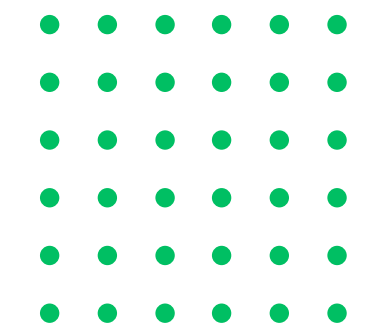


# PROYECTO WEB

Tienda virtual



## PROBLEMA A RESOLVER

Consumidores interesados en el cuidado del medio ambiente y en la economía circular enfrentan dificultades al intentar adquirir productos reutilizables o sustentables.

## SOLUCIÓN PROPUESTA

Se propone el desarrollo de Second Shop, una tienda virtual especializada en la venta de productos de segunda mano y sustentables.

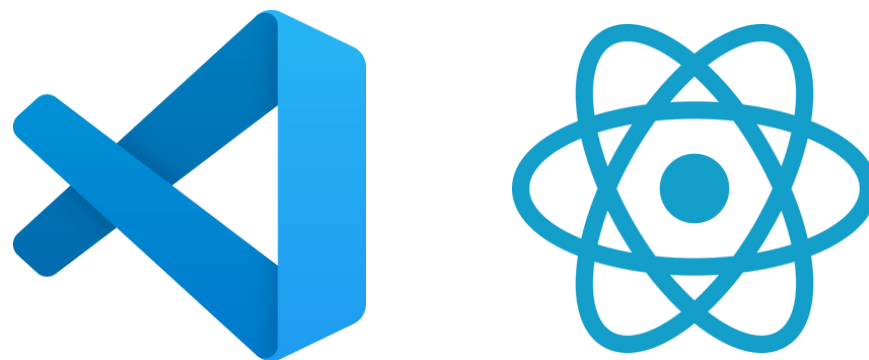


# ÁREA DE TRABAJO: ENTORNO DE DESARROLLO



## Backend

IntelliJ IDEA con Java 17,  
Spring Boot 3.4.4, Maven



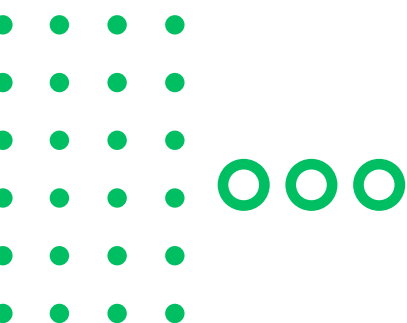
## Frontend

Visual Studio Code con  
React.js



## Base de datos

MySql con DBeaver

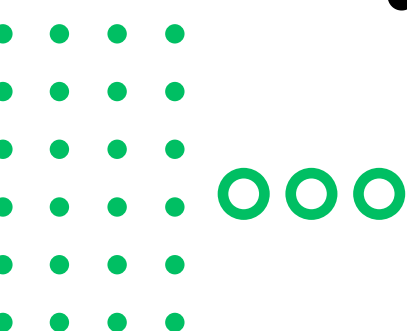
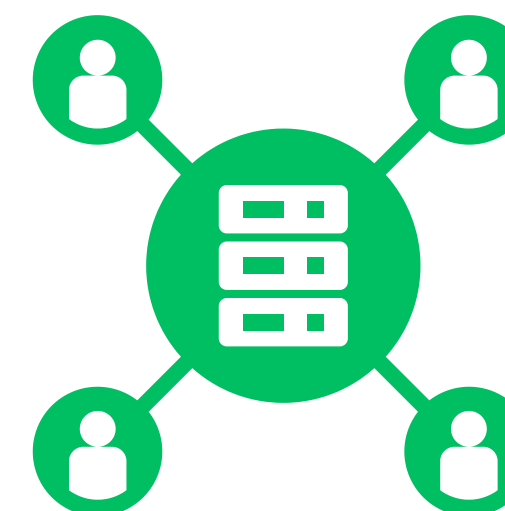
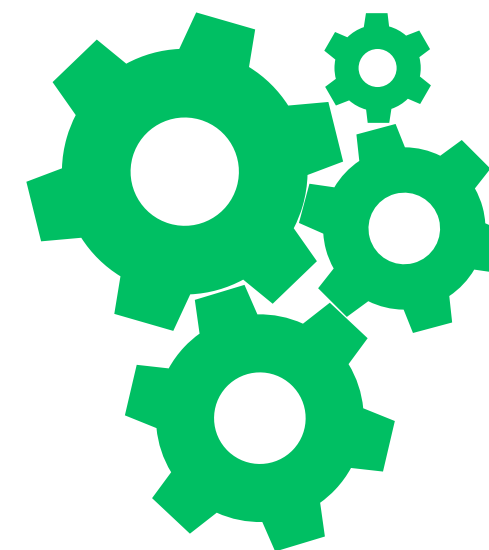


