



# Ingeniería del Software 2

## Memoria

2020 - 2021

MARIO CAMPOS SOBRINO

IGNACIO FLUXÁ CASAS

GUILLERMO GASCÓN CELDRÁN

ÁNGEL HORTELANO PÉREZ

AIRAM MARTÍN PERAZA

ANA MARÍA MARTÍNEZ VALDÉS

JORGE MORALES LÓPEZ

PATRICIA PASTOR PUENTE

# Índice

<b>1.- Introducción</b>	<b>2</b>
<b>2.- Herramientas y tecnologías usadas</b>	<b>2</b>
<b>3.- Descripción detallada del proyecto</b>	<b>3</b>
3.1.- Documentación	3
3.2.- Diagramas	3
3.3.- Pruebas	4
3.4.- Patrones	4

# 1.- Introducción

La aplicación *FootWorld* pretende gestionar la compra, la venta y el inventario de productos de una zapatería. De esta forma se tiene toda la información bien almacenada y ordenada de forma que es muy sencillo acceder a cualquier dato, tales como: información acerca de un producto, clientes, trabajadores, proveedores y cómo se han ido vendiendo los productos a lo largo del tiempo.

En la aplicación gestionaremos las siguientes entidades:

- Clientes
- Marcas
- Productos
- Proveedores
- Almacenes
- Trabajadores
- Ventas

## 2.- Herramientas y tecnologías usadas

La aplicación será un programa de escritorio que tendrá una interfaz gráfica y se conectará a una base de datos para recoger y almacenar información.

En cuanto a las tecnologías relativas a la aplicación hemos usado:

- Java, como lenguaje de programación, en su versión JDK 1.8
- MySQL, como base de datos, en su versión 8.

Para el desarrollo del proyecto hemos usado:

- IBM RSAD, como herramienta de diseño.
- Eclipse, como entorno de desarrollo.
- GitHub y SVN, como software de control.
- Google Docs, como herramienta para elaborar documentos.

## 3.- Descripción detallada del proyecto

### 3.1.- Documentación

Para tener bien reflejado el diseño de la aplicación y a modo de documentación, hemos realizado dos tipos de diagramas que reflejan su la estructura y el comportamiento de la misma:

Diagramas de clases, hechos con el lenguaje UML y donde se definen las clases por las que está compuesta la aplicación y su relación entre ellas.

Diagramas de secuencia, encapsulan el comportamiento de las diferentes partes de la aplicación mostrando la interacción entre dichas partes.

### 3.2.- Diagramas

Como hemos comentado anteriormente, hemos realizado diagramas de clase y secuencia. Las siguientes entidades disponen de dichos diagramas:

#### **Diagramas de clase**

- Almacén
- Cliente
- Producto
- Proveedor-Producto
- Venta
- Marca
- Trabajador
- Proveedor

#### **Diagramas de secuencia**

- Presentación
  - Almacén
  - Cliente
  - Marca
  - Producto
  - Venta
  - Proveedor
  - Trabajador
- Negocio
  - Almacén
  - Cliente
  - Producto
  - Proveedor-Producto
  - Venta

- Integración
  - Cliente
  - Producto
  - Venta

### 3.3.- Pruebas

Una vez terminada la aplicación, escribimos una serie de Unit Test para probar la aplicación en su totalidad. Para ello usamos el conocido framework de Java, JUnit, en su versión 5.

También se han realizado pruebas en la interfaz gráfica siguiendo todos los flujos posibles de interacción.

### 3.4.- Patrones

A continuación se muestra un listado de los patrones usados en la aplicación.

- 
- Controller, para encapsular el comportamiento de la aplicación en un solo objeto.
- Singleton, para evitar crear múltiples instancias de una misma clase.
- Abstract Factory, para generar las instancias necesarias por demanda.
- Transfer, actúa como un objeto que encapsula todos los datos necesarios.
- TOA, es un conjunto de transfers.
- Application Service, SA, usado en la capa de negocio.
- Data Access Object, DAO, usado en la capa que interactúa con la base de datos.