



# Sportify

Manual de Administración

Alejandro Alba

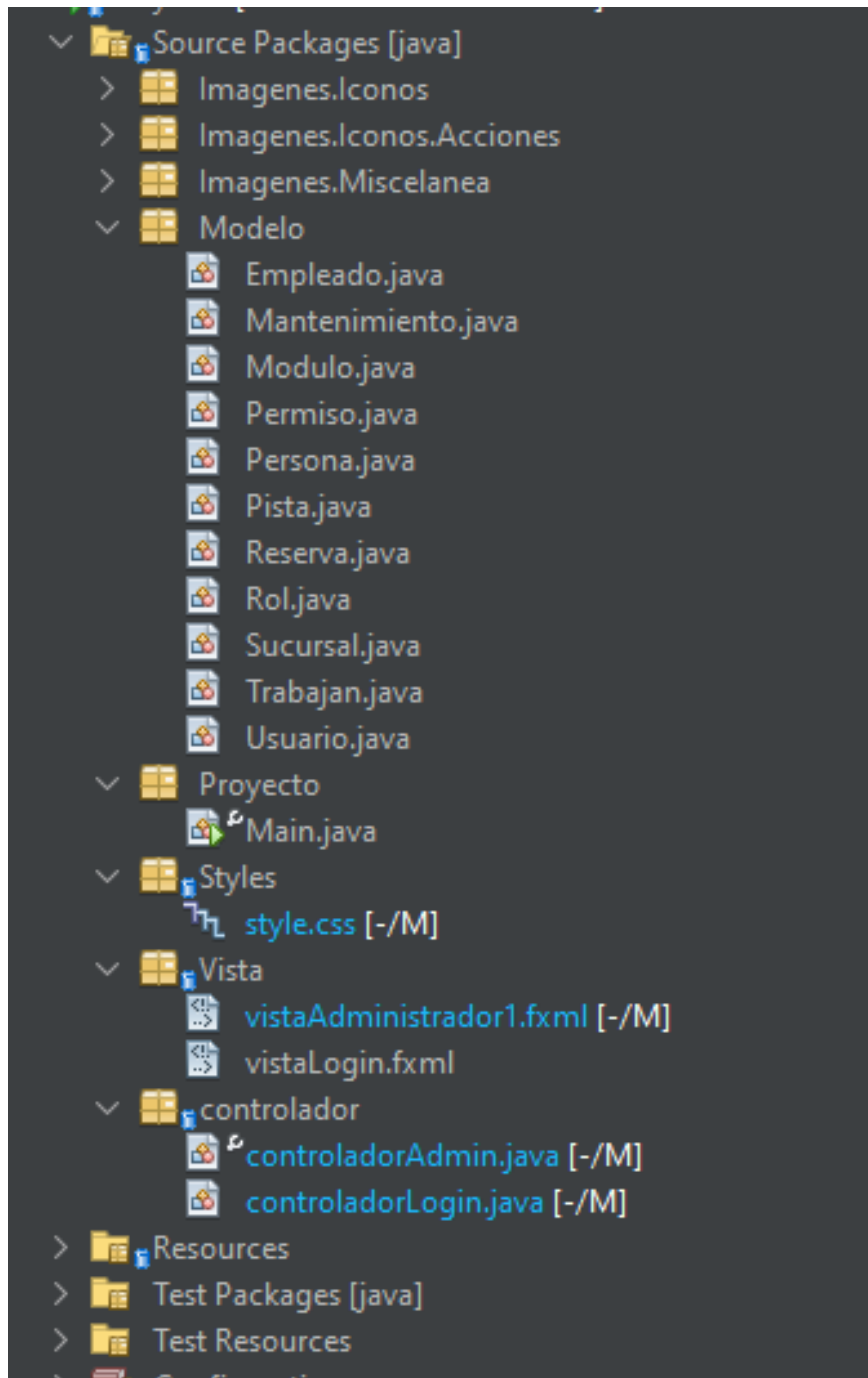


## Contents

Estructura del proyecto .....	2
Base de datos .....	3
Modelo .....	4
Vista .....	4
vistaLogin.fxml .....	4
vistaAdministrador.fxml .....	4
Controlador .....	5
controladorLogin.java .....	5
initialize() .....	5
Evento de btnEntrar() .....	5
sacarPersona() .....	5
existeUsuario() .....	5
controladorAdministrador.java .....	5
cargar*() .....	5
sacar*() .....	5
dameLista*() .....	6
sacarPermiso() .....	6
setControladorEnlace() .....	6

