



PROYECTO FINAL

FP DUAL

Descripción breve

Documentación explicativa del proyecto final de la formación Dual.

NTT Data

ALMA GARCÍA DE LA OBRA

agadelao@emeal.nttdata.com

ÍNDICE

1 -	Introducción	1
2 -	Diagrama Entidad-Relación	2
2.1 -	Entidades	3
2.2 -	Relaciones	3
3 -	Proyecto WEB	4
3.1 -	Software:.....	4
3.2 -	Dependencias:	4
3.3 -	Consideraciones previas de la Base de Datos.....	5
3.4 -	Estructura del proyecto Web	6
3.4.1 -	Capa de persistencia.....	7
3.4.2 -	Capa de servicios	10
3.4.3 -	Capa de controladores	14
3.5 -	Vistas.....	18

1 - Introducción

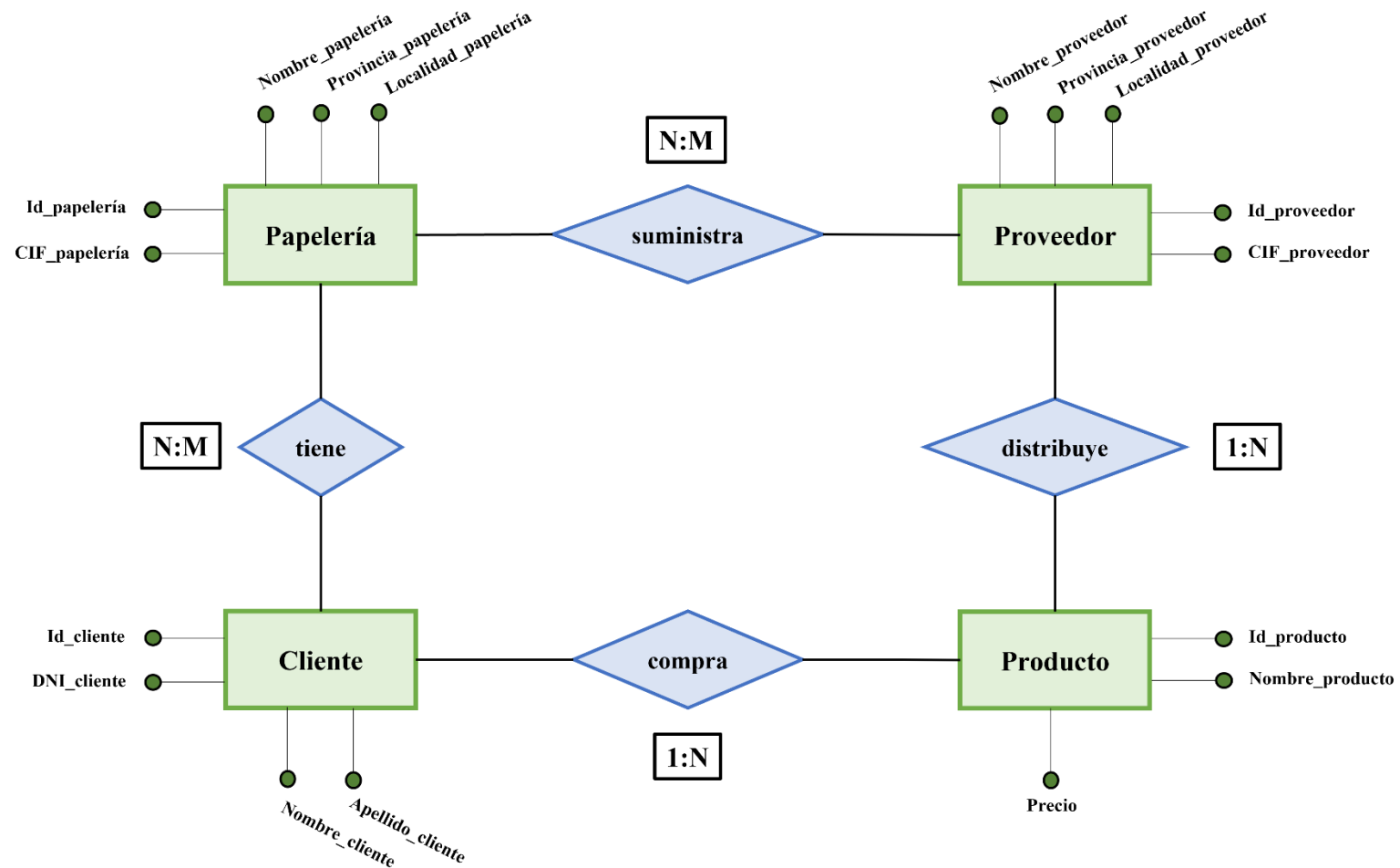
El mundo empresarial es un mundo en constante movimiento, por lo que es necesario un constante seguimiento de todas las transacciones que se producen, no solo para tener cierto control, sino para el posterior análisis de estas transacciones con el fin de mejorar los productos, los servicios, las estrategias de marketing, etc.

