia de usuarios funciona correctamente.

2.4.5.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R3:** se introducirá el identificador de un usuario registrado y un identificador de un usuario que no esté registrado en la base de datos.

2.4.5.3 Técnicas de caja blanca

2.4.5.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada. Cuando el argumento sea correcto la comprobación se efectuará apropiadamente.

2.4.5.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
ID	String	R3	{1} ID de usuario registrado.	{2} ID de usuario no registrado.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

2.4.6 Prueba C-05

2.4.6.1 Objetivo

Validar si el inicio de sesión de usuarios funciona correctamente.

2.4.6.2 Técnicas de caja negra

Aplicando las reglas especificadas a continuación se obtienen las siguientes clases de equivalencia:

- **R3:** se introducirá el identificador de un usuario registrado y un identificador de un usuario que no esté registrado en la base de datos.

2.4.6.3 Técnicas de caja blanca

2.4.6.4 Criterios de paso/fallo

En ninguno de los casos se debe generar una excepción no controlada. Cuando el argumento sea correcto el inicio de sesión se efectuará apropiadamente.

2.4.6.5 Resultado de aplicar las técnicas

Información	Tipo de dato	Regla	Clase válida	Clase no válida
ID	String	R3	{1} ID de usuario registrado.	{2} ID de usuario no registrado.

3 Casos de prueba

3.1 InterfazImportacion

3.1.1 Caso de prueba I-01-P-01

3.1.1.1 Contexto de ejecución

Valida la correcta interpretación y procesamiento de una línea que contenga todos los campos necesarios con valores correctos. En la base de datos no debe existir un usuario con el identificador *U-aaaaaa-000*.

3.1.1.2 Definición

Se llama al método *importarUsuarios()*

U; U-aaaaaa-000; 10/10/2010; Samuel; Soutullo Sobral; 77013889E

3.1.1.3 Clases que valida

1, 4

3.1.1.4 Resultado esperado

Importación del usuario con ID *U-aaaaaa-000*.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.1.2 Caso de prueba I-01-P-02

3.1.2.1 Contexto de ejecución

Valida la correcta interpretación de una línea que contenga una cantidad de campos inferior a la correcta.

3.1.2.2 Definición

Se llama al método *importarUsuarios()*
U; U-aaaaaa-000; 10/10/2010; Samuel; Soutullo Sobral

3.1.2.3 Clases que valida

2

3.1.2.4 Resultado esperado

Notificación de error por parte de la aplicación, abortando toda la importación. No se modifica la base de datos.

3.1.3 Caso de prueba I-01-P-03

3.1.3.1 Contexto de ejecución

Valida la correcta interpretación de una línea que contenga una cantidad de campos superior a la correcta.

3.1.3.2 Definición

Se llama al método *importarUsuarios()*
U; U-aaaaaa-000; 10/10/2010; Samuel; Soutullo Sobral; 77013889E; asdfg

3.1.3.3 Clases que valida

3

3.1.3.4 Resultado esperado

Notificación de error por parte de la aplicación, abortando toda la importación. No se modifica la base de datos.

3.1.4 Caso de prueba I-01-P-04

3.1.4.1 Contexto de ejecución

Valida la correcta interpretación de una línea en blanco.

3.1.4.2 Definición

Se llama al método *importarUsuarios()*
(Línea en blanco)

3.1.4.3 Clases que valida

5

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.1.4.4 Resultado esperado

No se notifica de ningún error, simplemente se salta la línea en blanco.

3.1.5 Caso de prueba I-02-P-01

3.1.5.1 Contexto de ejecución

Valida la correcta interpretación y procesamiento de una línea que contenga todos los campos necesarios con valores correctos. En la base de datos no debe existir un usuario con el identificador *U-aaaaaa-000*.

3.1.5.2 Definición

Se llama al método *importarUsuarios()*

U; U-aaaaaa-000; 10/10/2010; Samuel; Soutullo Sobral; 77013889E

3.1.5.3 Clases que valida

1, 4, 7, 8, 10, 12

3.1.5.4 Resultado esperado

Importación del usuario con ID *U-aaaaaa-000*.

3.1.6 Caso de prueba I-02-P-02

3.1.6.1 Contexto de ejecución

Valida el correcto comportamiento de la aplicación cuando el método llamado no se corresponde con el tipo de líneas del archivo. En caso de que se detecte un fallo al ejecutar la prueba, se realizará para todas las combinaciones de métodos y tipos de línea.