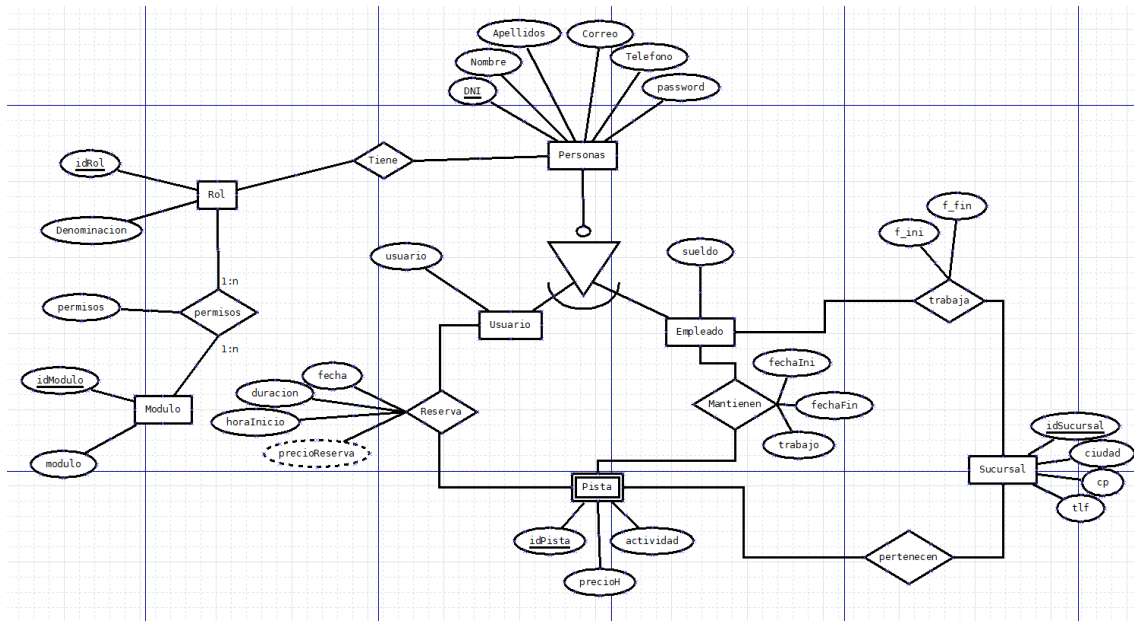
## Estructura del proyecto

El proyecto esta desarrollado de acuerdo a: Modelo / Vista / Controlador

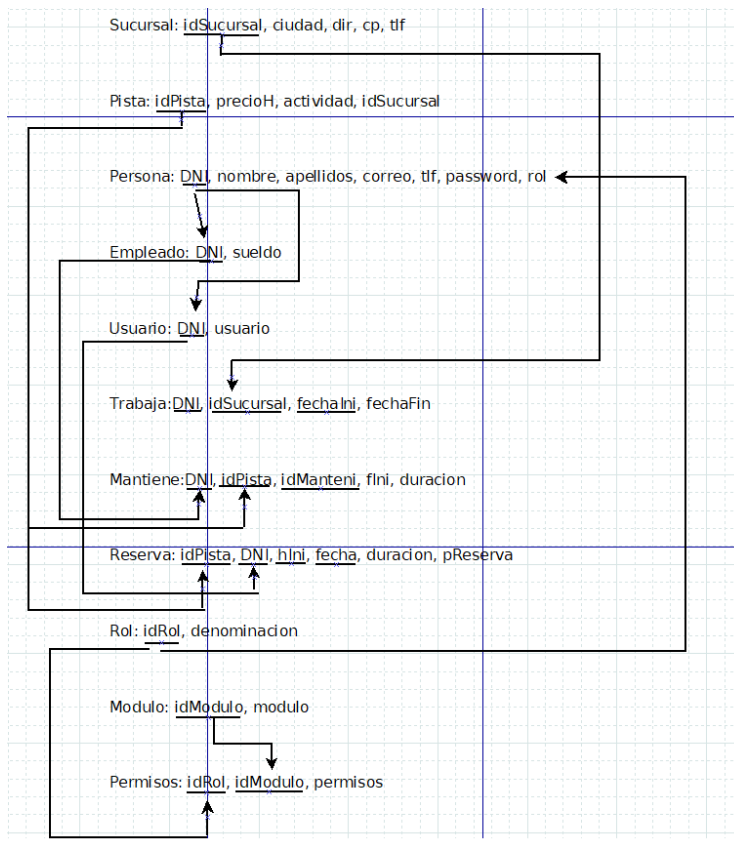


## Base de datos

La base de datos está compuesta por 11 tablas, a continuación se podrán ver en el diagrama entidad relación.



Este es el paso a tabla



## Modelo

Aquí se definen las clases con las que va a funcionar la aplicación

Todas las clases que hay se van a mostrar directamente en la aplicación menos:

- Persona.java ya que de esta clase heredan las clases Usuario.java y Empleado.java

## Vista

Aquí se muestran alojan los ficheros fxml que se van a mostrar cuando se ejecute la aplicación

vistaLogin.fxml

vistaAdministrador.fxml

ID Sucursal	Ciudad	Código Postal	Dirección	Teléfono
1	Granada	30000	Arabial	987654321
2	Sevilla	60000	Nervión	123987456
3	Barcelona	12000	Calle Sant Jordi	123456780
4	Logroño	234567	Gran Vía de Colón	678907778

## Controlador

Aquí se van a hacer todas las operaciones lógicas de la aplicación, también es donde se cargan los elementos de javafx que son los que se van a mostrar en la vista.

### `controladorLogin.java`

Aquí se controla el inicio de sesión

Los métodos más destacados son:

#### `initialize()`

Aquí se inicializan los componentes en caso de que haga falta.

#### `Evento de btnEntrar()`

Es el encargado de administrar los otros dos métodos y si está todo correcto nos manda al otro controlador, pasando a la persona como parámetro.

#### `sacarPersona()`

Devuelve un objeto Persona dado el dni que se inserta en el textfield.

#### `existeUsuario()`

Comprueba:

- Existe el usuario
- En caso de que exista de que la contraseña asignada al usuario sea la misma que se especifica

### `controladorAdministrador.java`

Es el que controla todo en la aplicación.