Por tanto, debemos tener constancia de forma persistente de los productos que se venden, a quiénes se vende, quiénes son los proveedores, etc.

El presente proyecto tiene como objetivo la realización de una aplicación web básica que permita realizar las operaciones básicas de un CRUD (Crear, Consultar, Actualizar y Borrar) en una serie de entidades relacionadas con el mundo empresarial. La temática escogida son las “Papelerías”.

2 - Diagrama Entidad-Relación

Nuestra aplicación estará formada por 4 entidades relacionadas según el siguiente diagrama de entidad-relación:



2.1 - Entidades

Las entidades de este proyecto son:

- **Papelería**: Representa los comercios dedicados a la venta de productos de papelería y de manualidades. Sus atributos principales son: ID, CIF, Nombre, Provincia y Apellido.
- **Proveedor**: Representa los proveedores que proveen a las papelerías de los productos de manualidades. Sus atributos principales son: ID, CIF, Nombre, Provincia y Localidad.
- **Cliente**: Representan las personas que adquieren productos de las papelerías. Sus atributos principales son: ID, DNI, Nombre y Apellido completo.
- **Producto**: Representan los productos de manualidades que son suministrados por los proveedores a las papelerías y comprados por los clientes.

2.2 - Relaciones

Las relaciones consideradas entre las entidades son las siguientes:

- **Papelería-Proveedor**: Se considera una relación N:M, es decir, una papelería puede comprar a varios proveedores y un proveedor puede suministrar a varias papelerías.
- **Papelería-Cliente**: Se considera una relación N:M, es decir, una papelería vende a varios clientes y un cliente puede comprar en varias papelerías diferentes.

- **Proveedor-Producto**: Se considera una relación 1:N, es decir, un proveedor puede proveer varios productos, pero un producto solo es vendido por un proveedor.
- **Cliente-Producto**: Se considera una relación 1:N, es decir, un cliente puede comprar varios productos pero un producto solo puede ser comprado por un cliente.

3 - **Proyecto WEB**

3.1 - **Software:**

Se han empleado los siguientes programas para el desarrollo del presente proyecto:

- **IDE**: Como IDE principal se ha empleado Eclipse, además de la herramienta de desarrollo Spring Initializr.
- **Base de datos/Gestor de la Base de datos**: La Base de datos empleada ha sido Oracle y el gestor empleado SQL Developer.
- **Otros**: NotePad++.