3.1.6.2 Definición

Se llama al método *importarCompra()*.

U; U-aaaaaa-000; 10/10/2010; Samuel; Soutullo Sobral; 77013889E

3.1.6.3 Clases que valida

14

3.1.6.4 Resultado esperado

La base de datos no se verá modificada dado que el archivo no contenía ninguna línea de venta. Opcionalmente, la aplicación notificará al usuario de la situación que se acaba de dar.

3.1.7 Caso de prueba I-02-P-03

3.1.7.1 Contexto de ejecución

Valida la correcta interpretación y procesamiento de una línea de venta cuando los identificadores de usuario e ítem que ésta referencia ya se encuentran en la base de datos. En la base de datos no existe la venta *V-aaaaaa-000*, pero sí el usuario *U-aaaaaa-000* y el ítem *I-aaaaaa-000*.

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

3.1.7.2 Definición

Se llama al método *importarCompra()*.

V; V-aaaaaa-000; 10/10/2010; U-aaaaaa-000; I-aaaaaa-000; 1; 1.53

3.1.7.3 Clases que valida

16

3.1.7.4 Resultado esperado

Se importará correctamente la nueva venta. La base de datos se verá modificada, de forma que la nueva venta sea insertada, referenciando correctamente al usuario e ítem correspondientes.

3.1.8 Caso de prueba I-02-P-04

3.1.8.1 Contexto de ejecución

Valida la correcta interpretación de una línea que contiene un campo con un tamaño inferior al necesario. Si la prueba falla se realizarán pruebas sobre los campos *apellidos*, *categoría* y *descripción*, validando así a mayores las clases 2, 3, 4, 5 y 6.

3.1.8.2 Definición

Se llama al método *importarUsuarios()*

U; U-aaaaaa-000; 10/10/2010; ; Soutullo Sobral; 77013889E

3.1.8.3 Clases que valida

2

3.1.8.4 Resultado esperado

La aplicación notificará del error correspondiente. La base de datos no se verá modificada.

3.1.9 Caso de prueba I-02-P-05

3.1.9.1 Contexto de ejecución

Valida la correcta interpretación de una línea que contiene un campo con un tamaño superior al necesario. Si la prueba falla se realizarán pruebas sobre los campos *apellidos*, *categoría* y *descripción*, validando así a mayores las clases 2, 3, 4, 5 y 6.

3.1.9.2 Definición

Se llama al método *importarUsuarios()*

U; U-aaaaaa-000; 10/10/2010; aaaaaa[...(x255)...]aaaa ; Soutullo Sobral; 77013889E

3.1.9.3 Clases que valida

3

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

3.1.9.4 Resultado esperado

La aplicación notificará del error correspondiente. La base de datos no se verá modificada.

3.1.10 Caso de prueba I-02-P-06

3.1.10.1 Contexto de ejecución

Valida la correcta interpretación de una línea que contiene un campo de fecha con un formato incorrecto. Si la prueba falla se realizarán pruebas sobre los campos de fecha de todos los tipos de línea.

3.1.10.2 Definición

Se llama al método *importarUsuarios()*

U; U-aaaaaa-000; 10-10-2010; Samuel; Soutullo Sobral; 77013889E

3.1.10.3 Clases que valida

8

3.1.10.4 Resultado esperado

La aplicación notificará del error correspondiente. La base de datos no se verá modificada.

3.1.11 Caso de prueba I-02-P-07

3.1.11.1 Contexto de ejecución

Valida la correcta interpretación de una línea que contiene un campo de precio con un formato incorrecto. Si la prueba falla se realizarán pruebas sobre los campos de unidades y cantidad.

3.1.11.2 Definición

Se llama al método *importarCompra()*

V; V-aaaaaa-000; 10/10/2010; U-aaaaaa-000; I-aaaaaa-000; 1; 125a

3.1.11.3 Clases que valida

11

3.1.11.4 Resultado esperado

La aplicación notificará del error correspondiente. La base de datos no se verá modificada.

