

# Bodegas & Viñedos

Adrián Álvarez Pérez - Diego Fernández Pérez

Desarrollo de Aplicaciones Web | Curso 2022-2023

**edix**



# Índice

⇒ <b>1.0 Portada</b>	Pág. 01
⇒ <b>2.0 Introducción</b>	Pág. 04
⇒ <b>3.0 Módulos formativos aplicados en el TFG</b>	Pág. 05
⇒ <b>4.0 Herramientas/Lenguajes utilizados</b>	Pág. 06
⇒ <b>5.0 Componentes del equipo y aportación realizada por cada uno</b>	Pág. 08
⇒ <b>6.0 Fases del proyecto</b>	Pág. 09
■ <b>6.1 Modelo de datos</b>	Pág. 09
• 6.1.1 Estructura	Pág. 10
• 6.1.2 Creación de tablas, atributos y inserción de datos	Pág. 11
■ <b>6.2 Diagrama de clases y casos de uso</b>	Pág. 14
• 6.2.1 Diagrama de clases	Pág. 15
• 6.2.2 Diagrama de casos de uso	Pág. 18
■ <b>6.3 Planificación del desarrollo</b>	Pág. 19
■ <b>6.4 Explicaciones de la funcionalidad del proyecto</b>	Pág. 20
■ <b>6.5 Design Thinking</b>	Pág. 26
• 6.5.1 Empatizar	Pág. 26
○ Investigación documental	Pág. 26
○ Benchmarking UI	Pág. 27
○ Perfiles encontrados	Pág. 28
• 6.5.2 Definir	Pág. 29
○ User Persona	Pág. 29
• 6.5.3 Idear	Pág. 30
○ How Might We	Pág. 30
○ Brainstorming	Pág. 31
• 6.5.4 Prototipar parte 1	Pág. 32
○ User Story Mapping	Pág. 32
○ Site Map	Pág. 33
• 6.5.5 Prototipar parte 2	Pág. 34
○ Cover	Pág. 34
○ Wireframes	Pág. 35

- 
- Guía de estilos \_\_\_\_\_ Pág. 36
  - Mockups \_\_\_\_\_ Pág. 37

⇒ **7.0 Conclusiones y mejoras del proyecto** \_\_\_\_\_ Pág. 38

⇒ **8.0 Bibliografía** \_\_\_\_\_ Pág. 49

⇒ **9.0 Anexos** \_\_\_\_\_ Pág. 40

## ⇒ 2.0 - Introducción

El proyecto está orientado a una página web basada en las bodegas y sus vinos, ya que, el vino es un producto local muy valorado tanto para los residentes de la península como para el cliente extranjero. En un principio, una bodega no suele contar con un equipo informático que le realice/mantenga un servicio web.

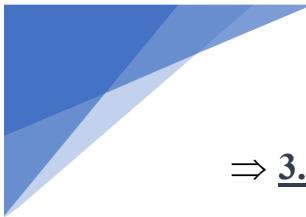
En España existen cientos de miles de bodegas de distintas comunidades autónomas junto con sus diversas denominaciones de origen.

El público que tendrá esta página web estará en el rango de edad entre los 35 años y los 60 años. Este rango de edad es el que más vino suele consumir.

Aunque se crearan campañas de marketing digital para animar a más edades a consumir nuestros grandes vinos nacionales.

Creemos que con esta web podemos crear un modelo/varios modelos (para un futuro) y ofrecer diferentes plantillas a modificar a gusto del cliente o bien crear una desde los cimientos. Es verdad, que en un principio sería difícil ganarse un nombre en este sector, pero si a la larga lo consiguiéramos podríamos llegar a tener muchos clientes potenciales, en el caso de que en la península no encontrásemos dichos clientes siempre podemos exportar nuestro producto al resto del continente europeo adaptándolo a los distintos productos típicos de cada país como podría ser el champagne en Francia o bien la cerveza en Alemania...

En un futuro siempre podemos optar también a realizar páginas con el mismo sistema para otros productos gourmets/exclusivos (Jamón, queso, sobrasada...) de pequeñas empresas que tienen un producto local muy bueno pero que no se han adaptado al mundo tecnológico actual, por lo tanto, creemos que tenemos un nicho de mercado altamente potencial.



## ⇒ 3.0 - Módulos formativos aplicados en el TFG

- **Diseño de interfaces web**
  - Apoyándonos en esta asignatura, nos ayudará a hacer una gran investigación sobre el usuario y la competencia. También nos ayudará a hacer una maquetación sobre cómo será la web, sus colores, imágenes, formularios etc.
- **Desarrollo entorno cliente**
  - Apoyándonos en esta asignatura, nos ayudará a que el frontend sea dinámico gracias a javascript y el lenguaje de marcado HTM5 y el de estilos CCS3 y bootstrap.
- **Desarrollo entorno servidor**
  - Apoyándonos en esta asignatura, nos ayudará a que el backend sea más fácil y rápido de hacer y encima tenga seguridad y todo esto gracias al framework Spring Boot y el servidor de aplicaciones apache Tomcat.
- **Lenguaje de marcas**
  - Apoyándonos en esta asignatura, nos ayudará a utilizar el lenguaje de marcado HTML5 y el lenguaje de diseño de la web CSS3.
- **Base de datos**
  - Apoyándonos en esta asignatura, nos ayudará a que MySQL Server sea el gestor de base de datos y la interfaz gráfica para crear la base de datos, tablas, campos y datos y luego verlos gráficamente será MySQL Workbench.
- **Programación**
  - Apoyándonos en esta asignatura, nos ayudara a que Java sea el lenguaje en el backend.
- **Entornos de desarrollo**
  - Apoyándonos en esta asignatura, nos ayudara a usar Git, GitHub y a realizar el diagrama de clases y casos de uso.

## ⇒ 4.0 - Herramientas/Lenguajes utilizados

- Para la maquetación de la web usaremos:

- Figma: programa para maquetar



- Técnica Design Thinking: basada en el usuario



- User Experience y User Interfaz



- Frontend:

- Bootstrap



- HTML5 | CSS3 | Javascript



- Backend:

- Lenguaje Java



- Framework Spring Boot MVC



- Acceso a la base de datos mediante Spring Data JPA



- Spring Security



- **IDE:**

- Eclipse: programa para hacer el backend con Spring Boot



- Visual Studio Code: programa para hacer el frontend con angular



- **Servidor de aplicaciones:**

- Apache Tomcat



- **Base de datos:**

- MySQL server 8.33



- **Interfaz gráfica de la base de datos:**

- MySQL Workbench 8.33



- **Control de versiones:**

- Git



- **Respositorio en GitHub:**

- GitHub



## ⇒ **5.0 - Componentes del equipo y aportación realizada por cada uno**



**Adrián Álvarez Pérez**



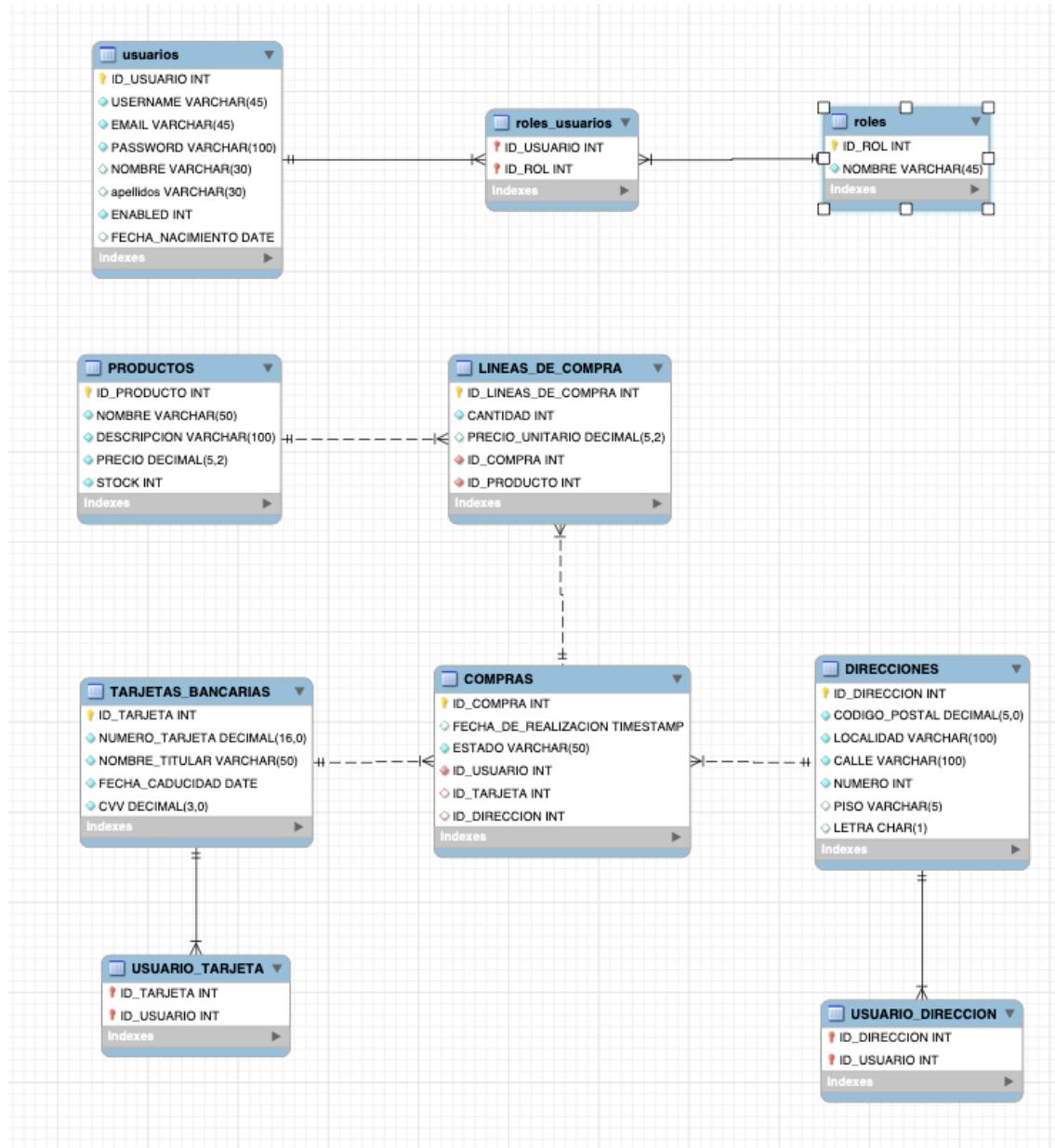
**Diego Fernández Pérez**

- Ambos integrantes han participado en todas las fases del proyecto.
- Dependiendo de la fase en la que se encontraba el proyecto por ejemplo el frontend, Diego participo en más porcentaje y en la parte de código del backend Adrián participo en más porcentaje, pero ambos estuvieron en ambas fases. Tanto para frontend como para backend se usó el control de versiones Git junto con el repositorio de GitHub.
- En el caso del Design Thinking ambos integrantes participaron por igual gracias a la comodidad que aporta el programa Figma en la que ambos pueden trabajar en tiempo real en la maquetación.
- En cuanto a la base de datos ambos integrantes participaron por igual realizando el script de la base de datos creando las tablas, atributos y rellenando el script de datos.
- Desde que comenzamos el proyecto, agendamos de lunes a viernes de 19:00 a 21:00 para dedicarle al proyecto y sábados, domingo y festivos prácticamente todo el día.
- Para toda la realización del proyecto se ha utilizado la herramienta de Discord para la realización de videollamada y compartir escritorio.

## ⇒ 6.0 - Fases del proyecto

- **Modelo de datos** → base de datos relacional basada en tablas, atributos(campos) y datos metidos en filas y columnas.

- Estructura



- Creación de tablas, atributos e inserción de datos

- Con el **create table** creamos las tablas y dentro creamos los campos.

- Con el **insert** metemos los datos a esos campos.

```
5   -----  
6   ----- TABLA_ROLES -----  
7  
8 • CREATE TABLE IF NOT EXISTS `Tienda`.`roles`(  
9     `ID_ROL` INT NOT NULL AUTO_INCREMENT,  
10    `NOMBRE` VARCHAR(45) NOT NULL,  
11  
12    PRIMARY KEY (`ID_ROL`)  
13 )  
14 ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARACTER SET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;  
15  
16 -- INSERTAMOS DATOS A LA TABLA DE ROLES  
17 • INSERT INTO `roles`(`ID_ROL`, `NOMBRE`)  
18     VALUES (1, 'ROLE_ADMINISTRADOR');  
19  
20 • INSERT INTO `roles`(`ID_ROL`, `NOMBRE`)  
21     VALUES (2, 'ROLE_CLIENTE');  
22  
23  
24 ----- TABLA_DIRECCIONES -----  
25  
26  
27 • CREATE TABLE DIRECCIONES(  
28     ID_DIRECCION INT AUTO_INCREMENT NOT NULL,  
29     CODIGO_POSTAL DEC(5) NOT NULL,  
30     LOCALIDAD VARCHAR(100) NOT NULL,  
31     CALLE VARCHAR(100) NOT NULL,  
32     NUMERO INT NOT NULL,  
33     PISO VARCHAR(5) NULL,  
34     LETRA CHAR NULL,  
35  
36     PRIMARY KEY (ID_DIRECCION)  
37 );
```

```

40
41 -----TABLA_TARJETAS_BANCARIAS-----
42
43 • CREATE TABLE TARJETAS_BANCARIAS(
44     ID_TARJETA INT AUTO_INCREMENT NOT NULL,
45     NUMERO_TARJETA DEC(16) NOT NULL,
46     NOMBRE_TITULAR VARCHAR(50) NOT NULL,
47     FECHA_CADUCIDAD DATE NOT NULL,
48     CVV DEC(3) NOT NULL,
49
50     PRIMARY KEY(ID_TARJETA)
51 );
52
53 -----TABLA_PRODUCTOS-----
54
55 CREATE TABLE PRODUCTOS (
56     ID_PRODUCTO INT AUTO_INCREMENT NOT NULL,
57     NOMBRE VARCHAR(50) UNIQUE NOT NULL,
58     DESCRIPCION VARCHAR(100) NOT NULL,
59     PRECIO DEC(5,2) NOT NULL,
60     STOCK INT NOT NULL,
61
62     PRIMARY KEY(ID_PRODUCTO)
63 );
64
65 -- INSERTAMOS DATOS A LA TABLA DE PRODUCTOS
66 INSERT INTO PRODUCTOS
67     VALUES(1, "Vino rosado", "Vino rosado", 29.95, 10);
68
69 INSERT INTO PRODUCTOS
70     VALUES(2, "Vino godello", "Vino godello", 19.95, 30);
71
72 INSERT INTO PRODUCTOS
73     VALUES(3, "Vino tinto", "Vino tinto", 4.95, 100);
74

```

```

-----TABLA_USUARIOS-----
CREATE TABLE IF NOT EXISTS `Tienda`.`usuarios` (
    `ID_USUARIO` int NOT NULL AUTO_INCREMENT,
    `USERNAME` VARCHAR(45) UNIQUE NOT NULL,
    `EMAIL` VARCHAR(45) UNIQUE NOT NULL,
    `PASSWORD` VARCHAR(100) NOT NULL,
    `NOMBRE` VARCHAR(30) NULL DEFAULT NULL,
    `apellidos` VARCHAR(30) NULL DEFAULT NULL,
    `ENABLED` INT NOT NULL DEFAULT '1',
    `FECHA_NACIMIENTO` DATE NULL DEFAULT NULL,
    PRIMARY KEY (`ID_USUARIO`)
)
ENGINE=InnoDB  DEFAULT CHARACTER SET=utf8mb4  COLLATE=utf8mb4_0900_ai_ci;

-- INSERTAMOS DATOS A LA TABLA USUARIOS ROLE ADMIN --> contraseña 12345
INSERT INTO `usuarios` (`EMAIL`, `USERNAME`, `PASSWORD`, `NOMBRE`, `apellidos`, `ENABLED`, `FECHA_NACIMIENTO`)
VALUES ('adrian@gmail.com', 'adrian', '$2a$10$pqMraQkIyQHtNj4c9RH8Qui/fxABpK8BW4r39mnUcFT7J6UtuzVzW', 'Adrian', 'Alvarez', 1, '1991-01-01');

INSERT INTO `usuarios` (`EMAIL`, `USERNAME`, `PASSWORD`, `NOMBRE`, `apellidos`, `ENABLED`, `FECHA_NACIMIENTO`)
VALUES ('diego@gmail.com', 'diego', '$2a$10$pqMraQkIyQHtNj4c9RH8Qui/fxABpK8BW4r39mnUcFT7J6UtuzVzW', 'Diego', 'Perez', 1, '1994-01-01');

-- INSERTAMOS DATOS A LA TABLA USUARIOS ROLE CLIENTE --> contraseña 12345
INSERT INTO `usuarios` (`EMAIL`, `USERNAME`, `PASSWORD`, `NOMBRE`, `apellidos`, `ENABLED`, `FECHA_NACIMIENTO`)
VALUES ('adrian@gmail.com', 'adri', '$2a$10$pqMraQkIyQHtNj4c9RH8Qui/fxABpK8BW4r39mnUcFT7J6UtuzVzW', 'Adri', 'Alvarez', 1, '1991-01-01');

INSERT INTO `usuarios` (`EMAIL`, `USERNAME`, `PASSWORD`, `NOMBRE`, `apellidos`, `ENABLED`, `FECHA_NACIMIENTO`)
VALUES ('diego@gmail.com', 'die', '$2a$10$pqMraQkIyQHtNj4c9RH8Qui/fxABpK8BW4r39mnUcFT7J6UtuzVzW', 'Die', 'Perez', 1, '1994-01-01');