3.2 - **Dependencias:**

Se han empleado las siguientes dependencias para el desarrollo del proyecto:

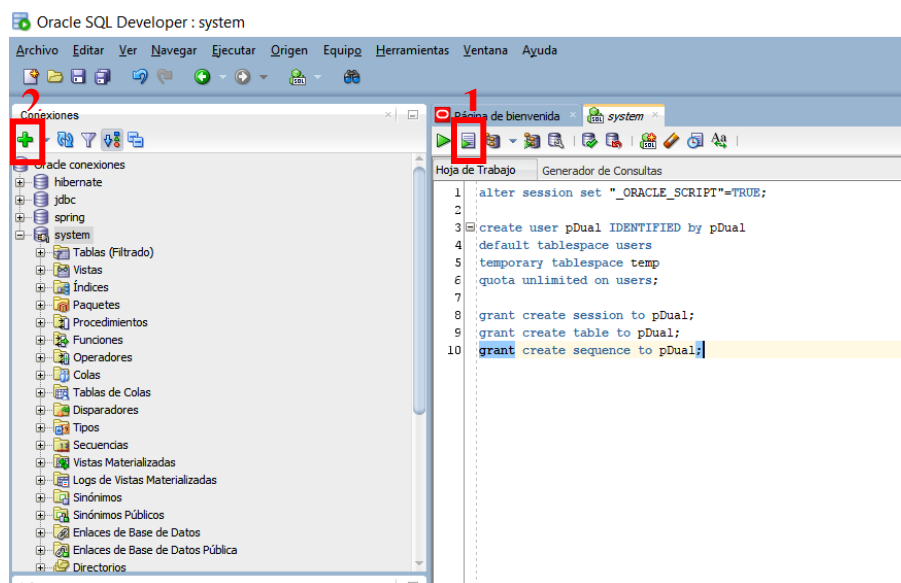
- **Spring Boot/Spring Core**: Para el desarrollo principal de la aplicación, la inyección de dependencias y el despliegue de la aplicación en un servidor embebido en Spring Boot.

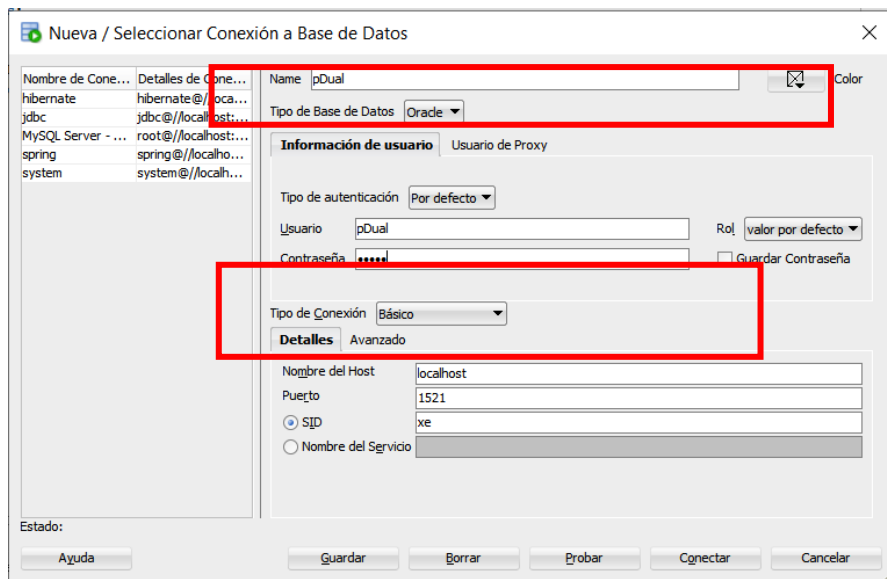
- **Spring Data/Hibernate/JDBC**: Para la conexión, manejo y gestión de la base de datos.
- **Spring MVC/Thymeleaf**: Para el desarrollo del Front de la aplicación.
- **SLF4J/Logback**: Para el manejo y uso de Loggers.
- **Maven-Javadoc**: Para el desarrollo de documentación.

3.3 - **Consideraciones previas de la Base de Datos**

Previamente al despliegue de la aplicación se debe de configurar el archivo “application.properties” de la carpeta *src/main/resources*, donde el usuario deberá definir la conexión a la base de datos que desea emplear.

