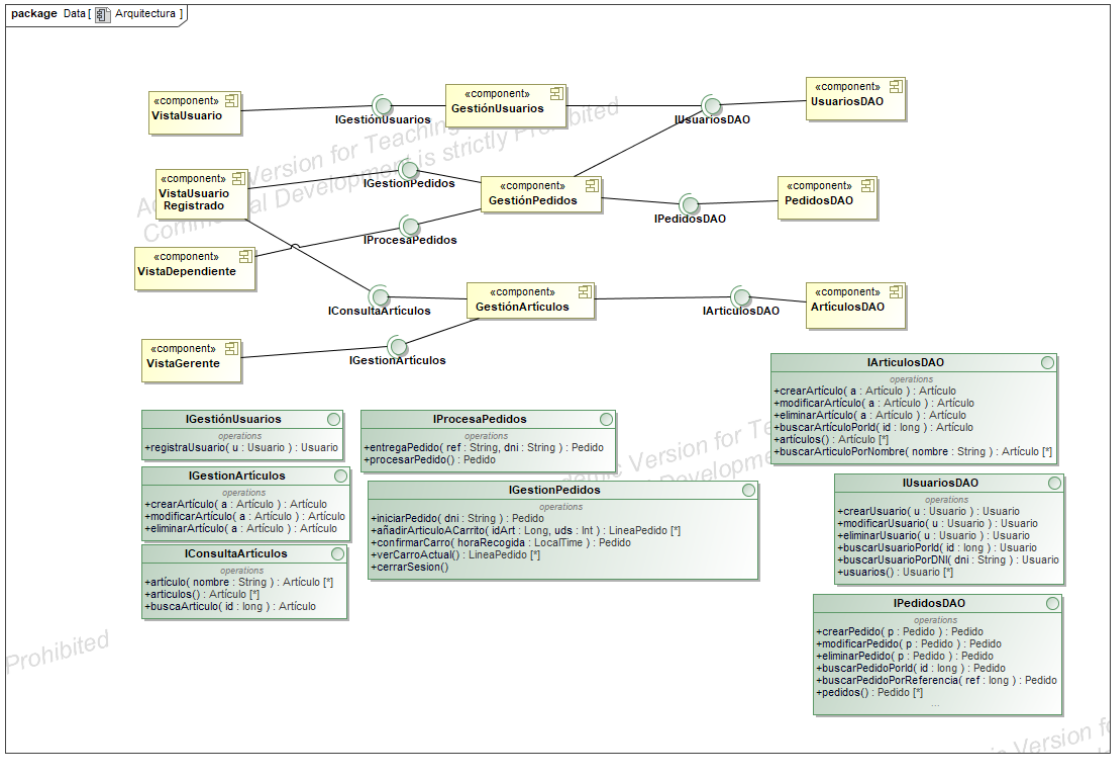
Informe de la arquitectura de la aplicación