3.1.12 Caso de prueba I-02-P-08

3.1.12.1 Contexto de ejecución

Valida la correcta interpretación de una línea cuyo identificador ya se encuentra en la base de datos. El usuario *U-aaaaaa-000* se encuentra previamente en la base de datos. En caso de que la prueba falle, se efectuarán también pruebas para cubrir la clase 15.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.1.12.2 Definición

Se llama al método *importarUsuarios()*

U; U-aaaaaa-000; 10/10/2010; Samuel; Soutullo Sobral; 77013889E

3.1.12.3 Clases que valida

13

3.1.12.4 Resultado esperado

La aplicación notificará del error correspondiente. La base de datos no se verá modificada.

3.1.13 Caso de prueba I-02-P-09

3.1.13.1 Contexto de ejecución

Valida la correcta interpretación de una línea de venta cuando los identificadores de usuario e ítem que ésta referencia no se encuentran en la base de datos.

3.1.13.2 Definición

Se llama al método *importarCompra()*.

V; V-aaaaaa-000; 10/10/2010; U-aaaaaa-000; I-aaaaaa-000; 1; 1.53

3.1.13.3 Clases que valida

17

3.1.13.4 Resultado esperado

La aplicación notificará del error correspondiente. La base de datos no se verá modificada.

3.1.14 Caso de prueba I-02-P-10

3.1.14.1 Contexto de ejecución

Valida la correcta interpretación de un conjunto de líneas de venta cuando dichas líneas tienen el mismo valor para el campo *VRef*. Se presupone que la base de datos contiene el usuario *U-aaaaaa-000* y los productos *I-aaaaaa-000* y *I-aaaaaa-001*.

3.1.14.2 Definición

Se llama al método *importarCompra()*.

V; V-aaaaaa-000; 10/10/2010; U-aaaaaa-000; I-aaaaaa-000; 1; 1.53

V; V-aaaaaa-000; 10/10/2010; U-aaaaaa-000; I-aaaaaa-001; 1; 2.64

3.1.14.3 Clases que valida

Este caso de prueba se ha obtenido en base a conjetura de errores.

3.1.14.4 Resultado esperado

Se añadirá a la base de datos una venta correspondiente al usuario *U-aaaaaa-000*, con los productos *I-aaaaaa-000* y *I-aaaaaa-001*.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.1.15 Caso de prueba I-03-P-01

3.1.15.1 Contexto de ejecución

Valida el correcto comportamiento del módulo de importación cuando se le pasa la ruta de un archivo existente. Se presupone que el fichero */home/usuario/fichero.csv* existe y es un fichero de importación válido. Si la prueba falla, se ejecutará contra todos los métodos de la interfaz.

3.1.15.2 Definición

- Se llama al método *importarUsuario()*
 - **Path:** *"/home/usuario/fichero.csv"*

3.1.15.3 Clases que valida

1, 3

3.1.15.4 Resultado esperado

La importación se realizará correctamente.

3.1.16 Caso de prueba I-03-P-02

3.1.16.1 Contexto de ejecución

Valida el correcto comportamiento del módulo de importación cuando se le pasa la ruta de un archivo con un formato inválido.

3.1.16.2 Definición

- Se llama al método *importarUsuario()*
 - **Path:** *"!·\$%&/()=?¿;"*

3.1.16.3 Clases que valida

2

3.1.16.4 Resultado esperado

Se notificará al usuario del error. No se generará ninguna excepción sin controlar.

3.1.17 Caso de prueba I-03-P-03

3.1.17.1 Contexto de ejecución

Valida el correcto comportamiento del módulo de importación cuando se le pasa la ruta de un archivo que no existe. Se presupone que el fichero *"/home/usuario/ficheroInexistente.csv"* no existe.

3.1.17.2 Definición

- Se llama al método *importarUsuario()*
 - **Path:** *"/home/usuario/ficheroInexistente.csv"*

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.1.17.3 Clases que valida

4

3.1.17.4 Resultado esperado

Se notificará al usuario del error. No se generará ninguna excepción sin controlar.

3.2 InterfazDAO

3.2.1 Caso de prueba D-01-P-01

3.2.1.1 Contexto de ejecución

Valida la inserción correcta de un usuario.

3.2.1.2 Definición

Usuario:

| U-abcdef-000 | Manuel | Soutoullo | 10-10-2010 | alumno |

3.2.1.3 Clases que valida

1,3,6,9,10.

3.2.1.4 Resultado esperado

Inserción del usuario con id: U-abcdef-000 en la base de datos.

3.2.2 Caso de prueba D-01-P-02

3.2.2.1 Contexto de ejecución

Valida la inserción correcta de un usuario.

