

# Sistema de muebleria

Nombre: Juan Nuñez Urrea

[https://github.com/J4un/Juan\\_Nu-ez\\_Evaluacion3](https://github.com/J4un/Juan_Nu-ez_Evaluacion3)

Este proyecto se centra en tomar el sistema backend de gestión de mueblería y crear la interfaz de usuario, además el poder dockerizar el proyecto, permitiendo así la máxima compatibilidad del programa y que este se ejecute en cualquier máquina.

Para lograrlo, se utilizan un conjunto de tecnologías además del código base de java: Spring Boot para el desarrollo, MySQL como base de datos y un patrón de arquitectura por capas para asegurar un código legible y escalable.

## Tecnologías Utilizadas:

Java 21

Spring Boot (Web, Data JPA, DevTools)

MySQL (base de datos)

Maven (gestión de dependencias)

JUnit 5 y Mockito (testeo de las operaciones en Service)

Postman (testeo CRUD simple)

Docker

Thymeleaf

## Antes de ejecutar

### Requisitos del sistema:

- Java 21 instalado y configurado como variable del sistema.
- IDE de preferencia para la ejecución de clases “.java”. Ej: IntelliJ IDEA 2024.1.2 o superior (Recomendado)
- Docker instalado
- Navegador de preferencia para visualización gráfica del programa

## Ejecución del programa

1. Clonar o descargar y extraer el repositorio y navegar a la carpeta raíz
2. Abrir la consola en esa carpeta raíz
3. Ejecutar el siguiente comando y esperar un par de minutos a que se ejecute correctamente:

```
docker-compose up --build
```

```
main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page template: index
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
main] com.muebleria.MuebleriaApplication      : Started MuebleriaApplication in 19.921 seconds (process running for 22.6)
```

4. Una vez se lee “Started MuebleriaAplication” se puede acceder a <http://localhost:8080>
5. Para cerrar se puede hacer cerrando la consola con Ctrl+C o utilizando el comando:

```
docker-compose down
```

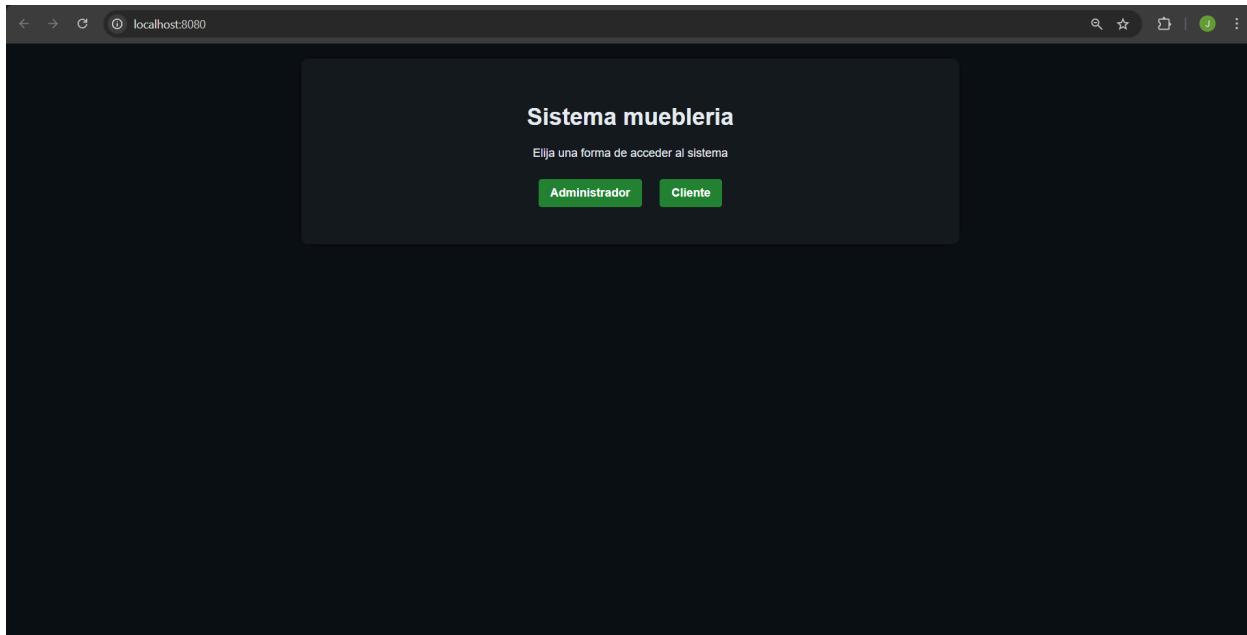
Notas:

Una vez creada la build del programa, se puede saltar el proceso de build simplemente utilizando el comando:

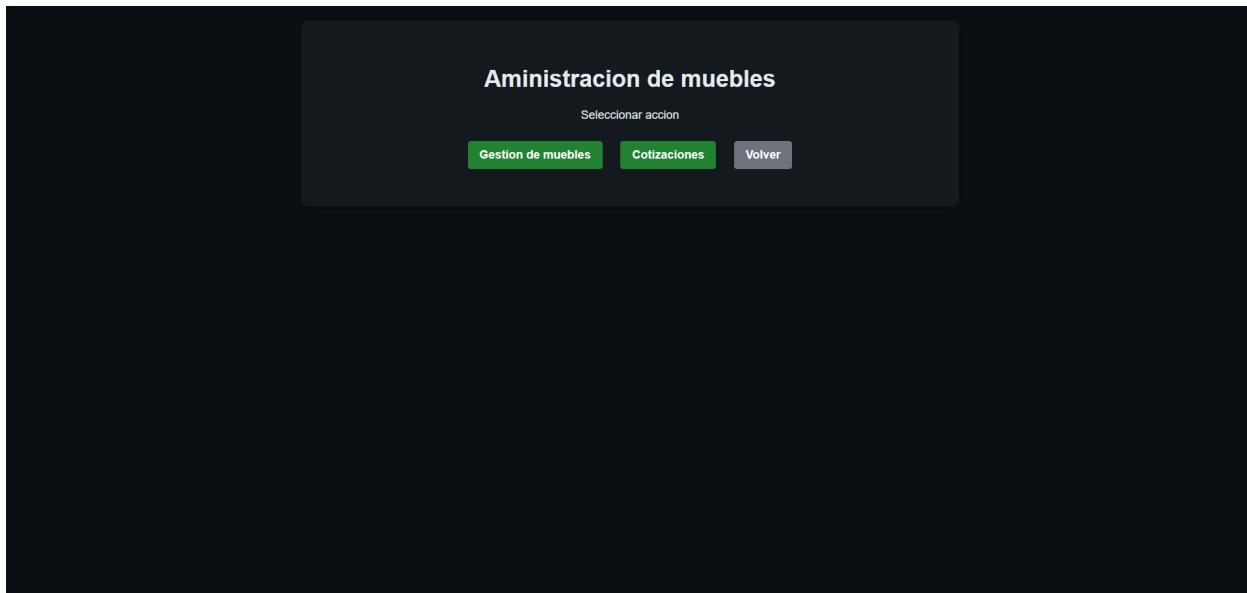
```
docker-compose up
```

## GUI

El sistema cuenta con una interfaz de usuario, la pantalla inicial permite elegir distintos tipos de usuarios para probar distintas funcionalidades



El administrador (Gustavo en este caso) cuenta con las opciones para administrar los muebles y confirmar las cotizaciones:

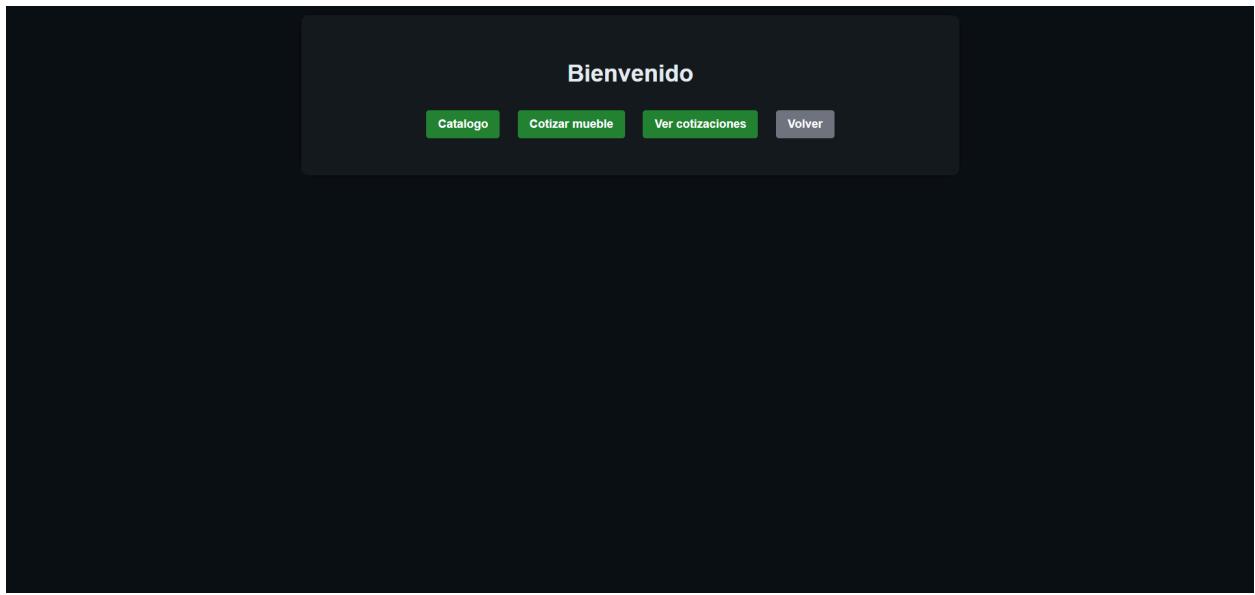


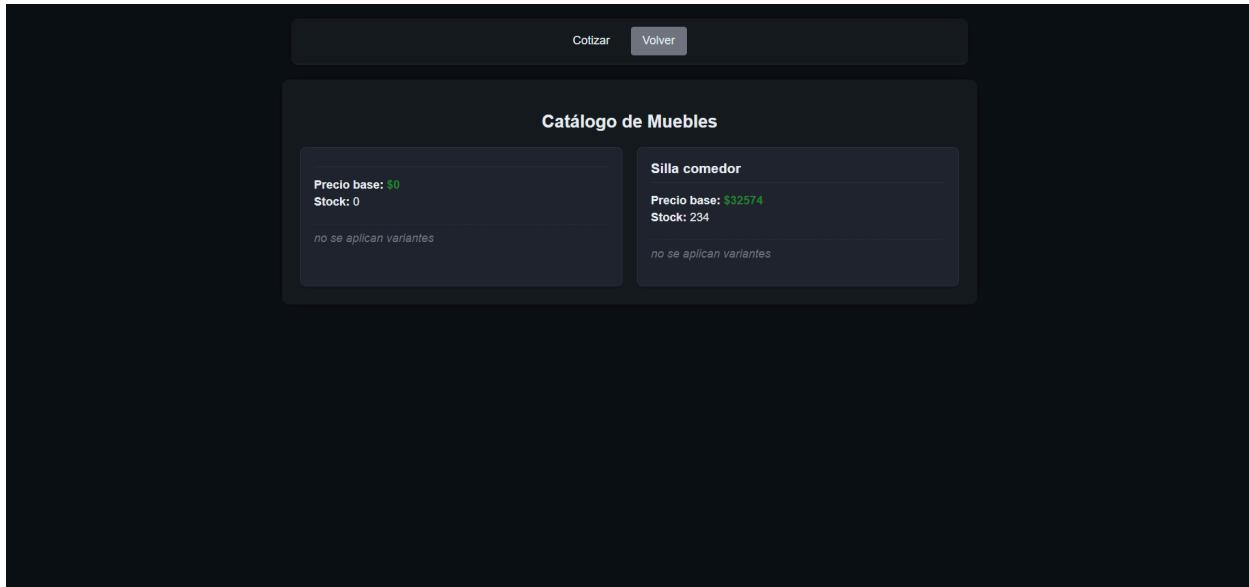
En la gestion de muebles puede modificar, agregar y cambiar las variantes de cada mueble...

The screenshot shows a dark-themed user interface for managing furniture. At the top, there's a header with a "Volver" button. Below it is a section titled "Gestion de muebles" with a "Crear mueble" button. A table lists two furniture items:

ID	Nombre	Stock	Precio	Accion
1		0	\$0	<button>Editar</button> <button>Variantes</button>
2	Silla comedor	234	\$32574	<button>Editar</button> <button>Variantes</button>

...Que luego el cliente puede ver en el catalogo, o puede realizar la cotización, y ver su estado en “Ver cotizaciones”





Problemas comunes:

#### En caso de PortError:

El programa puede advertir que el port configurado por defecto 8080 puede estar ocupado por otra aplicación, por lo que en este caso es recomendable cerrar la aplicación que esté utilizando el puerto en ese momento antes de ejecutar el programa.

#### Docker error:

El programa puede advertir que docker no está instalado o configurado. Es necesario asegurarse de abrir docker antes de ejecutar el programa, de lo contrario no encontrará docker en el sistema

#### .env no deja descargar

Si github no deja descargar el archivo .env, basta con crear un archivo llamado .env sin extensión en esa misma carpeta

Pegar y guardar:

```
MYSQL_ROOT_PASSWORD=mypassword
MYSQL_DATABASE=muebleria
MYSQL_USER=muebleria_user
MYSQL_PASSWORD=muebleria_password
DB_HOST=db
```

Esas son las credenciales por defecto del proyecto.