Javier Barquín Escalada

Sara Grela Carrera

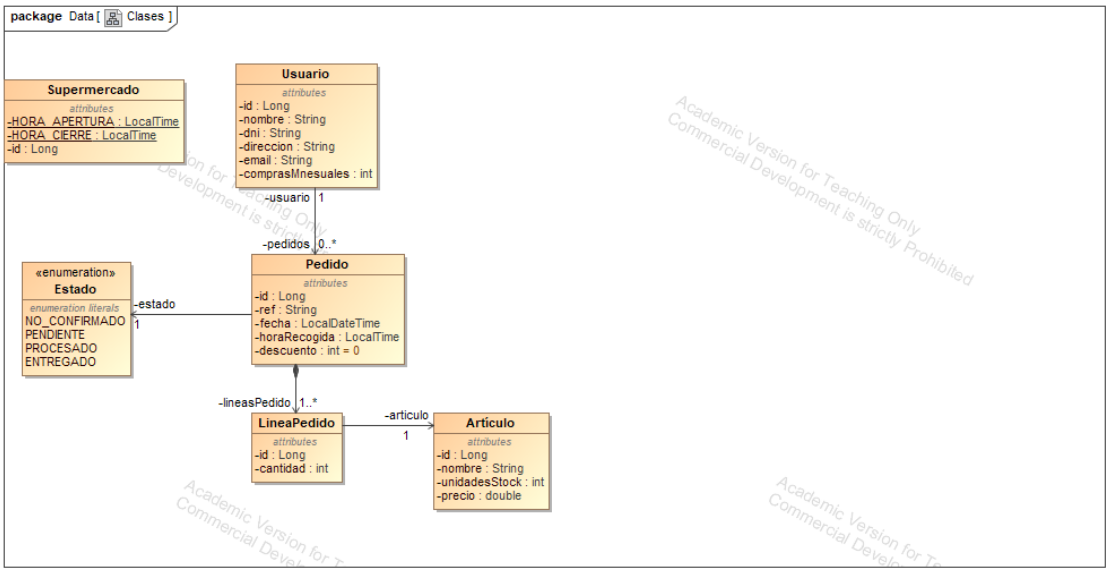
1.- Diseño arquitectónico

Para realizar la separación de la aplicación en componentes, nos hemos basado en la descripción inicial de los casos de uso que se proporcionaba en el enunciado. Para la capa de presentación nos hemos centrado en los posibles usuarios que había (usuario normal y usuario registrado) para hacer una separación de funcionalidades y que ninguno pueda acceder a métodos que no le corresponden y para la capa de negocio, se han agrupado los distintos casos de uso por funcionalidad.



2.- Diseño detallado

Hemos modificado el diagrama inicial aislando la clase “Supermercado”, ya que no se iba a persistir. De este modo, será únicamente una clase para consultar las horas de apertura y cierre. Además, también hemos añadido un identificador único como atributo a cada clase, que nos sirve para realizar búsquedas de instancias concretas.



3.- Plan de pruebas: Historia de Usuario "Realizar pedido"

Se aplicarán los siguientes niveles de prueba:

- Pruebas de aceptación: realizadas de forma manual, con lo cual no necesitan ningún framework o librería concreta.
- Pruebas de integración: para ello se hace uso del framework JUnit y la librería Mockito.
- Pruebas unitarias: se usará JUnit y la librería Mockito.
- Pruebas de interfaz de usuario: utilizando Selenium.

A continuación, se muestra una especificación detallada de los casos de prueba a aplicar en cada nivel mencionado anteriormente.

PRUEBAS DE ACEPTACIÓN

Se identifica el siguiente escenario:

A1: Realizar pedido.

a. Usuario con DNI válido, más de 10 compras mensuales y fecha de recogida válida. Se visualizará la referencia del pedido junto a su total a pagar (5% de descuento) y su hora de recogida.

b. Usuario con DNI válido, menos de 10 compras mensuales y fecha de recogida válida. Se visualizará la referencia del pedido junto a su total a pagar y su hora de recogida.

c. Usuario con DNI válido, menos de 10 compras mensuales y fecha de recogida no válida. Se visualizará un mensaje informando al usuario de que la fecha es errónea.

d. Usuario con DNI no válido. Se visualizará un mensaje informando al usuario de que el DNI es erróneo.

e. Usuario con DNI válido, menos de 10 compras mensuales, fecha de recogida válida y ningún artículo en el carrito. Se visualizará un mensaje indicando que no se ha comprado ningún producto y se redirigirá a la ventana de listado de artículos.

f. Usuario con DNI válido, menos de 10 compras mensuales, fecha de recogida válida y no hay stock suficiente de algún artículo. Se visualizará un mensaje indicando que no hay suficiente stock de un artículo.

PRUEBAS DE INTERFAZ DE USUARIO

Para las pruebas de interfaz de usuario se realizará lo siguiente

UIT1: Realizar pedido válido con descuento. Se corresponde con la prueba de aceptación A1.a:

a. Introducir un DNI válido (de un usuario con más de 10 compras ese mes) y dar a la opción de "entrar".
b. Escoger un artículo y dar a la opción "añadir al carro". Indicar las unidades que se deseen y darle a la opción "añadir a carro".
c. Dar a la opción de "ver carro".
d. Indicar una hora de recogida válida y dar a la opción "confirmar carro".
e. Comprobar que se muestra el número de referencia del pedido, su precio total (con un 5% de descuento) y su hora de recogida correctamente.