```

```

-----TABLA_COMPRA-----
```

```

CREATE TABLE COMPRAS (
    ID_COMPRA INT AUTO_INCREMENT NOT NULL,
    FECHA_DE_REALIZACION TIMESTAMP DEFAULT NOW(),
    ESTADO VARCHAR(50) NOT NULL,
    PRIMARY KEY(ID_COMPRA),
    ID_USUARIO INT NOT NULL,
    ID_TARJETA INT,
    ID_DIRECCION INT,
    FOREIGN KEY(ID_USUARIO) REFERENCES USUARIOS (ID_USUARIO),
    FOREIGN KEY(ID_TARJETA) REFERENCES TARJETAS_BANCARIAS (ID_TARJETA),
    FOREIGN KEY(ID_DIRECCION) REFERENCES DIRECCIONES (ID_DIRECCION)
);

-----TABLA_LINEA_DE_COMPRA-----
```

```

CREATE TABLE LINEAS_DE_COMPRA (
    ID_LINEAS_DE_COMPRA INT AUTO_INCREMENT NOT NULL,
    CANTIDAD INT NOT NULL,
    PRECIO_UNITARIO DEC(5,2),
    PRIMARY KEY(ID_LINEAS_DE_COMPRA),
    ID_COMPRA INT NOT NULL,
    ID_PRODUCTO INT NOT NULL,
    FOREIGN KEY(ID_COMPRA) REFERENCES COMPRAS (ID_COMPRA),
    FOREIGN KEY(ID_PRODUCTO) REFERENCES PRODUCTOS (ID_PRODUCTO)
);

```

```
-- -----TABLAS_RENACIDAS-----


CREATE TABLE USUARIO_DIRECCION(
    ID_DIRECCION INT NOT NULL,
    ID_USUARIO INT NOT NULL,
    PRIMARY KEY(ID_DIRECCION, ID_USUARIO),
    FOREIGN KEY(ID_DIRECCION) REFERENCES DIRECCIONES (ID_DIRECCION),
    FOREIGN KEY(ID_USUARIO) REFERENCES USUARIOS (ID_USUARIO)
);

CREATE TABLE USUARIO_TARJETA(
    ID_TARJETA INT NOT NULL,
    ID_USUARIO INT NOT NULL,
    PRIMARY KEY(ID_TARJETA, ID_USUARIO),
    FOREIGN KEY(ID_TARJETA) REFERENCES TARJETAS_BANCARIAS (ID_TARJETA),
    FOREIGN KEY(ID_USUARIO) REFERENCES USUARIOS (ID_USUARIO)
);

CREATE TABLE `roles_usuarios` (
    `ID_USUARIO` INT NOT NULL,
    `ID_ROL` INT NOT NULL,
    PRIMARY KEY (`ID_USUARIO`, `ID_ROL`),
    KEY `ID_ROL` (`ID_ROL`),
    CONSTRAINT `roles_usuarios_ibfk_1` FOREIGN KEY (`ID_USUARIO`) REFERENCES `usuarios` (`ID_USUARIO`),
    CONSTRAINT `roles_usuarios_ibfk_2` FOREIGN KEY (`ID_ROL`) REFERENCES `roles` (`ID_ROL`)
)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

#### -- INSERTAMOS DATOS A LOS ROLES\_USUARIOS

