# 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

1. Pedido realizado con éxito.
2. Pedido no realizado (Horario indicado no valido).
3. Error en acceso a base de datos. (Vuelve a resumen carrito)

A1.1 CU : Articulo Añadido

1. Articulo añadido con éxito.
2. Articulo no añadido (Stock insuficiente).
3. Error en acceso a base de datos. (Vuelve a lista articulos)

A1.2 CU : Login Usuario

1. Login correcto
   1. Se muestra lista de artículos
   2. Error acceso a base de datos. (Vuelve a login)
2. Login incorrecto (Usuario inexistente). (Vuelve a login)
3. Error en acceso a bse de datos (Vuelve a login)

**Pruebas unitarias**

**Pruebas unitarias de la capa DAO**

**Pruebas unitarias de la capa Bussines**

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

Clase GestionPedidos

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

|  |  |  |
| --- | --- | --- |
| UB1.a | articuloA = Articulo(“A”,20.0,30), 10 | LineaPedido(ArticuloA,10,200.0) (Comprobar que se ha añadido en la lista del componente) |
| UB1.b | articuloA = Articulo”A”,20.0,20), 20 |  |
| UB1.c | articuloB = Articulo(“B”,20.0,0), 10 | StockInsuficienteException |
| UB1.d | articuloF = Articulo(“F”,20.0,10), 10 | DataAccessException |