UIT2: Realizar pedido válido sin descuento. Se corresponde con la prueba de aceptación A1.b:

f. Introducir un DNI válido (de un usuario con menos de 10 compras ese mes) y dar a la opción de "entrar".
g. Escoger un artículo y dar a la opción "añadir al carro". Indicar las unidades que se deseen y darle a la opción "añadir a carro".
h. Dar a la opción de "seguir comprando".
i. Escoger otro artículo distinto y dar a la opción "añadir al carro". Indicar las unidades que se deseen darle a la opción "añadir a carro".
j. Dar a la opción de "ver carro".
k. Indicar una hora de recogida válida y dar a la opción "confirmar carro".
l. Comprobar que se muestra el número de referencia del pedido, su precio total (sin descuento) y su hora de recogida correctamente.

UIT3: Fecha de recogida no válida. Se corresponde con la prueba de aceptación A1.c:

m. Introducir un DNI válido (de un usuario con menos de 10 compras ese mes) y dar a la opción de "entrar".
n. Escoger un artículo y dar a la opción "añadir al carro". Indicar las unidades que se deseen y darle a la opción "añadir a carro".
o. Dar a la opción de "ver carro".
p. Indicar una hora de recogida no válida y dar a la opción "confirmar carro".
q. Comprobar que se muestra un mensaje indicando que la hora de recogida es errónea.

UIT4: DNI no válido. Se corresponde con la prueba de aceptación A1.d:

r. Introducir un DNI no válido y dar a la opción de "entrar".
s. Comprobar que se muestra un mensaje indicando que el DNI introducido es erróneo.

PRUEBAS DE INTEGRACIÓN

En este caso, solo se pedían las pruebas de integración de un componente de la capa de persistencia con uno de la de negocio, siempre que estuviesen involucrados en el caso de uso 'Realizar pedido'. El orden de las pruebas y los casos de prueba a realizar sería el siguiente:

1. GestiónPedidos (negocio) con PedidosDao(DAO). Se usarían los mismos casos de prueba definidos como UGIC.x en la sección de pruebas unitarias, aquí renombrados como IGIC.x.

PRUEBAS UNITARIAS

- · **Pruebas unitarias de la capa de negocio:**

Se aplica prueba de métodos, siendo los casos de prueba definidos para el método "iniciarPedido(String dni) : Pedido" de la interfaz "IGestionPedidos" los siguientes:

Tabla 1.

Identificador	Entrada	Valor esperado
UGIC.1a	DNI válido (12345678A)	Pedido inicializado (estado = NO_CONFIRMADO y pedido.usuario.dni = 12345678A)
UGIC.1b	DNI no válido (99999999A)	null

Por otro lado, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "añadirArticuloACarrito(Long idArt, int uds): LineaPedido[*]" de la interfaz "IGestionPedidos" los siguientes:

Tabla 2.

Identificador	Entrada	Valor esperado
UGIC.1c	(id artículo existente, 1) Nota: unidades < artículo.stock	Las líneas del pedido que había previamente en el carrito más una nueva con el artículo con ese idArt y las unidades indicadas.
UGIC.1d	(id artículo existente, 100) Nota: unidades > artículo.stock	Retorna Null y no se añade el artículo al carrito.
UGIC.1e	(id artículo existente, 0)	El carrito del pedido no se modifica (sigue habiendo las mismas líneas que había antes).
UGIC.1f	(id artículo no existente, 2)	Retorna Null y las líneas que había ya en el pedido no cambian.
UGIC.1g	(Null, 2)	Retorna Null y las líneas que había ya en el pedido no cambian.
UGIC.1h	(id artículo existente, -1)	Retorna Null, no se añade el artículo y las líneas que había ya en el pedido no cambian.

Después, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "confirmarCarro(LocalTime horaRecogida): Pedido" de la interfaz "IGestionPedidos" los siguientes:

Tabla 3.