```
INSERT INTO `roles_usuarios`(`ID_USUARIO`, `ID_ROL`)
VALUES (1,1);
```

```
INSERT INTO `roles_usuarios`(`ID_USUARIO`, `ID_ROL`)
VALUES (2,1);
```

```
INSERT INTO `roles_usuarios`(`ID_USUARIO`, `ID_ROL`)
VALUES (3,2);
```

```
INSERT INTO `roles_usuarios`(`ID_USUARIO`, `ID_ROL`)
VALUES (4,2);
```

## ■ Diagrama de clases y casos de uso

### ○ Diagrama de clases

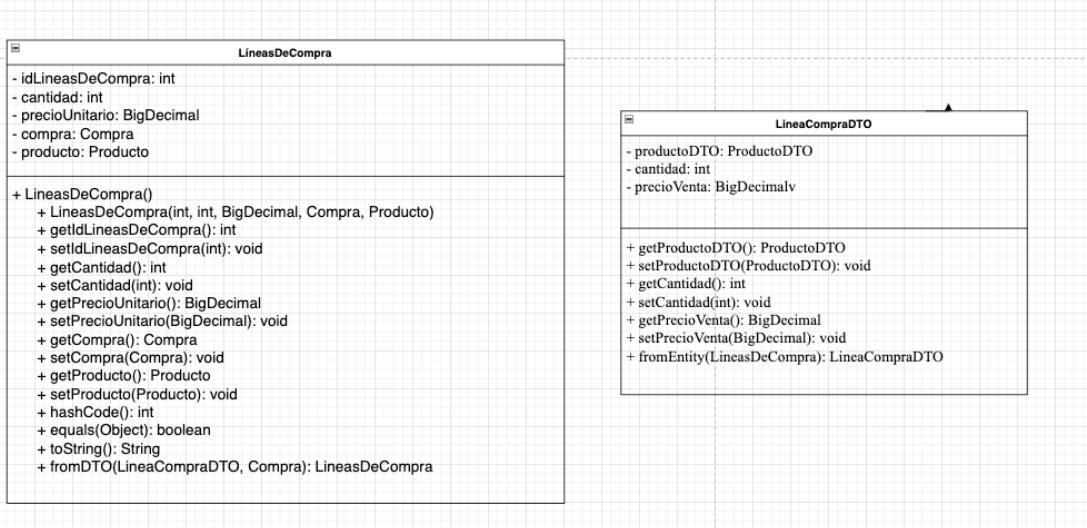


Producto	
- idProducto: int	
- descripcion: String	
- nombre: String	
- precio: BigDecimal	
- stock: int	
- lineasDeCompras: List<LineasDeCompra>	
+ getIdProducto(): int	
+ setIdProducto(int): void	
+ getDescripcion(): String	
+ setDescripcion(String): void	
+ getNombre(): String	
+ setNombre(String): void	
+ getPrecio(): BigDecimal	
+ setPrecio(BigDecimal): void	
+ getStock(): int	
+ setStock(int): void	
+ getLineasDeCompras(): List<LineasDeCompra>	
+ setLineasDeCompras(List<LineasDeCompra>): void	
+ addLineasDeCompra(LineasDeCompra): LineasDeCompra	
+ removeLineasDeCompra(LineasDeCompra): LineasDeCompra	
+ hashCode(): int	
+ equals(Object): boolean	
+ toString(): String	
+ fromDTO(ProductoDTO): Producto	

ProductoDTO	
- idProducto: int	
- descripcion: String	
- nombre: String	
- precio: BigDecimal	
- stock: int	
+ getIdProducto(): int	
+ setIdProducto(int): void	
+ getDescripcion(): String	
+ setDescripcion(String): void	
+ getNombre(): String	
+ setNombre(String): void	
+ getPrecio(): BigDecimal	
+ getStock(): int	
+ setStock(int): void	
+ fromEntity(Producto): ProductoDTO	

Compra	
- idCompra: int	
- estado: String	
- fechaDeRealizacion: Date	
- direccion: Direccion	
- tarjetasBancaria: TarjetasBancaria	
- usuario: Usuario	
- lineasDeCompras: List<LineasDeCompra>	
+ Compra()	
+ Compra(int, String, Date, Direccion, TarjetasBancaria, Usuario, List<LineasDeCompra>)	
+ Compra(int, String, Date, Usuario)	
+ getIdCompra(): int	
+ setIdCompra(int): void	
+ getEstado(): String	
+ setEstado(String): void	
+ getFechaDeRealizacion(): Date	
+ setFechaDeRealizacion(Date): void	
+ getDireccion(): Direccion	
+ setDireccion(Direccion): void	
+ getTarjetasBancaria(): TarjetasBancaria	
+ setTarjetasBancaria(TarjetasBancaria): void	
+ getUsuario(): Usuario	
+ setUsuario(Usuario): void	
+ getLineasDeCompras(): List<LineasDeCompra>	
+ setLineasDeCompras(List<LineasDeCompra>): void	
+ addLineasDeCompra(LineasDeCompra): LineasDeCompra	
+ removeLineasDeCompra(LineasDeCompra): LineasDeCompra	
+ hashCode(): int	
+ equals(Object): boolean	
+ toString(): String	

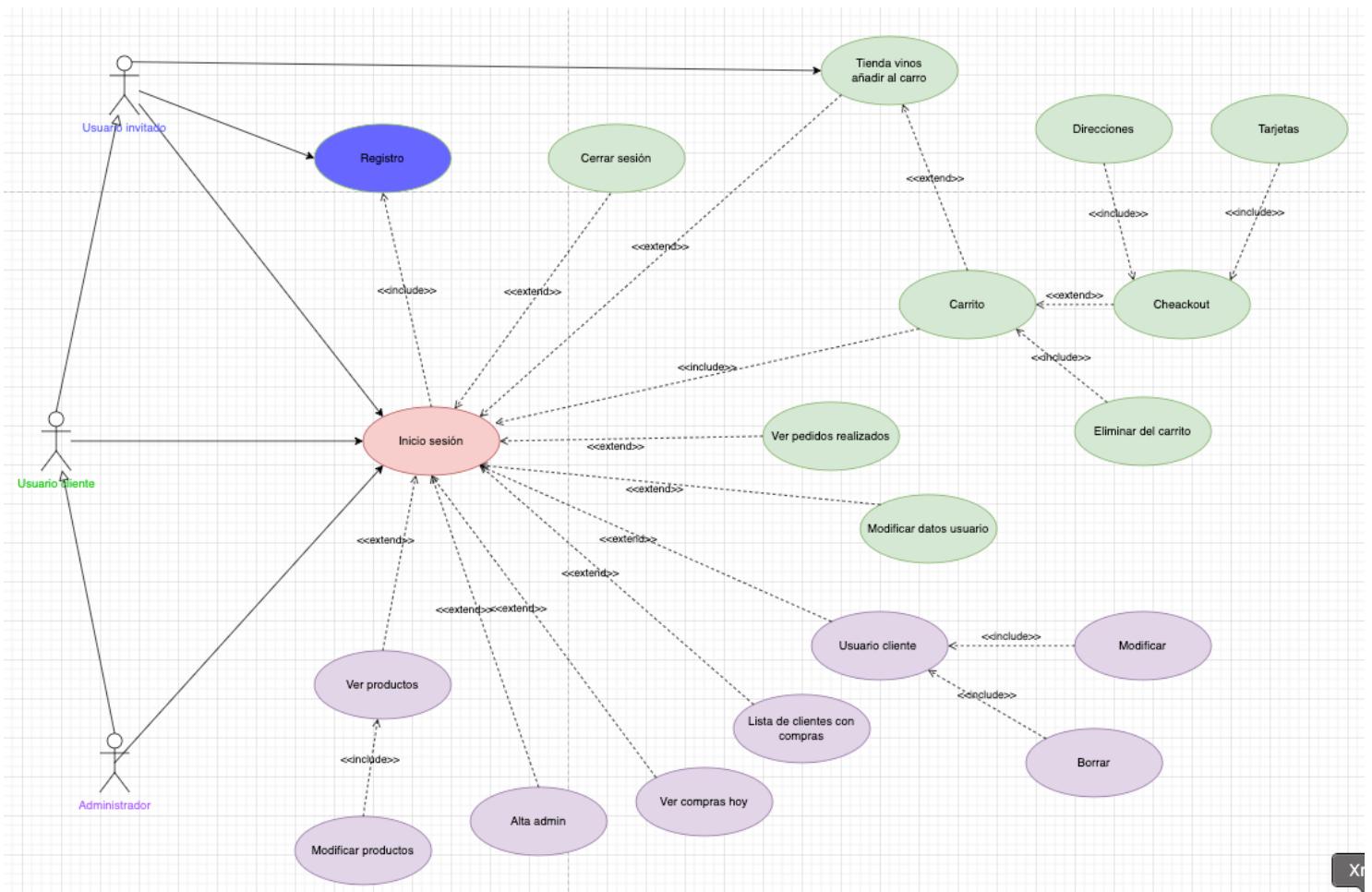
CompraDTO	
- idUsuario: int	
- cantidadPedidos: Integer	
- totalCompra: BigDecimal	
+ getIdUsuario(): int	
+ setIdUsuario(int): void	
+ getCantidadPedidos(): Integer	
+ setCantidadPedidos(Integer): void	
+ getTotalCompra(): BigDecimal	
+ setTotalCompra(BigDecimal): void	
+ getSerialversionuid(): long	



Direcciones
<ul style="list-style-type: none"> <li>- idDireccion: int</li> <li>- calle: String</li> <li>- codigoPostal: BigDecimal</li> <li>- letra: String</li> <li>- localidad: String</li> <li>- numero: int</li> <li>- piso: String</li> <li>- compras: List&lt;Compra&gt;</li> <li>- usuarios: List&lt;Usuario&gt;</li> </ul>
<ul style="list-style-type: none"> <li>+ Direccion()</li> <li>+ Direccion(int, String, BigDecimal, String, String, int, String, List&lt;Compra&gt;, List&lt;Usuario&gt;)</li> <li>+ Direccion(int, String, BigDecimal, String, String, int, String)</li> <li>+ getIdDireccion(): int</li> <li>+ setIdDireccion(int): void</li> <li>+ getCalle(): String</li> <li>+ setCalle(String): void</li> <li>+ getCodigoPostal(): BigDecimal</li> <li>+ setCodigoPostal(BigDecimal): void</li> <li>+ getLetra(): String</li> <li>+ setLetra(String): void</li> <li>+ getLocalidad(): String</li> <li>+ setLocalidad(String): void</li> <li>+ getNumero(): int</li> <li>+ setNumero(int): void</li> <li>+ getPiso(): String</li> <li>+ setPiso(String): void</li> <li>+ getCompras(): List&lt;Compra&gt;</li> <li>+ setCompras(List&lt;Compra&gt;): void</li> <li>+ addCompra(Compra): Compra</li> <li>+ removeCompra(Compra): Compra</li> <li>+ getUsuarios(): List&lt;Usuario&gt;</li> <li>+ setUsuarios(List&lt;Usuario&gt;): void</li> <li>+ hashCode(): int</li> <li>+ equals(Object): boolean</li> <li>+ toString(): String</li> </ul>

TarjetasBancarias
<ul style="list-style-type: none"> <li>- idTarjeta: int</li> <li>- cvv: BigDecimal</li> <li>- fechaCaducidad: Date</li> <li>- nombreTitular: String</li> <li>- numeroTarjeta: BigDecimal</li> <li>- compras: List&lt;Compra&gt;</li> <li>- usuarios: List&lt;Usuario&gt;</li> </ul>
+ method(type): type

○ Diagrama de casos de uso



Todo pasa por inicio de sesión →

Si estas como invitado vemos unas cosas

Si estamos como clientes vemos otras cosas

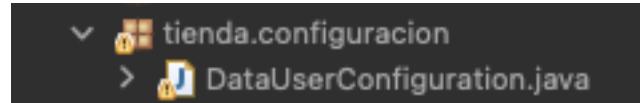
Si estamos con administrador vemos otras cosas



## ■ Planificación del desarrollo

- Despues de reunirnos para planificar el proyecto decidimos y apuntamos en un folio, todas las funcionalidades que tendría la web, aunque según fueron pasando las semanas se añadieron nuevas funcionalidades.
  
- En primer lugar, empezamos por el “Design Thinking”.
  - Primero hicimos la investigación de la competencia y después la investigación del usuario. Parte 1.
  
  - Según acabamos lo anterior comenzamos con la parte 2 cover y mockups.
  
- Al acabar el design thinking comenzamos Base de datos mysql  
La creamos y metimos las tablas, los campos y algunos datos.
  
- Creacion proyecto spring boot mvc
  - Primero decimos la estructura de carpetas, paquetes y clases, jsp...
  - Luego empezamos a meter el código de la parte del servidor → controlar, configuración seguridad, modelo beans, dao, repository y a la vez se fueron metiendo los JSP para ir haciendo pruebas de que el back funcionaba en el front.
  - Configuramos el propertys para la conexión con MySQL y decidir el puerto http por el que funcionara la web.
  - Configuramos el POM con las dependencias necesarias.
  - La parte de los estilos con ccs y bootstrap lo dejamos para lo último.
  
- Memoria
  - La memoria se fue haciendo según se iban haciendo todos los pasos anteriores.

- Explicaciones de la funcionalidad del proyecto



En esta clase de java tenemos la configuración de seguridad, para autenticarnos, para ver que vistas no requieren autenticarse, para codificar las contraseñas.

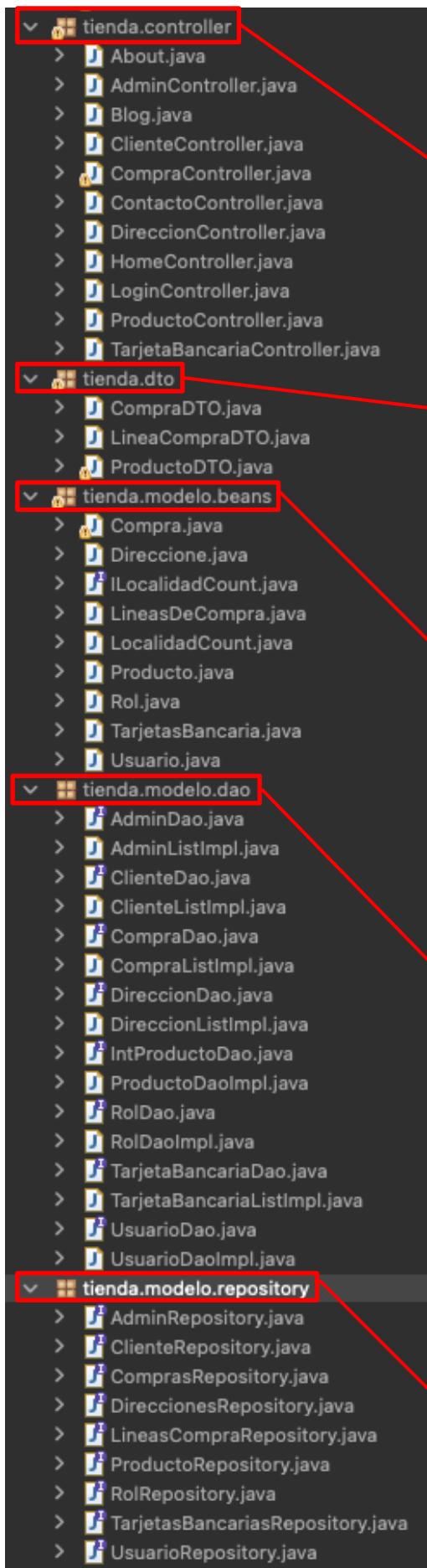
```
// CONFIGURACION DE AUTENTICACION
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.jdbcAuthentication()
        .dataSource(dataSource)
        .usersByUsernameQuery("SELECT username, password, enabled FROM Usuarios WHERE username=?")
        .authoritiesByUsernameQuery(
            "SELECT u.username, r.nombre "
            + "FROM roles_usuarios ru "
            + "INNER JOIN Usuarios u ON u.ID_USUARIO = ru.ID_USUARIO "
            + "INNER JOIN Roles r ON r.id_rol = ru.id_rol "
            + "WHERE u.username = ?"
        );
}