3.2.2.2 Definición

Usuario:

| x | Manuel | Soutoullo | 10-10-2010 | alumno |

3.2.2.3 Clases que valida

2

3.2.2.4 Resultado esperado

Notificación de error por parte de la aplicación. Sin consecuencias en la base de datos.

3.2.3 Caso de prueba D-01-P-03

3.2.3.1 Contexto de ejecución

Valida la inserción correcta de un usuario.

3.2.3.2 Definición

Usuario:

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

|U-abcde-000 | Manuel | Soutoullo | 10-10-2010 | alumno |

3.2.3.3 Clases que valida

4

3.2.3.4 Resultado esperado

Notificación de error por parte de la aplicación. Sin consecuencias en la base de datos.

3.2.4 Caso de prueba D-01-P-04

3.2.4.1 Contexto de ejecución

Valida la inserción correcta de un usuario.

3.2.4.2 Definición

Usuario:

|U-abcdef-0000 | Manuel | Soutoullo | 10-10-2010 | alumno |

3.2.4.3 Clases que valida

8

3.2.4.4 Resultado esperado

Notificación de error por parte de la aplicación. Sin consecuencias en la base de datos.

3.2.5 Caso de prueba D-01-P-05

3.2.5.1 Contexto de ejecución

Valida la inserción correcta de un usuario.

3.2.5.2 Definición

Usuario:

|U-abcdef-| Manuel | Soutoullo | 10-10-2010 | alumno |

3.2.5.3 Clases que valida

10

3.2.5.4 Resultado esperado

Notificación de error por parte de la aplicación. Sin consecuencias en la base de datos.

3.2.6 Caso de prueba D-01-P-06

3.2.6.1 Contexto de ejecución

Valida la inserción correcta de un usuario.

Se supone que la base de datos solo cuenta con el usuario "U-aaaaaa-000", nombre Limón Novoa, fecha 10-10-2010 y tipo estudiante.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.2.6.2 Definición

Usuario:

|U-aaaaaa-000 | Manuel | Soutoullo | 10-10-2010 | alumno |

3.2.6.3 Clases que valida

12

3.2.6.4 Resultado esperado

En la base de datos el usuario cuyo id es U-aaaaaa-000 pasa a tener el nombre Manuel Soutoullo.

3.2.7 Caso de prueba D-01-P-07

3.2.7.1 Contexto de ejecución

Valida la inserción correcta de un usuario.

Se supone que la base de datos solo cuenta con el usuario "U-aaaaaa-000", nombre Limón Nova, fecha 10-10-2010 y tipo estudiante.

3.2.7.2 Definición

Usuario:

|U-dddddd-111 | Manuel | Soutoullo | 10-10-2010 | alumno |

3.2.7.3 Clases que valida

13

3.2.7.4 Resultado esperado

Se debe indicar que el usuario no existe y, por tanto, no se puede modificar.

Como alternativa puede decidir insertar el nuevo usuario en la base.

3.2.8 Caso de prueba D-02-P-01

3.2.8.1 Contexto de ejecución

Valida la inserción correcta de un ítem.

3.2.8.2 Definición

Ítem:

|I-abcdef-000 | Robot limpiapiscinas | Limpia piscinas | Exteriores | 50
|10/10/2010|

3.2.8.3 Clases que valida

1

3.2.8.4 Resultado esperado

Inserción del ítem con id: I-abcdef-000 en la base de datos.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.2.9 Caso de prueba D-02-P-02

3.2.9.1 Contexto de ejecución

Valida la inserción correcta de una venta.

3.2.9.2 Definición

Venta:

|V-abcdef-000 | 10/10/2010 | 20 |

3.2.9.3 Clases que valida

3

3.2.9.4 Resultado esperado

Inserción de la venta con id: V-abcdef-000 en la base de datos.

3.2.10 Caso de prueba D-02-P-03

3.2.10.1 Contexto de ejecución

Valida la actualización correcta de un item. Suponemos que el único item que existe en la base de datos es |I-abcdef-000 | Robot limpiapiscinas | Limpia piscinas | Exteriores | 50 |10/10/2010|

3.2.10.2 Definición

Item:

|I-abcdef-000 | Robot limpiapiscinas | Limpia piscinas de forma eficiente
| Exteriores | 50 |10-10-2010|

3.2.10.3 Clases que valida

5

3.2.10.4 Resultado esperado

Actualización correcta del item anterior con la descripción actualizada.

