

Configuraciones necesarias para ejecutar BazarBoost

Las configuraciones que se necesitan realizar son dos:

- Actualizar la ruta de imágenes
- Actualizar la configuración de conexión a la base de datos
- Creación del esquema y la carga de datos de prueba.

A continuación se detalla cómo realizar ambas configuraciones.

1. Actualizar la ruta de imágenes:

Debe definir una ruta absoluta válida en su entorno en el archivo `application.properties`. Actualice la propiedad de la siguiente manera:

Ruta para imágenes:

```
app.imagenes.directorio={su_ruta}
```

Ejemplo:

```
app.imagenes.directorio=C:/Users/Alumno/Documents/ArchivosDiplomadoJava/Bazarboost/bazarboost/src/main/resources/estaticos/img
```

2. Configurar la conexión a la base de datos:

Defina el puerto, usuario y contraseña correspondientes a su base de datos en `application.properties`:

Configuración de la conexión a la base de datos

```
spring.datasource.url=jdbc:mariadb://localhost:{puerto_de_la_bd}/bazarboost
```

```
spring.datasource.username={su_usuario}
```

```
spring.datasource.password={su_contraseña}
```

```
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
```

```
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
```

Reemplace `{su_ruta}`, `{puerto_de_la_bd}`, `{su_usuario}` y `{su_contraseña}` con los valores específicos de su entorno para garantizar la correcta configuración.

3. Creación del esquema y carga de datos de prueba

Ejecute la clase de pruebas BazarBoostTests.java ubicada dentro del directorio **/test**. Esta clase ejecutará los archivos **schema.sql** y **data.sql** ubicados en el directorio **/resources**, dichos archivos SQL crearán el esquema de la base de datos y la poblarán con datos de prueba.

Clases de relevancia en el proyecto

Las tablas que elegí para la creación del CRUD son:

- Categorías
- Descuentos
- MetodosPago
- Direcciones

Por lo tanto, las clases de interés en el proyecto Spring Boot son las siguientes:

- Categoria
 - CategoriaController (para renderizado de plantillas)
 - CategoriaRestController (para API RESTful)
 - CategoriaService
 - CategoriaServiceImpl
 - CategoriaRepository
 - Categoria
 - CategoriaBaseDTO
 - CategoriaCreacionDTO
 - CategoriaEdicionDTO
- Descuento
 - DescuentoController (para renderizado de plantillas)
 - DescuentoRestController (para API RESTful)
 - DescuentoService
 - DescuentoServiceImpl
 - DescuentoRepository
 - Descuento
 - DescuentoVendedorDTO
- MetodosPago
 - MetodoPagoController (para renderizado de plantillas)
 - MetodoPagoRestController (para API RESTful)
 - MetodoPagoService
 - MetodoPagoServiceImpl
 - MetodoPagoRepository
 - MetodoPago
 - MetodoPagoDTO

- MetodoPagoBaseDTO
 - MetodoPagoCreacionDTO
 - MetodoPagoEdicionDTO
- Direcciones
 - DireccionController (para renderizado de plantillas)
 - DireccionRestController (para API RESTful)
 - DireccionService
 - DireccionServiceImpl
 - DireccionRepository
 - Direccion
 - DireccionDTO
 - DireccionBaseDTO
 - DireccionCreacionDTO
 - DireccionEdicionDTO