// CONFIGURACIÓN DE SEGURIDAD HTTP
@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .csrf().disable()
        .authorizeRequests()
        // DIRECTORIOS ESTÁTICOS NO REQUIEREN AUTENTICACIÓN
        .antMatchers("/bootstrap/**", "/images/**", "/css/**", "/js/**", "/img/**").permitAll()
        .antMatchers('/rest/demo-bcrypt/**').permitAll()

        // VISTAS PÚBLICAS NO REQUIEREN AUTENTICACIÓN
        .antMatchers(
            "/"
            , "/inicio"
            , "/loginForm"
            , "/logout"
            , "/registro"
            , "/contacto/formContacto"
            , "/blog/bodega"
            , "/sobre/bodega"
            , "/sobre/nosotros"
            , "/search"
            , "/producto/**"
            , "/compra/carrito"
            , "/compra/carrito/**"
            , "/rest/**"
        ).permitAll()

        .anyRequest().authenticated()
        .and()
        .formLogin().disable();
}

// CONFIGURACIÓN DEL CODIFICADOR DE CONTRASEÑAS
@Bean
public PasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}
```



En el **controlador** tenemos todas las clases que interactúan con los JSP y la base de datos o bien guardando información o bien sacando información por los JSP.

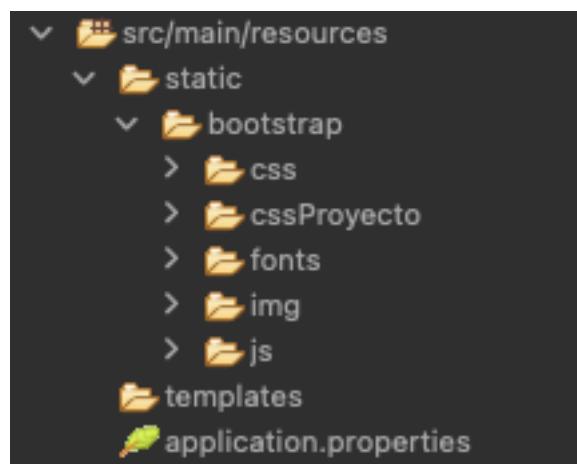
#### Los **DTO** → DATA TRANSFER OBJECT

Saca solo los campos que tú quieras, de una clave ajena.

En el **beans** están todas las propiedades que están ancladas a la base de datos.

En el **dao** están guardados todos los métodos que hacen de intermediario entre el repositorio y el controlador.

En el **repository** están guardados por defecto métodos para hacer prácticamente cualquier cosa como dar altas, bajas, modificar...



- En la carpeta **bootstrap/css** tenemos los css de bootstrap.
- En la carpeta **bootstrap/cssProyecto** están los css que introducimos nosotros.
- En la **img** están todas las imágenes del proyecto.
- Y en la **js** están todos los scripts de javascript con bootstrap.

## También tenemos el application properties →

Donde configuramos el puerto http, la carpeta de las vistas, la base de datos JPA, seguridad....

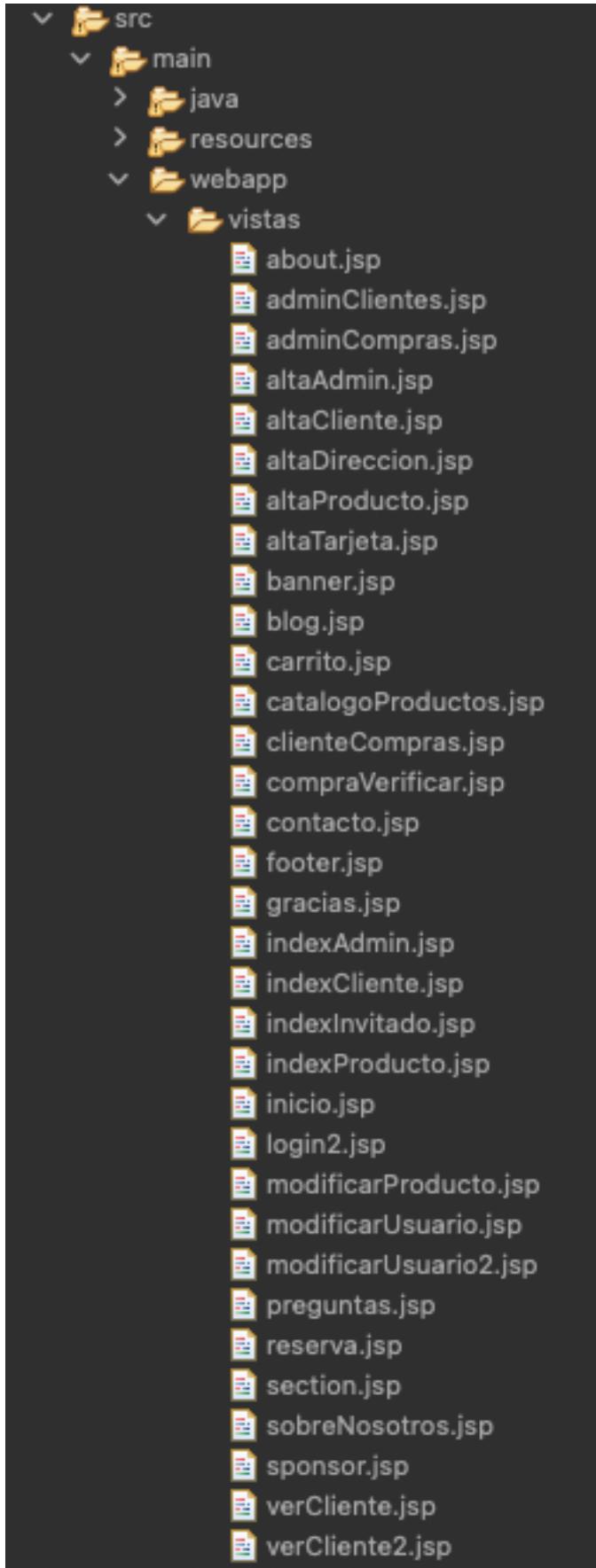
```
server.port=8087

# para las vistas con jsp
spring.mvc.view.prefix=/vistas/
spring.mvc.view.suffix=.jsp

# para mysql 8 bbdd
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/Tienda?serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=12121212 ← Cambiar el password si en vuestra base de datos si
                                         tenéis otra para el root.