3.2.11 Caso de prueba D-02-P-04

3.2.11.1 Contexto de ejecución

Valida la inserción correcta de un item.

3.2.11.2 Definición

Item:

| x | Robot limpiapiscinas | Limpia piscinas de forma eficiente | Exteriores
| 50 |10-10-2010|

3.2.11.3 Clases que valida

2

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.2.11.4 Resultado esperado

Muestra un error y no se pierde la consistencia de la base de datos.

3.2.12 Caso de prueba D-02-P-05

3.2.12.1 Contexto de ejecución

Valida la inserción correcta de una venta.

3.2.12.2 Definición

Venta:

| x | 10/10/2010 | 20 |

3.2.12.3 Clases que valida

4

3.2.12.4 Resultado esperado

Muestra un error y no se pierde la consistencia de la base de datos.

3.2.13 Caso de prueba D-02-P-06

3.2.13.1 Contexto de ejecución

Valida la actualización correcta de un ítem.

Suponemos que el único ítem que existe en la base de datos es |I-abcdef-000 | Robot limpiapiscinas | Limpia piscinas | Exteriores | 50 |10/10/2010|

3.2.13.2 Definición

Ítem:

| I-abcdef-999 | Maletín de cuero | Maletín de portátil | Accesorios | 50 |10-10-2010 |

3.2.13.3 Clases que valida

5

3.2.13.4 Resultado esperado

Muestra un error indicando que el elemento a editar no existe o, alternativamente, inserta en la base el elemento nuevo.

3.2.14 Caso de prueba D-03-P-01

3.2.14.1 Contexto de ejecución

Valida si el usuario introducido está logeado en el sistema.

Suponemos que el único usuario existente en la base tiene el id: I-abcdef-000.

3.2.14.2 Definición

Id usuario:

I-abcdef-000

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.2.14.3 Clases que valida

1,3

3.2.14.4 Resultado esperado

Indica el correcto estado de login del usuario.

3.2.15 Caso de prueba D-03-P-02

3.2.15.1 Contexto de ejecución

Valida si el usuario introducido está logeado en el sistema.

Suponemos que el único usuario existente en la base tiene el id: I-abcdef-000.

3.2.15.2 Definición

Id usuario:

I-abcdef-999

3.2.15.3 Clases que valida

2

3.2.15.4 Resultado esperado

Muestra un error indicando que el usuario no existe. Si simplemente indica que no está logueado se considerará que no pasa la prueba.

3.2.16 Caso de prueba D-03-P-03

3.2.16.1 Contexto de ejecución

Valida si el usuario introducido está logeado en el sistema.

Suponemos que el único usuario existente en la base tiene el id: I-abcdef-000.

3.2.16.2 Definición

Id usuario:

x

3.2.16.3 Clases que valida

4

3.2.16.4 Resultado esperado

Muestra un error indicando que el usuario no existe. Si simplemente indica que no está logueado se considerará que no pasa la prueba.

3.2.17 Caso de prueba D-04-P-01

3.2.17.1 Contexto de ejecución

Valida si el usuario introducido está logeado en el sistema.

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

3.2.17.2 Definición

Estado:

Aceptar

3.2.17.3 Clases que valida

1,3

3.2.17.4 Resultado esperado

El pedido pasa a ser una compra en la base de datos.

3.2.18 Caso de prueba D-04-P-02

3.2.18.1 Contexto de ejecución

Valida si el usuario introducido está logeado en el sistema.

3.2.18.2 Definición

Estado:

Rechazar

3.2.18.3 Clases que valida

5

3.2.18.4 Resultado esperado

El pedido se marca como rechazado o bien se elimina de la base de datos.

3.2.19 Caso de prueba D-04-P-03

3.2.19.1 Contexto de ejecución

Valida si el usuario introducido está logeado en el sistema.

3.2.19.2 Definición

Estado:

""

3.2.19.3 Clases que valida

2

3.2.19.4 Resultado esperado

Se muestra un error indicando que el estado introducido no es correcto.

3.2.20 Caso de prueba D-04-P-04

3.2.20.1 Contexto de ejecución

Valida si el usuario introducido está logeado en el sistema.

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

3.2.20.2 Definición

Estado:

abcd

3.2.20.3 Clases que valida

4

3.2.20.4 Resultado esperado

Se muestra un error indicando que el estado introducido no es correcto.