Identificador	Entrada	Valor esperado
UGIC.1i	(12:00 am)	Pedido creado en Base de Datos (con identificador)
UGIC.1j	(8:00)	null
UGIC.1k	(22:00)	null

Seguidamente, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "buscarPrimerPedidoPendiente() : Pedido" de la interfaz "IGestionPedidos" los siguientes:

Tabla 4.

Identificador	Entrada	Valor esperado
UGIC.1l	Existen varios pedidos pendientes	Primer pedido (según su fecha) que esté pendiente
UGIC.1m	No existen pedidos pendientes	Null
UGIC.1n	Existe un único pedido pendiente	El único pedido

Finalmente, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "verCarroActual() : LineaPedido[*]" de la interfaz "IGestionPedidos" los siguientes:

Tabla 5.

Identificador	Entrada	Valor esperado
UGIC.1ñ	Existe un pedido vacío	Lista de Líneas de pedido vacía
UGIC.1o	Existe un pedido con un artículo	Lista de Líneas de pedido con el artículo introducido
UGIC.1p	Existe un pedido con varios artículos	Lista de Líneas de pedido con los artículos introducido

- **Pruebas unitarias de la capa de persistencia:**

Se aplica prueba de métodos, siendo los casos de prueba definidos para el método "crearPedido(Pedido p) : Pedido" de la interfaz "PedidosDAO" los siguientes:

Tabla 6.

Identificador	Entrada	Valor esperado
UGIC.1q	(Pedido no existente)	Pedido inicializado (con un id proporcionado por la Base de Datos)
UGIC.1r	(Pedido existente)	null

Por otro lado, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "modificarPedido(Pedido p) : Pedido" de la interfaz "PedidosDAO" los siguientes:

Tabla 7.

Identificador	Entrada	Valor esperado
UGIC.1s	(Pedido no existente)	Pedido creado e inicializado (con un id proporcionado por la Base de Datos)
UGIC.1t	(Pedido existente)	Pedido actualizado

Después, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "eliminarPedido(Pedido p) : Pedido" de la interfaz "PedidosDAO" los siguientes:

Tabla 8.

Identificador	Entrada	Valor esperado
UGIC.1u	(Pedido no existente)	null
UGIC.1v	(Pedido existente)	El mismo pedido, pero ya no está en Base de Datos

Luego se aplica prueba de métodos, siendo los casos de prueba definidos para el método "buscarPedidoPorId(Long id) : Pedido" de la interfaz "PedidosDAO" los siguientes:

Tabla 9.

Identificador	Entrada	Valor esperado
UGIC.1w	(id existente)	Pedido

UGIC.1x	(id no existente)	Null

A continuación, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "buscarPedidoPorReferencia(String ref) : Pedido" de la interfaz "PedidosDAO" los siguientes:

Tabla 10.

Identificador	Entrada	Valor esperado
UGIC.1y	(referencia existente)	Pedido
UGIC.1z	(referencia no existente)	Null

Por último, se aplica prueba de métodos, siendo los casos de prueba definidos para el método "pedidos() : List<Pedido>" de la interfaz "PedidosDAO" los siguientes:

Tabla 11.

Identificador	Entrada	Valor esperado
UGIC.1aa	Existe una lista de pedidos	Lista con pedidos
UGIC.1bb	No existe ningún pedido en la Base de Datos	Lista vacía

PRUEBAS IMPLEMENTADAS

PRUEBAS UNITARIAS:

- **Capa de negocio:**
  Método 'añadirArtículoACarrito(Long idArt, int uds): LineaPedido[*]'.

Tabla 12.

Identificador	Entrada	Valor esperado
UGIC.1c	(1L, 1) Notas: • 1L es idArt de artículo existente • 1 < artículo.stock	Las líneas del pedido que había previamente en el carrito más una nueva con el artículo con ese idArt y las unidades indicadas.

UGIC.1d	(2L, 100) Notas: • 2L es idArt de artículo existente • 100 > artículo.stock	Retorna Null, no se añade el artículo y las líneas que había ya en el pedido no cambian.
UGIC.1e	(2L, 0) Notas: • 2L es idArt de artículo existente	Retorna Null, no se añade el artículo y las líneas que había ya en el pedido no cambian.
UGIC.1f	(0L, 2) Notas: • 0L es idArt de artículo NO existente.	Retorna Null y las líneas que había ya en el pedido no cambian.
UGIC.1g	(Null, 2)	Retorna Null y las líneas que había ya en el pedido no cambian.
UGIC.1h	(2L, -1) • 2L es idArt de artículo existente	Retorna Null, no se añade el artículo y las líneas que había ya en el pedido no cambian.