spring.jpa.generate-ddl=false
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.show-sql=true
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
logging.level.org.springframework.security=DEBUG

# usuarios security
# spring.security.user.name=admin
# spring.security.user.password=admin
# spring.security.user.roles=CLIENTE
```



Aquí tenemos todas las vistas del proyecto.

Estas vistas están ancladas a archivos css, js, img, controlador

```

https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.13</version>
    <relativePath /> <!-- lookup parent from repository -->
  </parent>
  <groupId>Tienda</groupId>
  <artifactId>TrabajoFinalAD5</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>TrabajoFinalAD5</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-security</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>true</optional>
    </dependency>
    <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

```

- En el POM tenemos las dependencias que tienen que estar metidas, si no el proyecto no funciona.
- Hay dependencias para JPA, MySQL, JSTL, Tomcat, Security...

- **Design Thinking** → se centra en el usuario, cuanto más conozcamos al usuario, mejor experiencia con nuestros productos.

- **Empatizar** → Investigación

- Investigación documental:

- La competencia

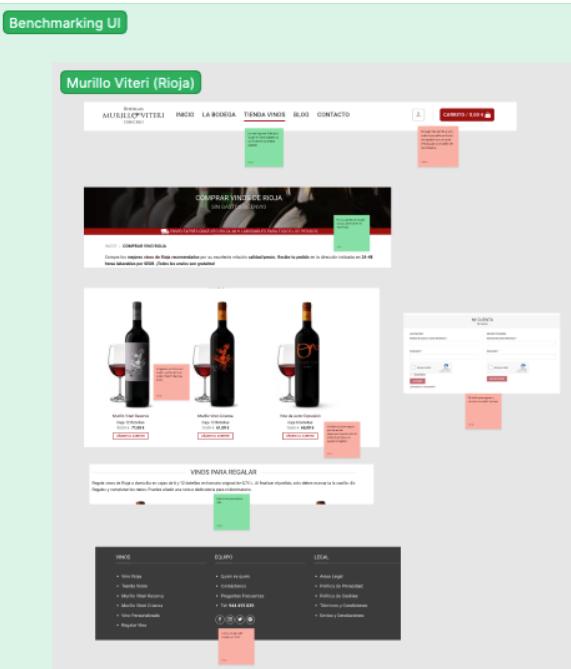
- Situación del mercado

- Tendencia

**Investigación documental**

• Estado actual del mercado	• Investigar otra web	• Qué ofrecen otra web ?	• Tendencia	• Modalidades de pago
<p>El vino es el mejor del mundo.</p> <p>Adrián</p>	<p>La venta 'online', que se disparó durante la pandemia, affanza su tendencia al alza por los nuevos hábitos de los compradores.</p> <p>Adrián</p>	<ul style="list-style-type: none"> <li>-Ribera del duero</li> <li>-Albariño</li> <li>-Rioja</li> <li>-Castro ventosa</li> <li>-Pitacum</li> <li>-Abad don bueno</li> <li>-Prada a tope</li> </ul> <p>Adrián</p>	<ul style="list-style-type: none"> <li>-Carrito</li> <li>-Búsqueda catalogo</li> <li>-Chat ayuda</li> <li>-Pedidos a través del tfn</li> <li>-Catalogo de vinos</li> <li>-Filtros por relevancia, mas vendido, precio - a +, precio + a -</li> <li>-Creación de una cuenta</li> <li>-Envíos internacionales a 44 países.</li> </ul> <p>Adrián</p>	<ul style="list-style-type: none"> <li>-Pago con tarjeta</li> <li>-Transferencia bancaria</li> <li>-Paypal</li> <li>-Bilbao</li> <li>-Apple pay, Google pay</li> <li>-Contanremoto</li> </ul> <p>Aurora</p>
<p>En los dos últimos años la venta online de vinos a permitido a productores y distribuidores seguir conectados con sus clientes habituales y sumar nuevos consumidores.</p> <p>Adrián</p>	<p>El mercado online gana peso entre las bodegas mas pequeñas.</p> <p>Adrián</p>			
<p>Más compras de vino online, con un aumento espectacular que obedece a cambios en los hábitos de consumo y de compra, con visto influidos por la pandemia. Son comportamientos similares que se han producido en todo el mundo. Los confinamientos, las restricciones a la movilidad y los cierres o limitaciones en la hostelería, han llevado a los consumidores a volcarse o refugiarse en compras desde casa a golpe de clic. El comercio online de vino crece.</p> <p>Adrián</p>				

○ Investigación de la competencia: Benchmarking UI



**Murillo Viteri (Rioja)**

The screenshot shows the homepage of the Murillo Viteri website. At the top, there's a navigation bar with links for 'MURILLO VITERI', 'PACIO', 'LA BODEGA', 'TIENDA VINES', 'BLOG', and 'CONTACTO'. A prominent red button labeled 'COMPRAR VINO' is visible. Below the navigation, there's a section titled 'COMPRAR VINO DE RIOJA SIN GASTOS DE ENVÍO' featuring three wine bottles. To the right, there's a search bar and a login form. At the bottom, there's a section for 'VINES PARA REGALAR' with a list of gift options.


**Vinos Ribera (Ribera del duero)**

The screenshot shows the homepage of the Vinos Ribera website. It features a header with the brand name and a navigation menu. Below the header, there's a section for 'PROTECCIÓN AL COMPRADOR' with a rating of '4,74 Excelente'. It displays several wine bottles with their names and prices: 'PAUADA DE ATALAYA 2009' (12,95 €), 'CEPA GUERRERO Crianza 2010' (11,80 €), and 'FINCA REBALDO 2010' (9,40 €). There's also a 'Cosecha 14/15' section with a price of 6,35 €. The footer contains links for 'INFORMACIÓN', 'DISTRIBUIDORES', 'ACCESORIOS', 'CONSEJOS', and 'CONTACTO'.



- Investigación del usuario: Perfiles encontrados

Profiles encontrados

Perfil	Características