3.2.21 Caso de prueba D-05-P-01

3.2.21.1 Contexto de ejecución

Comprueba el correcto funcionamiento de la recuperación del historial de compra de un usuario.

El usuario U-abcdef-000 existe y tiene compras en su historial.

3.2.21.2 Definición

Id usuario:

U-abcdef-000

3.2.21.3 Clases que valida

1,3

3.2.21.4 Resultado esperado

Se recupera el historial de compras del usuario.

3.2.22 Caso de prueba D-05-P-02

3.2.22.1 Contexto de ejecución

Comprueba el correcto funcionamiento de la recuperación del historial de compra de un usuario.

El usuario U-abcdef-000 existe y tiene compras en su historial.

3.2.22.2 Definición

Id usuario:

U-a-000

3.2.22.3 Clases que valida

2

3.2.22.4 Resultado esperado

Se muestra un error indicando que el id proporcionado no es válido.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.2.23 Caso de prueba D-05-P-03

3.2.23.1 Contexto de ejecución

Comprueba el correcto funcionamiento de la recuperación del historial de compra de un usuario.

El usuario U-abcdef-000 existe y tiene compras en su historial.

3.2.23.2 Definición

Id usuario:

U-abcdef-999

3.2.23.3 Clases que valida

4

3.2.23.4 Resultado esperado

Se muestra un error indicando que el usuario no existe.

3.2.24 Caso de prueba D-05-P-04

3.2.24.1 Contexto de ejecución

Comprueba el correcto funcionamiento de la recuperación del historial de compra de un usuario.

El usuario U-abcdef-123 existe y **no** tiene compras en su historial.

3.2.24.2 Definición

Id usuario:

U-abcdef-123

3.2.24.3 Clases que valida

Este caso se infiere del estudio por valores límite.

3.2.24.4 Resultado esperado

Se devuelve una lista vacía de compras de forma que sea entendible por el usuario que el historial está vacío.

Si no se indica de ninguna forma que la operación ya ha acabado y que no hay resultados se considera que no pasa la prueba.

3.3 Interfaz Estadística

3.3.1 Nota

Dado que realizando todos los casos de prueba teniendo en cuenta las distintas entradas y métodos de la interfaz serían 16 casos, se ha decidido probar 1 entrada por método. La entrada incorrecta se probará en el método que calcula los valores brutos de ventas debido a su sencillez (es una consulta directa a un elemento de la BBDD sin la realización de ningún tipo de operación).

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

3.3.2 Caso de prueba E-01-P-01

3.3.2.1 Contexto de ejecución

Valida la consulta de ventas totales diarias.

3.3.2.2 Definición

Modo:

-3

3.3.2.3 Clases que valida

2

3.3.2.4 Resultado esperado

El resultado será un mensaje de error indicando que se ha realizado la consulta de manera incorrecta (se ha introducido un modo erróneo).

3.3.3 Caso de prueba E-01-P-02

3.3.3.1 Contexto de ejecución

Valida la consulta de la media de ventas semanales.

3.3.3.2 Definición

Modo:

2

3.3.3.3 Clases que valida

3

3.3.3.4 Resultado esperado

El resultado será la media de ventas semanales.

3.3.4 Caso de prueba E-01-P-03

3.3.4.1 Contexto de ejecución

Valida la consulta del histograma de ventas mensuales.

3.3.4.2 Definición

Modo:

3

3.3.4.3 Clases que valida

4

3.3.4.4 Resultado esperado

El resultado será el histograma de ventas mensuales.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.3.5 Caso de prueba E-01-P-04

3.3.5.1 Contexto de ejecución

Valida la consulta de los porcentajes de las ventas diarias.

3.3.5.2 Definición

Modo:

1

3.3.5.3 Clases que valida

4

3.3.5.4 Resultado esperado

El resultado será el porcentaje correspondiente a las ventas diarias.

3.4 InterfazControlador

3.4.1 Caso de prueba C-01-P-01

3.4.1.1 Contexto de ejecución

Comprobar que la búsqueda de productos funciona correctamente. En la base de datos existe un producto con ID *I-aaaaaa-000*.

3.4.1.2 Definición

Producto:

I-aaaaaa-000

3.4.1.3 Clases que valida

1, 3, 6, 9

3.4.1.4 Resultado esperado

Se recupera el producto con el ID solicitado.