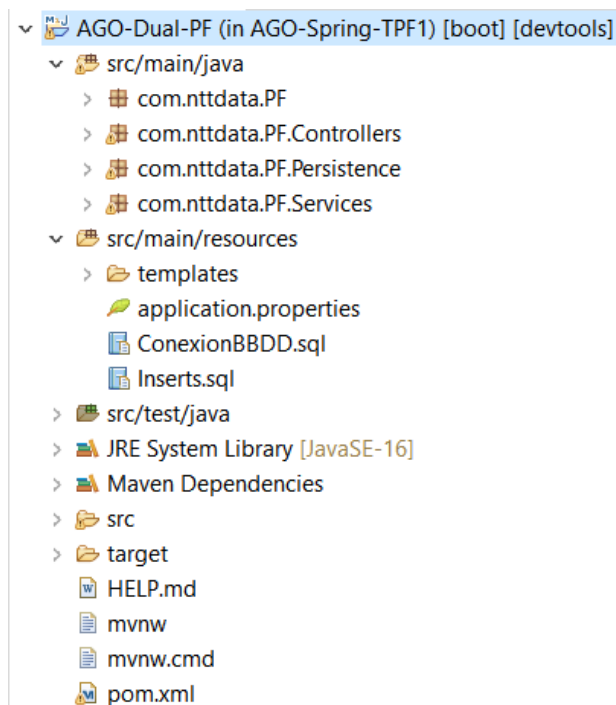
NOTA: Si el usuario lo desea se ha adjuntado en esta misma carpeta un script llamado “**ConexionBBDD**” que permite crear una conexión a la base de datos de Oracle, en caso de que se desee emplear una conexión aparte para este proyecto. Únicamente deberá lanzar el script desde un usuario que posea los permisos necesarios, como *System*, y posteriormente realizar la conexión desde el SQL Developer con el usuario y contraseña establecidos en el script.



















3.4 - Estructura del proyecto Web

Este proyecto web se ha estructurado en base a la estructura básica de un proyecto realizado con Spring: Main, Controladores, Persistencia y Servicios.



3.4.1 - Capa de persistencia

Se corresponde con la capa de la aplicación Web que se encarga del manejo y gestión de la base de datos.

- ▼  com.nttdata.PF.Persistence
 - >  AbstractEntity.java
 - >  Client.java
 - >  ClientRepositoryI.java
 - >  CustomizedProductRepository.java
 - >  CustomizedProductRepositoryImpl.java
 - >  CustomizedStationeryRepository.java
 - >  CustomizedStationeryRepositoryImpl.java
 - >  Product.java
 - >  ProductRepositoryI.java
 - >  Provider.java
 - >  ProviderRepositoryI.java
 - >  Stationery.java
 - >  StationeryRepositoryI.java

Está formada por los elementos descritos a continuación.

ENTIDADES		
ENTIDAD	DESCRIPCIÓN	ATRIBUTOS
AbstractEntity	Entidad superior de la cual extienden todas las demás	<ul style="list-style-type: none"> - updateUser (String) - updateDate (Date)
Client	Representa la tabla <u>CLIENTS</u> de la base de datos y se corresponde con la entidad <u>Cliente</u> del diagrama E-R.	<ul style="list-style-type: none"> - idClient (Long) - dniClient (String) - nameClient (String) - surnameClient (String) - productsClient (List<Product>) - stationeriesList (List<Stationery>)
Product	Representa la tabla <u>PRODUCTS</u> de la base de datos y se corresponde con la entidad <u>Producto</u> del diagrama E-R.	<ul style="list-style-type: none"> - idProduct (Long) - nameProduct (String) - priceProduct (double) - idClient (Client) - idProvider (Provider)
Provider	Representa la tabla <u>PROVIDERS</u> de la base de datos y se corresponde con la entidad <u>Proveedor</u> del diagrama E-R.	<ul style="list-style-type: none"> - idProvider (Long) - cifProvider (String) - nameProvider (String) - provinceProvider (String) - cityProvider (String) - productsProvider (List<Product>) - stationeriesList (List<Stationery>)
Stationery	Representa la tabla <u>STATIONERIES</u> de la base de datos y se corresponde con la entidad <u>Papelería</u> del diagrama E-R.	<ul style="list-style-type: none"> - idStationery (Long) - cifStationery (String) - nameStationery (String) - provinceStationery (String) - cityStationery (String) - providerList (List<Provider>) - clientList (List<Client>)