Adrián	<ul style="list-style-type: none"><li>• Apasionado del vino</li><li>• 34 años</li><li>• Una churrascada sin vino no es una mariscada.</li></ul>
Adrián	<ul style="list-style-type: none"><li>• Apasionado de vino</li><li>• 48 años</li><li>• Sin un vaso de vino, no puedo comer.</li></ul>
Adrián	<ul style="list-style-type: none"><li>• Apasionado del vino</li><li>• 62 años</li><li>• Ronda de vinos</li></ul>

## • Definir

- User Persona: Objetivos | Motivaciones | Frustraciones
- POV: esas necesidades del usuario las tenemos que resolver.

**User Persona**

The diagram illustrates the connection between a User Persona profile and a corresponding POV (Point of View) profile for a user named Miguel.

**User Persona Profile (Left):**

- Basic Info:** Name: Mónica Martínez, Age: 35, Gender: Female, Nationality: Spanish, Education: University, Work Experience: 10 years.
- Profile Picture:** A portrait of a woman with dark hair and a pink top.
- Motivation:** Mónica es una persona divertida con mucha suerte. Los padres de su marido la han presentado como una excelente persona, con un sentido del humor muy contagioso sobre todo en el trabajo. Lleva 3 años trabajando al lado de su marido, pero a los 10 años de casados se ha separado. Sabe que no tiene mucho tiempo para vivir su vida, pero quiere ser una buena madre y esposa. Tiene una gran pasión por la cocina y el arte. Le encanta cocinar y decorar sus platos. Le encanta viajar y descubrir nuevos lugares.
- Personality:** Inquisitive, Creative, Thrill-seeking, Independent.
- Goals:** Publish, Learn, Create.
- Personas:** Family, Friends, Colleagues.

**Motivations (Mónica):**

- Distribuir de las ideas asociadas con las imágenes con su familia.
- Trabajar con su marido más.

**Needs / Objectives (Mónica):**

- Que su marido esté a su lado. Que le dé apoyo emocional y económico.
- Que su marido le devuelva su confianza.
- Que su marido responda sus mensajes.

**POV (Right):**

**Basic Info:** Age: 48, Gender: Male, Salary: 1000€, Title: Director, Company: Nuevo Nivel, Location: Madrid.

**Profile Picture:** A portrait of a man with short brown hair and a beard, looking directly at the camera.

**Motivation:** La necesidad de conseguir una mejor calidad de vida y mejorar su situación económica.

**Needs / Objectives (Miguel):**

- Teletrabajo todos los días y no tener que viajar para ir a trabajar ni tener que educar.
- Necesita que le devuelvan el producto a la gente de casa o en el trabajo.

**Frustration:**

- Que no pueda disponer de su tiempo libre para leer y hacer deporte.
- Que no pueda enviar el correo electrónico.
- Que no pueda cumplir la relación familiar en la web.

**POV (Bottom Right):**

**Motivation:** Miguel necesita una mejor calidad de vida y mejorar su situación económica.

**Needs / Objectives (Miguel):**

- Que su marido le devuelva su confianza.
- Que su marido le responda sus mensajes.
- Que su marido le devuelva su amor.

**Frustration:**

- No tener viene en las reuniones con su marido.
- Kitchens rapidas.
- Caminar contrarrestando.

- **Idear**

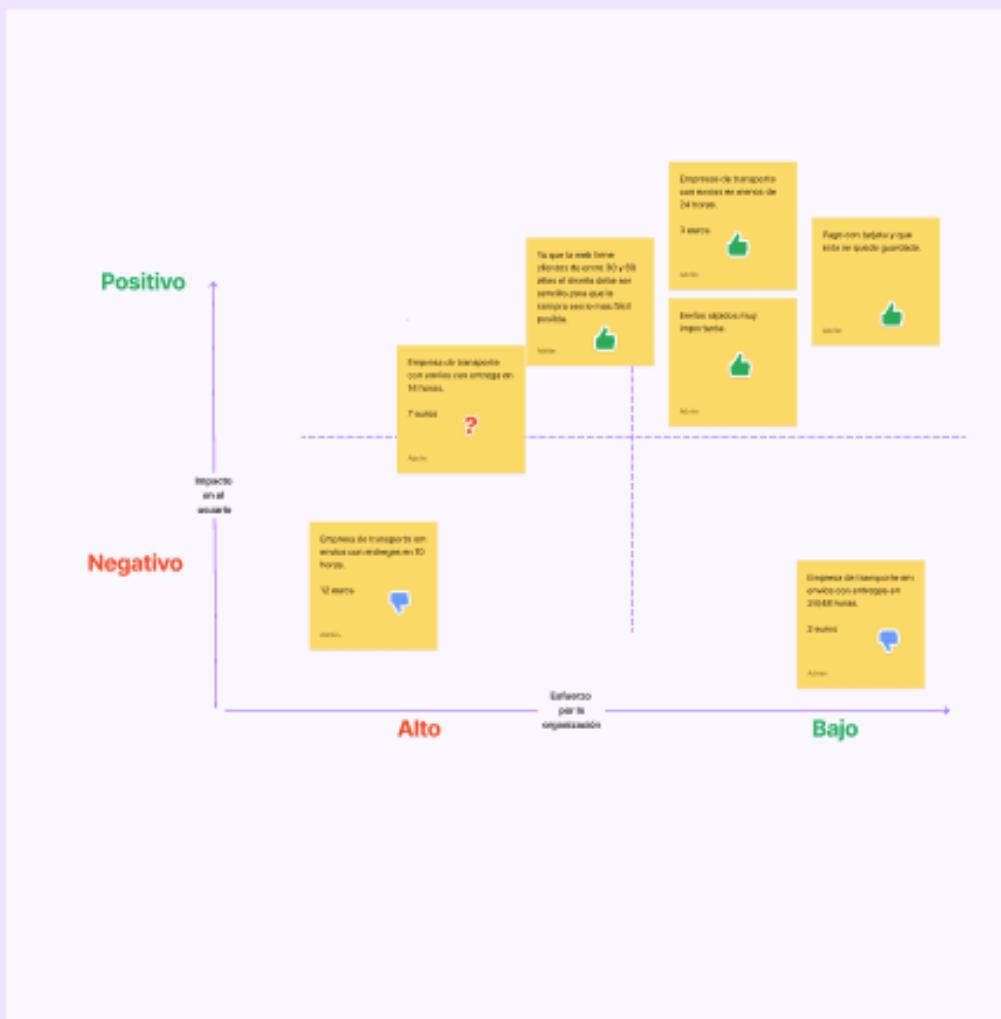
- How Might We: - Proponer ideas en equipo

- Todas son buenas
- Centrarse en el POV
- No discutir
- Una hora



- Brainstorming: votación de las ideas, si valen o no valen

Matriz María



- **Prototipar parte 1**

- User Story Mapping:

- MVP1: lo mínimo que necesita una web para funcionar.

- MVP 2: añades funcionalidades en futuras actualizaciones a medio plazo.

- MVP 3: añades funcionalidades en futuras actualizaciones a más largo plazo.



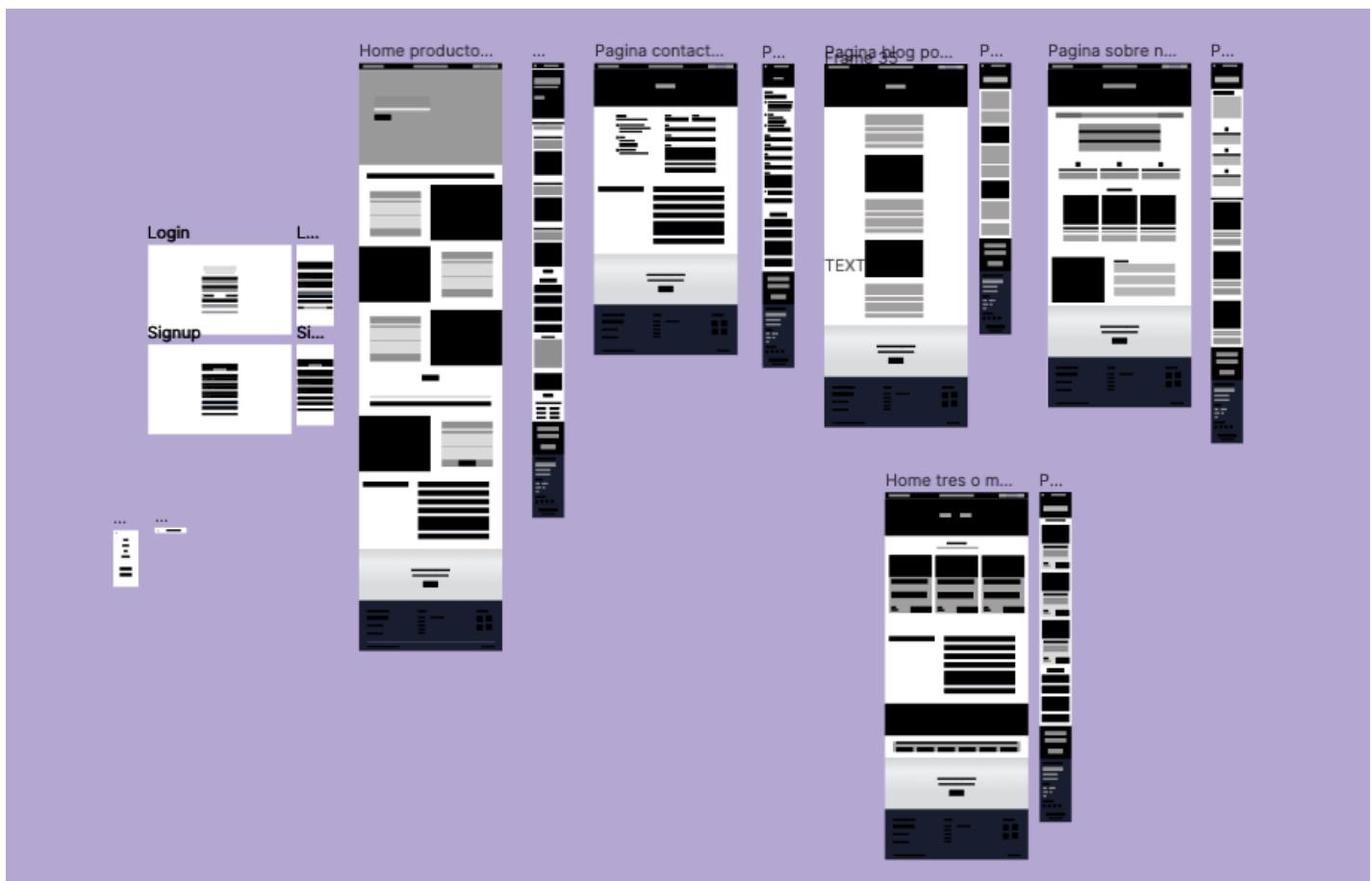
- Site Map: mapa del sitio web



- 
- **Prototipar parte 2**
    - Cover: logo de la web

# Bodegas & Viñedos

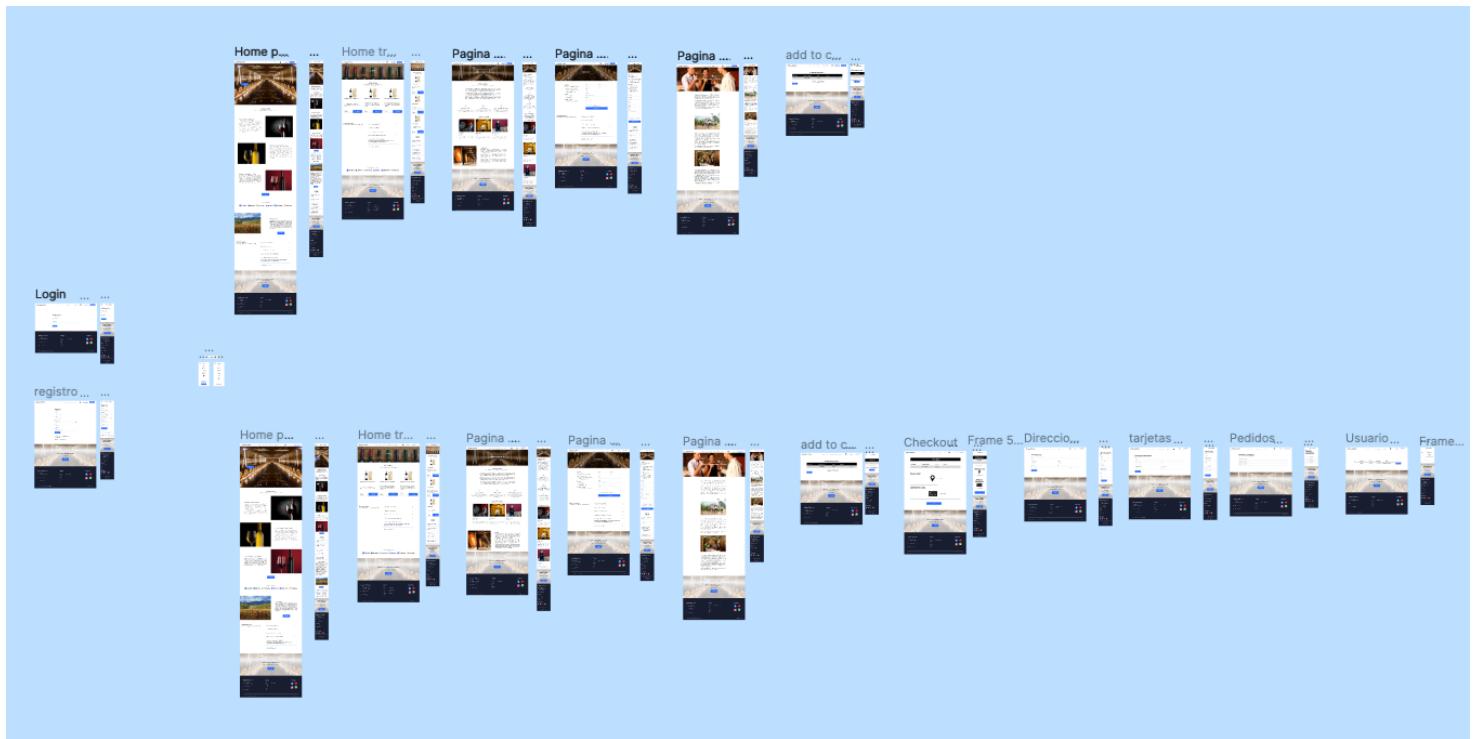
- Wireframes: maquetación de la web en blanco y negro



- Guía de estilos: colores, botones, tipografía, iconográfica, fotográfica, formularios

The image is a composite of several design-related screenshots. At the top left, there's a section titled 'Tipografía Mobile' showing 'Plus Jakarta Sans' with various font weights and sizes. Below it is another 'Tipografía mobile' section for 'Plus Jakarta Sans'. The middle section shows a 'Actions' button with a dropdown menu containing 'Acciones', 'Borrar', 'Actualizar', 'Nuevo', and 'Importar'. To the right of this are two color palettes: one with blue, yellow, and red colors, and another with blue, green, and orange colors. Below these palettes are five colored squares with labels: 'Color Azul Oscuro' (blue), 'Color Gris Oscuro' (dark grey), 'Degrado Claro' (light green), 'Color Azul Oscuro' (blue), 'Color Gris Oscuro' (dark grey), 'Degrado Claro' (light green), 'Degrado Claro' (light green), 'Acento' (brown), 'Degrado Claro' (light green), 'Acento' (brown), and 'Degrado Claro' (light green). The bottom section is titled 'Iconografia' and shows a grid of various icons including a plus sign, a list, a location pin, a phone, a checkmark, a user profile, a video camera, a play button, a search icon, a magnifying glass, a person icon, a gear, a link, and social media icons for Facebook, YouTube, Instagram, and WhatsApp. The bottom right section is titled 'Formularios' and shows examples of input fields, a text area, and checkboxes.