3.4.2 Caso de prueba C-01-P-02

3.4.2.1 Contexto de ejecución

Comprobar que la búsqueda de productos funciona correctamente. En la base de datos no existe ningún producto.

3.4.2.2 Definición

Producto:

I-aaaaaa-000

3.4.2.3 Clases que valida

2

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.4.2.4 Resultado esperado

No se recupera ningún producto.

3.4.3 Caso de prueba C-01-P-03

3.4.3.1 Contexto de ejecución

Comprobar que la búsqueda de productos funciona correctamente. Si la prueba falla se efectuarán pruebas a mayores para cubrir las clases 7 y 8.

3.4.3.2 Definición

Producto:

I-aaaaa-000

3.4.3.3 Clases que valida

4

3.4.3.4 Resultado esperado

No se recupera ningún producto.

3.4.4 Caso de prueba C-01-P-04

3.4.4.1 Contexto de ejecución

Comprobar que la búsqueda de productos funciona correctamente. Si la prueba falla se efectuarán pruebas a mayores para cubrir las clases 7 y 8.

3.4.4.2 Definición

Producto:

I-aaaaaaa-000

3.4.4.3 Clases que valida

5

3.4.4.4 Resultado esperado

No se recupera ningún producto.

3.4.5 Caso de prueba C-01-P-05

3.4.5.1 Contexto de ejecución

Comprobar que la búsqueda de productos funciona correctamente.

3.4.5.2 Definición

Producto:

I-aaaaaa

3.4.5.3 Clases que valida

10

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.4.5.4 Resultado esperado

No se recupera ningún producto.

3.4.6 Caso de prueba C-02-P-01

3.4.6.1 Contexto de ejecución

Comprobar que la inserción de ventas funciona correctamente.

3.4.6.2 Definición

Fecha:

10/10/2010

3.4.6.3 Clases que valida

1

3.4.6.4 Resultado esperado

La compra con la fecha indicada se inserta correctamente en la base de datos.

3.4.7 Caso de prueba C-02-P-02

3.4.7.1 Contexto de ejecución

Comprobar que la inserción de ventas funciona correctamente.

3.4.7.2 Definición

Fecha:

x

3.4.7.3 Clases que valida

2

3.4.7.4 Resultado esperado

Se muestra un error que informa que la fecha es incorrecta y se mantiene el estado anterior de la base de datos.

3.4.8 Caso de prueba C-03-P-01

3.4.8.1 Contexto de ejecución

Comprobar que la realización de pedidos funciona correctamente.

3.4.8.2 Definición

Estado:

Aceptado

3.4.8.3 Clases que valida

1

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.4.8.4 Resultado esperado

Se marca como aceptado en la base de datos el pedido especificado y se declara como compra.

3.4.9 Caso de prueba C-03-P-02

3.4.9.1 Contexto de ejecución

Comprobar que la realización de pedidos funciona correctamente.

3.4.9.2 Definición

Estado:

Rechazado

3.4.9.3 Clases que valida

2

3.4.9.4 Resultado esperado

Se marca como rechazado en la base de datos el pedido especificado y sería válido que el pedido fuese eliminado.

3.4.10 Caso de prueba C-03-P-03

3.4.10.1 Contexto de ejecución

Comprobar que la realización de pedidos funciona correctamente.

3.4.10.2 Definición

Estado:

abcd

3.4.10.3 Clases que valida

3

3.4.10.4 Resultado esperado

Mostrar un error indicando que el estado introducido no es válido. No se debe alterar el estado de la base de datos.

3.4.11 Caso de prueba C-04-P-01

3.4.11.1 Contexto de ejecución

Validar si la comprobación de la existencia de usuarios funciona correctamente. Se supone que el único usuario registrado es U-abcdef-000.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: <i>PDP_v1.doc</i>	

3.4.11.2 Definición

Id usuario:

U-abcdef-000

3.4.11.3 Clases que valida

1

3.4.11.4 Resultado esperado

Se debe indicar que el usuario introducido pertenece al sistema.

3.4.12 Caso de prueba C-04-P-02

3.4.12.1 Contexto de ejecución

Validar si la comprobación de la existencia de usuarios funciona correctamente.

Se supone que el único usuario registrado es U-abcdef-000.