INTERFACES

INTERFAZ	DESCRIPCIÓN	MÉTODOS	IMPLEMENTACIÓN
ClientRepositoryI	Métodos para la gestión de la tabla CLIENTS de la base de datos. Extiende de <u>JpaRepository</u> .	- Client findByDniClient (String dni)	-
ProviderRepositoryI	Métodos para para la gestión de la tabla PROVIDERS de la base de datos. Extiende de <u>JpaRepository</u> .	- Provider findByCifProvider (String cifProvider)	-
CustomizedProductRepository	Interfaz empleada para generar los métodos desarrollados con Criteria para la tabla PRODUCTS.	- List<Product> productsClient (Long idClient)	CustomizedProductRepositoryImpl
ProductRepositoryI	Métodos para para la gestión de la tabla PRODUCTS de la base de datos. Extiende de <u>JpaRepository</u> y de <u>CustomizedProductRepositor</u> .	- Product findByNameProduct (String nameProduct)	-
CustomizedStationeryRepository	Interfaz empleada para generar los métodos desarrollados con Criteria para la tabla STATIONERIES.	- List<Stationery> searchStationeryByIdClient (Long idClient)	CustomizedStationeryRepositoryImpl
StationeryRepositoryI	Métodos para para la gestión de la tabla STATIONERIES de la base de datos. Extiende de <u>JpaRepository</u> y de <u>CustomizedStationeryRepositoy</u> .	- Stationery findByNameStationery (String nameStationery) - Stationery findByCifStationery (String cifStationery)	-

3.4.2 - Capa de servicios

Se corresponde con la capa de la aplicación Web que se encarga de gestionar los servicios ofrecidos por la aplicación.

```
▼ com.nttdata.PF.Services
  > ClientDto.java
  > ClientServicel.java
  > ClientServiceImpl.java
  > ProductDto.java
  > ProductServicel.java
  > ProductServiceImpl.java
  > ProviderDto.java
  > ProviderServicel.java
  > ProviderServiceImpl.java
  > StationeryDto.java
  > StationeryServicel.java
  > StationeryServiceImpl.java
```

Está formada por los elementos descritos a continuación.

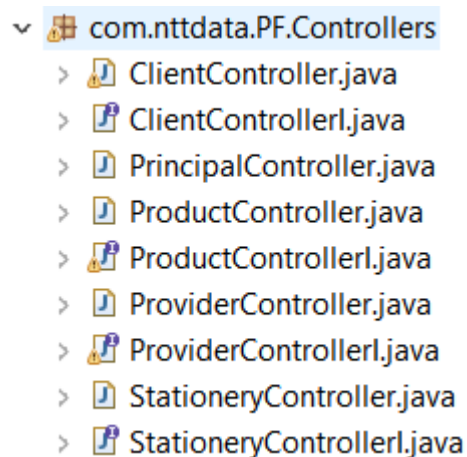
DATA TRANSFER OBJECTS (DTOs)		
ENTIDAD	DESCRIPCIÓN	ATRIBUTOS
ClientDto	DTO correspondiente a la entidad CLIENT	<ul style="list-style-type: none"> - dniClient (String) - nameClient (String) - surnameClient (String)
ProductDto	DTO correspondiente a la entidad PRODUCT	<ul style="list-style-type: none"> - nameProduct (String) - priceProduct (double)
ProviderDto	DTO correspondiente a la entidad PROVIDER	<ul style="list-style-type: none"> - cifProvider (String) - nameProvider (String)
StationeryDto	DTO correspondiente a la entidad STATIONERY	<ul style="list-style-type: none"> - cifStationery (String) - nameStationery (String)