Te quiero mucho  
intellij



## FUNCIONALIDADES ESPERADAS

- Registro y login de usuarios con verificación por token.
- Panel de administración para gestionar productos y usuarios.
- Visualización de productos disponibles.
- Agregar productos al carrito.
- Protección de rutas (seguridad en frontend y backend)
- Gestión de usuarios y roles (usuario, administrador)



# FUNCIONALIDAD DEL BACKEND



- **Configuración:** Contiene las clases de configuración de seguridad, como filtros JWT, reglas de autorización y punto de entrada para errores.
- **Controller:** Controladores REST que manejan las peticiones HTTP. Son la puerta de entrada del sistema.
- **Listener:** Contiene eventos personalizados y sus listeners, usados en funcionalidades como el registro de usuarios y la activación de cuentas.
- **Modelos:** Contiene las entidades del sistema: representan las tablas de la base de datos (Carrito, Producto, Usuario, etc.).
- **Repo:** Contiene interfaces que extienden JpaRepository o CrudRepository. Son los accesos a la base de datos.
- **Service:** Contiene la lógica de negocio. Los controladores llaman a los servicios para realizar operaciones.
- **Token:** Gestión de tokens de verificación para confirmar cuentas.
- **Util:** Clases utilitarias, como la que maneja JWT (JwtUtils).



# JWT



# EJEMPLO DE FUNCIONAMIENTO DE PETICIÓN

**SecurityFilterChain:** Verifica si la petición tiene el permiso pertinente

```
.requestMatchers("/api/auth/**").permitAll()
```



**UsuarioController:** Recibir peticiones HTTP del frontend.

```
actualizarUsuario(id, usuarioActualizado)
```



**UsuarioService:** Definir operaciones lógicas.

```
actualizarUsuario(id, usuario)
```



**UsuarioServiceImpl:** Implementar la lógica de negocio.

```
actualizarUsuario(id, usuario) {  
    ...  
}
```



**UserRepo:** Interactuar con la base de datos.

```
+ findById(id)  
+ save(usuario)
```

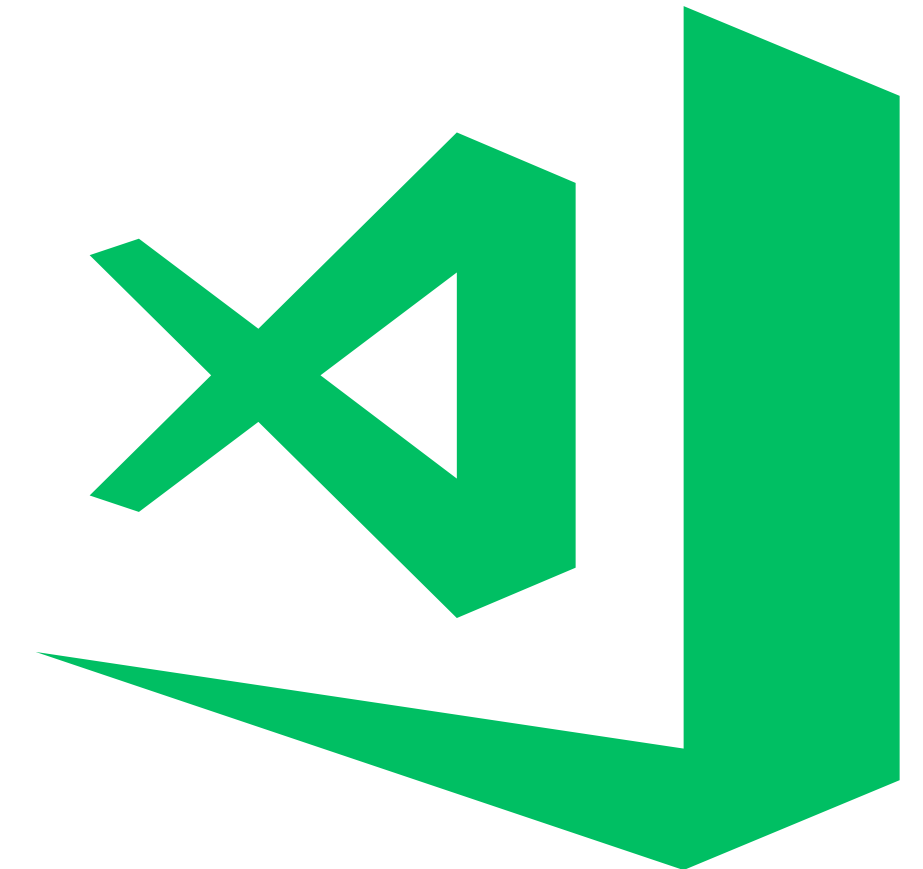


**Usuario:** Representa los datos del usuario (nombre, email, contraseña...).



## ESTRUCTURA DEL FRONTEND

- **Componentes:** Este contiene bloques reutilizables de la interfaz de usuario que son unidades funcionales o visuales independientes dentro de la aplicación.
- **Imágenes:** Folder con todas las imágenes que fueron usadas en componentes o vistas.
- **Services:** Este folder contiene el archivo principal para las comunicaciones con el backend, especialmente del perfil de usuario. Facilitada con 'axios'
- **Stylesheets:** Carpeta que contiene todos los respectivos CSS para los componentes y vistas de usuario. Posteriormente se importan en el componente o vista.
- **Views:** Este contiene los componentes que representan páginas completas que agrupa componentes.
- **App.js:** Esta es la estructura base del frontend. Es el componente principal donde se monta la aplicación. Principalmente usado para definir las rutas de vistas completas para conectarlas.