- Mockups: maquetación de la web como quedaría realmente con todos sus colores, botones...TODO.



## ⇒ 7.0 - Conclusiones y mejoras del proyecto

En cuanto a las conclusiones del proyecto, creemos que hemos logrado las expectativas que teníamos desde un principio en base a la idea que habíamos planteado.

Al inicio habíamos desarrollado la parte del front en angular y habíamos logrado la implementación de unas cuantas funcionalidades expuestas en el anteproyecto. A medida que fuimos avanzando nos dimos cuenta de que no poseíamos tantos conocimientos de conectar la api creada en Spring a nuestra parte frontal del proyecto y a pesar de muchos intentos e investigaciones relacionadas con el tema tuvimos que tomar la decisión de desarrollar nuestro trabajo dentro del mismo entorno de Spring mvc.

Respecto a nuestras funcionalidades (en entorno Spring mvc) hemos conseguido implementar todas las mencionadas en el anteproyecto y alguna que otra más.

En cuanto a nuestros resultados obtenidos hemos logrado una implementación de login con seguridad, un proceso de compra con sesión o bien desde el cliente invitado guardar sus productos en la cesta para cuando inicie sesión y/o se registre siga con los productos cargados.

También logramos crear un formulario de contacto enviando los datos a una mensajería externa con mailtho. Por otro lado, conseguimos crear varias funcionalidades en el apartado administrador bastante útiles para el cliente final que recibiese esta web. Consulta de compras, listado de clientes, modificación de usuarios, pedidos totales, etc.

En resumen, podríamos decir que estamos contentos con los resultados obtenidos a pesar de que sabemos que existirían mejoras para un futuro en nuestra web, pero orgullosos y satisfechos de haber conseguido el resultado final.

## ⇒ 8.0 - Bibliografía

- **Documentación oficial y tutoriales →**

Documentación oficial de Spring Boot. (<https://spring.io/projects/spring-boot>): Recursos, guías y documentación detallada sobre el desarrollo con Spring Boot.

Documentación oficial de Bootstrap. (<https://getbootstrap.com/docs/5.0/getting-started/introduction/>): Documentación completa de Bootstrap, incluyendo ejemplos y guías de uso.

Documentación oficial de MySQL. (<https://dev.mysql.com/doc>): Manuales y referencias de MySQL para administración y desarrollo de bases de datos.

Figma Community. (<https://www.figma.com/community>): Recursos y plantillas compartidas por la comunidad de Figma, incluyendo tutoriales y ejemplos de diseño de interfaces.

- **Artículos y tutoriales en línea →**

Baeldung. (<https://www.baeldung.com/>): Artículos y tutoriales sobre desarrollo con Spring Boot y otros temas relacionados con Java.

Codecademy. (<https://www.codecademy.com/learn/learn-spring>): Curso interactivo para aprender Spring Boot.

W3Schools MySQL Tutorial. (<https://www.w3schools.com/sql/>): Tutoriales y ejemplos prácticos de SQL y MySQL.

BootstrapBay. (<https://bootstrapbay.com/blog/>): Artículos, tutoriales y recursos adicionales sobre Bootstrap y diseño web responsive.



## ⇒ 9.0 - Anexos

- **Contenido del proyecto:**

- Figma “Design Thinking”
  - Bodegas y viñedos parte 1.jam
  - Bodegas y viñedos parte 2.fig
  - <https://www.figma.com/files/project/87876081/-TFG?fuid=1152000419334319036>
- Framework Spring Boot MVC
  - Puerto navegador 8087 → localhost:8087/
  - Carpeta proyecto → Grupo14SpringBootMVC
  - Eclipse instalación en el Marketplace spring tools 4
  - Usuario cliente → edix Contraseña → 12345
  - Admin → edix2023 Contraseña → 12345
- Base de datos
  - Instalar MySQL server y MySQL Workbench
  - Ejecutar el Script en MySQL Workbench
  - Script → Grupo14ScriptMySQL.sql
  - Cambiar la contraseña en el application properties si en vuestra base de datos en el root tenéis otra.
- Memoria
  - Grupo14Memoria.pdf
- Diagrama UML
  - Grupo14UMLCasoDeUso.drawio
  - Grupo14UMLClases.drawio