SERVICIOS			
INTERFAZ	IMPLEMENTACIÓN	MÉTODOS	DESCRIPCIÓN
ClientServiceI	ClientServiceImpl	Client addClient (ClientDto clientDto)	Añade un cliente a la BBDD
		Client updateClient (ClientDto clientDto)	Actualiza un cliente de la BBDD
		void deleteClient (final String dniClient)	Elimina un cliente de la BBDD
		List<ClientDto> showAllClients()	Lista todos los clientes de la BBDD
		Client searchClientByDni (final String dniClient)	Busca un cliente de la BBDD por su DNI
		List<ProductDto> showAllProductsBought (final String dniClient)	Lista todos los productos comprados por un cliente
		List<StationeryDto> showStationariesFromClient (final String dniClient)	Lista todas las papelerías a las que ha acudido un cliente
ProductServiceI	ProductServiceImpl	Product addProduct (final ProductDto productDto)	Añade un producto a la BBDD
		Product updateProduct (final ProductDto productDto)	Actualiza un producto de la BBDD
		void deleteProduct (final String nameProduct)	Elimina un producto de la BBDD
		List<ProductDto> showAllProducts()	Lista todos los productos de la BBDD
		Product findByNameProduct (final String nameProduct)	Busca un producto de la BBDD por su nombre

SERVICIOS			
INTERFAZ	IMPLEMENTACIÓN	MÉTODOS	DESCRIPCIÓN
ProviderServiceI	ProviderServiceImpl	Provider addProvider (final ProviderDto providerDto)	Añade un proveedor a la BBDD
		Provider updateProvider (final ProviderDto providerDto)	Actualiza un proveedor de la BBDD
		deleteProvider (final String cifProvider)	Elimina un proveedor de la BBDD
		List<ProviderDto> showAllProviders()	Lista todos los proveedores de la BBDD
		Provider findByCifProvider (final String cifProvider)	Busca un proveedor de la BBDD por su CIF
StationeryServiceI	StationeryServiceImpl	Stationery addStationery (final StationeryDto stationeryDto)	Añade una papelería a la BBDD
		Stationery updateStationery (final StationeryDto stationeryDto)	Actualiza una papelería de la BBDD
		void deleteStationery (final String nameStationery)	Elimina una papelería de la BBDD
		List<StationeryDto> showAllStationeries()	Lista todas las papelerías de la BBDD
		Stationery findByNameStationery (final String nameStationery)	Busca una papelería de la BBDD por su nombre
		Stationery findByCifStationery (final String cifStationery)	Busca una papelería de la BBDD por su CIF

3.4.3 - Capa de controladores

Se corresponde con la capa de la aplicación Web que maneja todas las peticiones realizadas por el usuario a través de las vistas devolviendo la respuesta solicitada haciendo uso de los servicios.



Está formada por los elementos descritos a continuación.

CONTROLADORES			
INTERFAZ	IMPLEMENTACIÓN	MÉTODOS	DESCRIPCIÓN
PrincipalControllerI	PrincipalController	String homePage()	Dirige a la vista home.html
ClientControllerI	ClientController	String clientPage()	Dirige a la vista client.html
		String addOrUpdateClient (final ClientDto clientDto)	Añade o actualiza un cliente según los parámetros que recibe por formulario. Redirige a la vista client.html
		String deleteClient (Model model, @PathVariable String dniClient)	Elimina un cliente de la BBDD. Redirige a la vista client.html
		String showAllClients (Model model)	Lista todos los clientes de la BBDD. Redirige a la vista clientList.html
		String searchClientByDni (Model model, final String dniClient)	Busca un cliente de la BBDD por su DNI. Redirige a la vista cardClient.html o notFound.html si no se encuentra el cliente
		String showAllProductsBought (Model model, final String dniClient)	Lista todos los productos comprados por un cliente. Redirige a la vista productsClientList.html
		String showStationariesFromClient (Model model, final String dniClient)	Lista todas las papelerías a las que ha acudido un cliente. Redirige a la vista stationeriesClientList.html