- **Capa de persistencia:**
  Método buscarPedidoPorReferencia(String ref) : Pedido'.

Tabla 13.

Identificador	Entrada	Resultado
UGIC.1t	Existe una lista de pedidos pendientes	Primer pedido de la lista
UGIC.1u	No existe ningún pedido pendiente	Lista vacía

PRUEBAS DE INTEGRACIÓN:

Tabla 14.

Identificador	Entrada	Resultado
IGIC.1c	(1L, 1) Notas: • 1L es idArt de artículo existente • 1 < artículo.stock	Las líneas del pedido que había previamente en el carrito más una nueva con el artículo con ese idArt y las unidades indicadas.

IGIC.1d	(1L, 100) Notas: • 1L es idArt de artículo existente • 100 > artículo.stock	Retorna Null y no se añade el artículo al carrito.
IGIC.1e	(2L, 0) Notas: • 2L es idArt de artículo existente	El carrito del pedido no se modifica (sigue habiendo las mismas líneas que había antes).
IGIC.1f	(0L, 2) Notas: • 0L es idArt de artículo NO existente	Retorna Null y las líneas que había ya en el pedido no cambian.
IGIC.1g	(Null, 2)	Retorna Null y las líneas que había ya en el pedido no cambian.
IGIC.1h	(2L, -1) • 2L es idArt de artículo existente	Retorna Null, no se añade el artículo y las líneas que había ya en el pedido no cambian.

PRUEBAS DE ACEPTACIÓN:

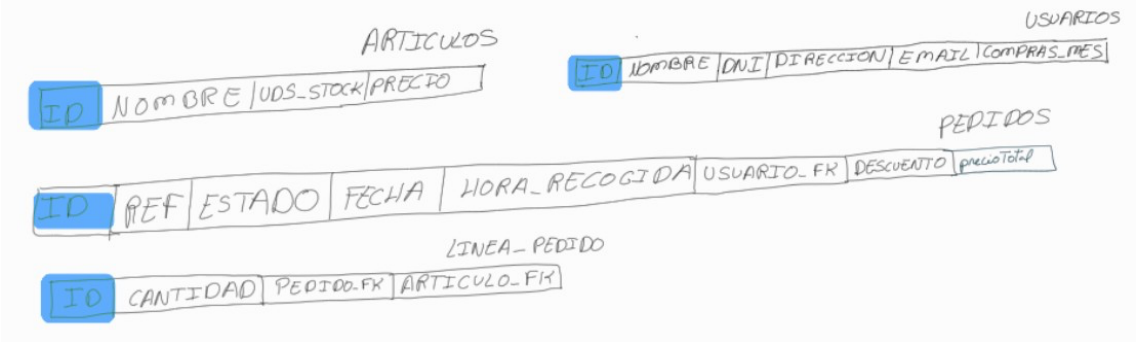
Han sido automatizadas con la herramienta Selenium.

Tabla 15.

Identificador	**Entrada**	**Resultado**
UIT2	DNI existente introducido: 12345678A Primer artículo seleccionado: “Manzana Golden” y unidades “2” Segundo artículo seleccionado: “Agua de Solares 1L” y unidades “1” Hora de recogida introducida: 12:49	Pedido realizado correctamente

4.- Extra: diseño de la base de datos

Adjuntamos el diseño que decidimos para persistir las clases.


ARTICULOS
ID
NOMBRE
UDS_STOCK
PRECIO
USUARIOS
ID
NOMBRE
DNI
DIRECCION
EMAIL
COMPRAS_MES
PEDIDOS
ID
REF
ESTADO
FECHA
HORA_RECOGIDA
USUARIO_FK
DESCUENTO
precioTotal
LINEA_PEDIDO
ID
CANTIDAD
PEDIDO_FK
ARTICULO_FK