

Plan de pruebas Supermercado UC

Este documento recoge la especificación del plan de pruebas para la aplicación SupermercadoUC,

Contiene las pruebas:

Aceptación – Basadas en la especificación de los casos de uso, se ejecutan de forma manual

Unitarias –

- Capa DAO
- Capa Bussines

Pruebas de Aceptación

En base a los casos de uso se identifican los siguientes escenarios (suponemos que la interfaz por construcción no permite dejar ningún campo requerido en blanco, para evitar casos de prueba referidos a información no disponible):

A1. CU: Realizar Pedido

- a. Pedido realizado con éxito.
- b. Pedido no realizado (Horario indicado no valido).
- c. Error en acceso a base de datos. (Vuelve a resumen carrito)

A1.1 CU : Artículo Añadido

- a. Artículo añadido con éxito.
- b. Artículo no añadido (Stock insuficiente).
- c. Error en acceso a base de datos. (Vuelve a lista articulos)

A1.2 CU : Login Usuario

- a. Login correcto
 - a. Se muestra lista de artículos
 - b. Error acceso a base de datos. (Vuelve a login)
- b. Login incorrecto (Usuario inexistente). (Vuelve a login)
- c. Error en acceso a bse de datos (Vuelve a login)

Pruebas unitarias

Pruebas unitarias de la capa DAO

Para la implementación de las pruebas será necesario el uso de mocks para las entidades Query y EntityManager, se implementarán estos mocks mediante mockito

Clase ArticulosDAO

- Método `articuloConStock(Articulo articulo, int cantidad)`: boolean throws `DataAccesException`

Para la implementación de las pruebas será necesario el uso de mocks para las interfaces DAO correspondiente, se implementarán estos mocks mediante mockito.

UP1.a	<code>Articulo(10,"A",10), 10</code>	True
UP1.b	<code>Articulo(5,"B",5), 10</code>	False
UP1.c	<code>Articulo(100,"F",20), 20</code>	<code>DataAccesException</code>

Pruebas unitarias de la capa Bussines

Clase GestionPedidos

- Método `incluyeArticulo(Articulo articulo, int cantidad)` throws `DataAccessException`, `StockInsuficienteException`

UB1.a	<code>articuloA = Articulo("A",20.0,30), 10</code>	<code>LineaPedido(ArticuloA,10,200.0)</code> (Comprobar que se ha añadido en la lista del componente)
UB1.b	Con <code>LineaPedido</code> de UB1.a en <code>GestionPedidos</code> <code>ArticuloA = Articulo("A",20.0,30),10</code>	<code>LineaPedido(ArticuloA,20,400.0)</code> (Comprobar que se ha actualizado en la lista del componente)
UB1.c	Con <code>LineaPedido</code> de UB1.b en <code>GestionPedidos</code> <code>ArticuloA = Articulo("A",20.0,30),20</code>	<code>LineaPedido(ArticuloA,20,400.0)</code> (Comprobar que no se ha actualizado en la lista del componente) y se lanza excepción <code>StockInsuficienteException</code>
UB1.e	<code>articuloB = Articulo("B",20.0,0), 10</code>	<code>StockInsuficienteException</code>
UB1.f	<code>articuloF = Articulo("F",20.0,10), 10</code>	<code>DataAccessException</code>

Informe Ejecución de las pruebas

Durante la ejecución de las pruebas unitarias de la capa Bussines, se encontraron fallos en las pruebas UB1.b y UB1.c en la actualización de la línea pedido ya existente. Corrigiendo el método para que cumpliera la prueba no se encontraron más fallos.

La ejecución de las pruebas de la capa de persistencia fue satisfactoria.