CONTROLADORES			
INTERFAZ	IMPLEMENTACIÓN	MÉTODOS	DESCRIPCIÓN
ProductControllerI	ProductController	String productPage()	Dirige a la vista product.html
		String addOrUpdateProduct(ProductDto productDto)	Añade o actualiza un producto según los parámetros que recibe por formulario. Redirige a la vista product.html
		String deleteProduct(Model model, @PathVariable String nameProduct)	Elimina un producto de la BBDD. Redirige a la vista product.html
		String showAllProducts(Model model)	Lista todos los productos de la BBDD. Redirige a la vista productList.html
		String searchProductByName(Model model, String nameProduct)	Busca un producto de la BBDD por su nombre. Redirige a la vista cardProduct.html o notFound.html si no se encuentra el producto
ProviderControllerI	ProviderController	String providerPage()	Dirige a la vista provider.html
		String addOrUpdateProvider(ProviderDto providerDto)	Añade o actualiza un proveedor según los parámetros que recibe por formulario. Redirige a la vista provider.html
		String deleteProvider(Model model, @PathVariable String cifProvider)	Elimina un proveedor de la BBDD. Redirige a la vista provider.html
		String showAllProviders(Model model)	Lista todos los proveedores de la BBDD. Redirige a la vista providerList.html
		String searchProviderByCif(Model model, String cifProvider)	Busca un proveedor de la BBDD por su CIF. Redirige a la vista cardProvider.html o notFound.html si no se encuentra el proveedor

CONTROLADORES			
INTERFAZ	IMPLEMENTACIÓN	MÉTODOS	DESCRIPCIÓN
StationeryControllerI	StationeryController	String stationeryPage()	Dirige a la vista stationery.html
		String addOrUpdateStationery(StationeryDto stationeryDto)	Añade o actualiza una papelería según los parámetros que recibe por formulario. Redirige a la vista stationery.html
		String deleteStationery(Model model, @PathVariable String nameStationery)	Elimina una papelería de la BBDD. Redirige a la vista stationery.html
		String showAllStationeries(Model model)	Lista todas las papelerías de la BBDD. Redirige a la vista stationeryList.html
		String searchStationeryByName(Model model, String nameStationery)	Busca una papelería de la BBDD por su nombre. Redirige a la vista cardStationery.html o notFound.html si no se encuentra la papelería

3.5 - Vistas

Las vistas del proyecto se encuentran en la carpeta *src/main/resources/templates*.

NOTA: Se recomienda para una mejor visualización de las vistas la ejecución del script que se encuentra en *src/main/resources* llamado “inserts”, que inserta datos en las tablas generadas por la aplicación. En la aplicación no se ha implementado las funciones de que un cliente compre un producto o vaya a una papelería. Este script añade estos datos que podrá visualizarlos a acceder desde la vista correspondiente.

VISTAS	
VISTA (.html)	DESCRIPCIÓN
home	Página principal de la aplicación
client	Página principal de los clientes
clientList	Página que muestra los clientes de la BBDD listados en una tabla
cardClient	Página que muestra el perfil de un cliente concreto
productClientList	Página que muestra los productos comprados por un cliente listados en una tabla
stationeriesClientList	Página que muestra las papelerías visitadas por un cliente listadas en una tabla
product	Página principal de los productos
productList	Página que muestra los productos de la BBDD listados en una tabla
cardProduct	Página que muestra el perfil de un producto concreto
provider	Página principal de los proveedores
providerList	Página que muestra los proveedores de la BBDD listados en una tabla
cardProvider	Página que muestra el perfil de un proveedor concreto

VISTAS	
VISTA (.html)	DESCRIPCIÓN
stationery	Página principal de las papelerías
stationeryList	Página que muestra las papelerías de la BBDD listados en una tabla
cardStationery	Página que muestra el perfil de una papelería concreta
notFound	Página que se muestra cuando no se encuentra un elemento
Errors	Página que se muestra cuando se produce cualquier error