3.4.12.2 Definición

Id usuario:

U-abcdef-999

3.4.12.3 Clases que valida

2

3.4.12.4 Resultado esperado

Se debe indicar que el usuario introducido **no** pertenece al sistema.

3.4.13 Caso de prueba C-05-P-01

3.4.13.1 Contexto de ejecución

Comprobar si el inicio de sesión de usuarios funciona correctamente.

Se supone que el único usuario registrado es U-abcdef-000.

3.4.13.2 Definición

Id usuario:

U-abcdef-000

3.4.13.3 Clases que valida

1

3.4.13.4 Resultado esperado

El usuario pasa a estar logueado en el sistema.

ENSO GrEI	<i>Proceso de Pruebas</i>	01/04/2017
	Doc.: PDP_v1.doc	

3.4.14 Caso de prueba C-05-P-02

3.4.14.1 Contexto de ejecución

Comprobar si el inicio de sesión de usuarios funciona correctamente.
Se supone que el único usuario registrado es U-abcdef-000.

3.4.14.2 Definición

Id usuario:
U-abcdef-999

3.4.14.3 Clases que valida

2

3.4.14.4 Resultado esperado

Se muestra un error indicando que el usuario no está registrado en el sistema y se aborta el inicio de sesión.

4 Procedimientos de prueba

No es necesario seguir ningún orden concreto al ejecutar los casos de prueba, pues todos ellos son idempotentes.

5 Anexo: plantillas

5.1 Plantilla de plan de pruebas

- **Identificador único del documento.**
- **Introducción.**
- **Elementos software a probar:** indica los módulos de software que abarcarán las pruebas definidas.
- **Características a probar:** concreta lo anterior para definir los requisitos funcionales sobre los que se plantean las pruebas.
- **Enfoque general de la prueba:** define el procedimiento seguido a la hora de diseñar las pruebas.
- **Criterios de paso/fallo para cada elemento:** define los criterios que se seguirán para considerar una prueba concreta como superada.
- **Criterios de suspensión y requisitos de reanudación:** define situaciones que implicarán la finalización y replantearse el proceso de pruebas.
- **Documentos a entregar:** índice de los documentos que forman el proceso de pruebas.
- **Actividades de preparación y ejecución de pruebas:** define las actividades necesarias para inicializar el entorno sobre el que se ejecutarán las pruebas.
- **Necesidades de entorno:** indica el contexto de datos necesario para evitar los fallos de configuración a la hora de ejecutar las pruebas.
- **Responsabilidades en la organización y realización de las pruebas:** indica los roles de los integrantes del equipo encargado de realizar las pruebas.

ENSO GrEI	Proceso de Pruebas	01/04/2017
	Doc.: PDP_v1.doc	

- **Necesidades de personal y de formación:** define los conocimientos necesarios por parte de los recursos humanos asociados a la planificación y ejecución de las pruebas.
- **Esquema de tiempos:** diagrama que indica las responsabilidades y tiempos de cada tarea necesaria para la realización del proceso de pruebas.
- **Riesgos asumidos por el plan y planes de contingencia para cada riesgo:** planificación de los riesgos asociados al proceso de pruebas.
- **Aprobaciones y firmas con nombre y puesto desempeñado:** muestra la aprobación de los integrantes acerca de este documento de especificación.

5.2 Plantilla de diseño de pruebas

- **Objetivo:** indica qué funcionalidad busca probar
- **Técnicas de caja negra:** se indica el desarrollo de aplicación de las técnicas de caja negra, separando los métodos a utilizar (generación de clases de equivalencia, análisis de valores límite, conjetura de errores, etc.).
- **Técnicas de caja blanca:** se indica el desarrollo de aplicación de las técnicas de caja blanca.
- **Criterios de paso/fallo:** indica los criterios a valorar para que la prueba se considere como superada.
- **Resultado de aplicar las técnicas:** incluye los anexos que pueden resultar de la aplicación de las técnicas anteriores, como puede ser la tabla de clases de equivalencia, grafos, etc.

5.3 Plantilla de casos de prueba

- **Contexto de ejecución:** se describe el objetivo del caso de prueba y el estado del sistema a la hora de ejecutarse.
- **Definición:** define las entradas y métodos a ejecutar.
- **Clases que valida:** relaciona el caso de prueba con las clases de equivalencia que cubre de la prueba.
- **Resultado esperado:** salida esperada por el programa para que el caso se considere un éxito.