Al tener 10 pestañas tiene repetido 10 veces los mismos métodos.

Estos son los más destacables (el "\*" especifica que esta que se usa para distintos objetos de Modelo).

#### `cargar*()`

El método se activa por el evento del botón cargar de cada pestaña, aquí recoge los datos del elemento que está seleccionado y los pone en el formulario de inserción, sirve sobre todo para Actualizar.

#### `sacar*()`

Devuelve el objeto especificado dada la clave, se usa para rellenar los combobox necesarios.

dameLista\*()

Devuelve un ObsevableList del objeto especificado.

Hay métodos que tienen un parámetro llamado permiso, aquí simplemente controla de que si tiene un permiso restringido filtra solo por donde aparezca el DNI del objeto persona que se ha dado como parámetro desde controladorLogin.java.

Estos métodos se usan para rellenar las tablas

sacarPermiso()

Devuelve el permiso dado el usuario y el modulo

setControladorEnlace()

Este método es como el initialize, aquí se declaran todos los componentes necesarios, aquí se:

- Rellenan las tablas
- Rellenan combobox
- Se crean los menús contextuales

//Se especifican los menus contextuales para eliminar y cargar

```
contextMenu.setAutoHide(true);
```

```
item.setOnAction(event -> {
```

```
    eliminar();
```

```
});
```

```
contextMenu.getItems().add(item);
```

```
tv.setContextMenu(contextMenu);
```

```
tv.setOnMouseClicked(event -> {
```

```
    if (event.getButton() == MouseButton.PRIMARY && event.getClickCount() == 2) {
```

```
        cargar();
```

```
    }
```

```
});
```

- Se controla la vista de las pestañas

//Se comprueba de que la persona que ha iniciado tiene el permiso "R" de lectura, si no lo tiene oculta la pestaña

```
if (!sacarPermiso(1, persona.getRol()).contains("R")) {
```

```
    tbGeneral.getTabs().remove(tabSucursal);
```

```
}
```