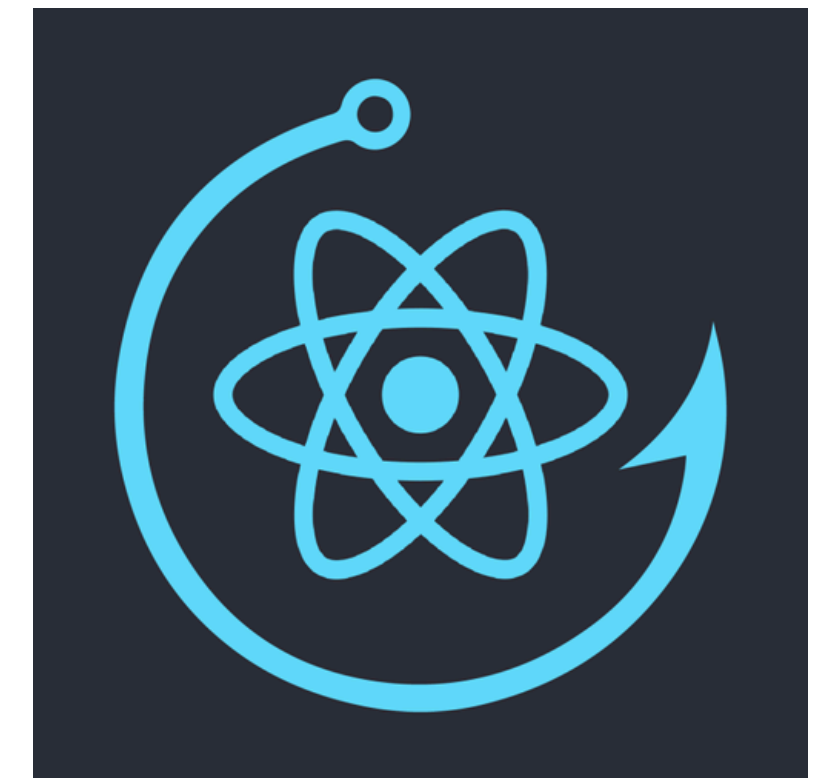
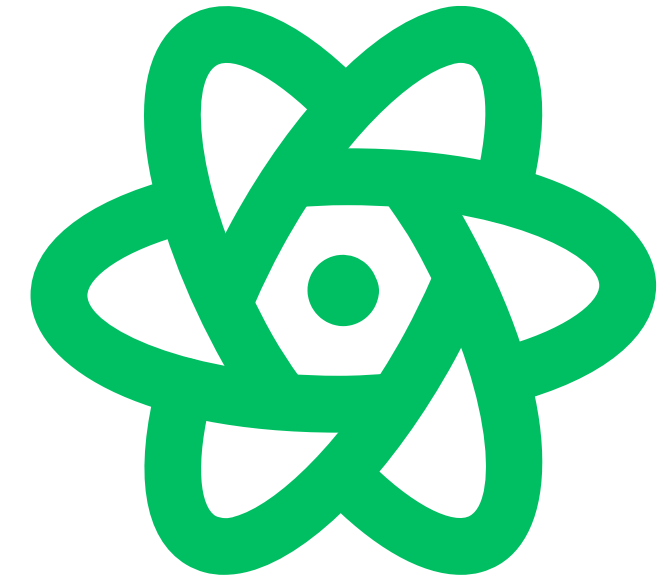
# Bootstrap



# HOOKS PRINCIPALES

## HOOK

El “hook” en React son funciones especiales que permiten usar características de React como el estado o ciclo de vida.



- **useState:** Permite crear y manejar variables dentro de un componente react. Guarda un dato que puede cambiar con el tiempo.
- **useEffect:** Ejecuta código cuando el componente se monta, actualiza o desmonta.
- **Link:** Crea un enlace para navegar entre páginas sin recargar el navegador. Por ejemplo un botón o texto que lleva de la página principal a la de productos.
- **useNavigate:** Esta herramienta permite cambiar de página desde el código, sin que el usuario haga clic en un enlace.



ooo

# VISUALIZACIÓN DE CÓDIGOS Y DEL RESULTADO FINAL

•  
•  
•  
•  
•  
•  
•