



## **Diagrama de componentes**

### **Integrantes**

Esteban Hernando Pedraza Solano

Sebastián Sandoval Dueñas

### **Asignatura**

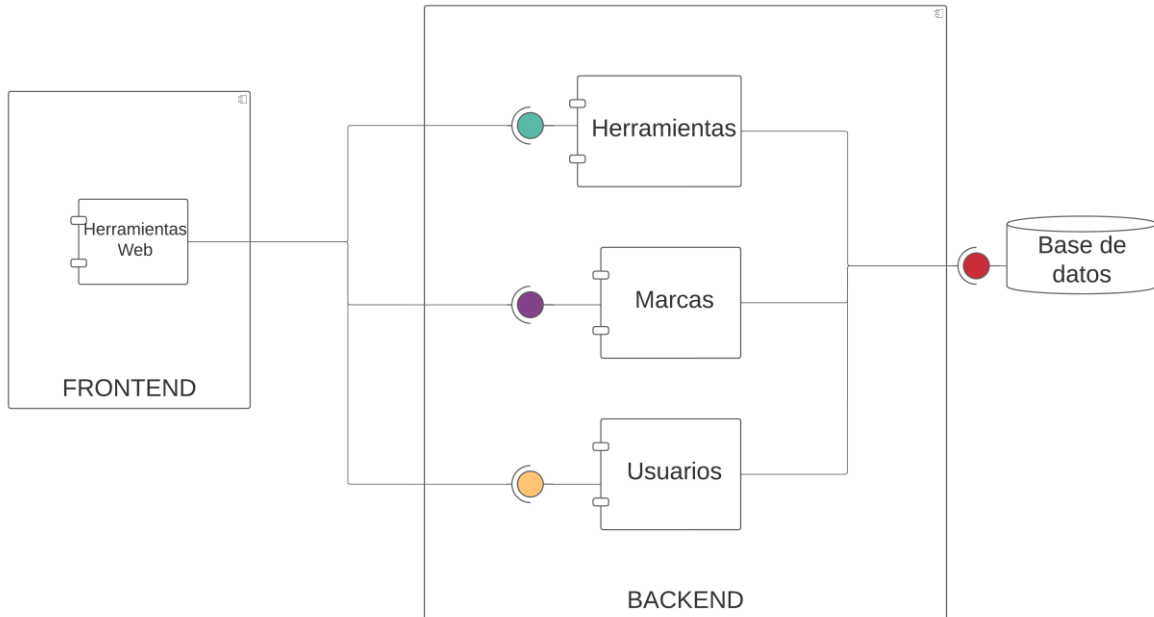
Desarrollo Web

### **Profesor**

John Faber Ramírez Castillo

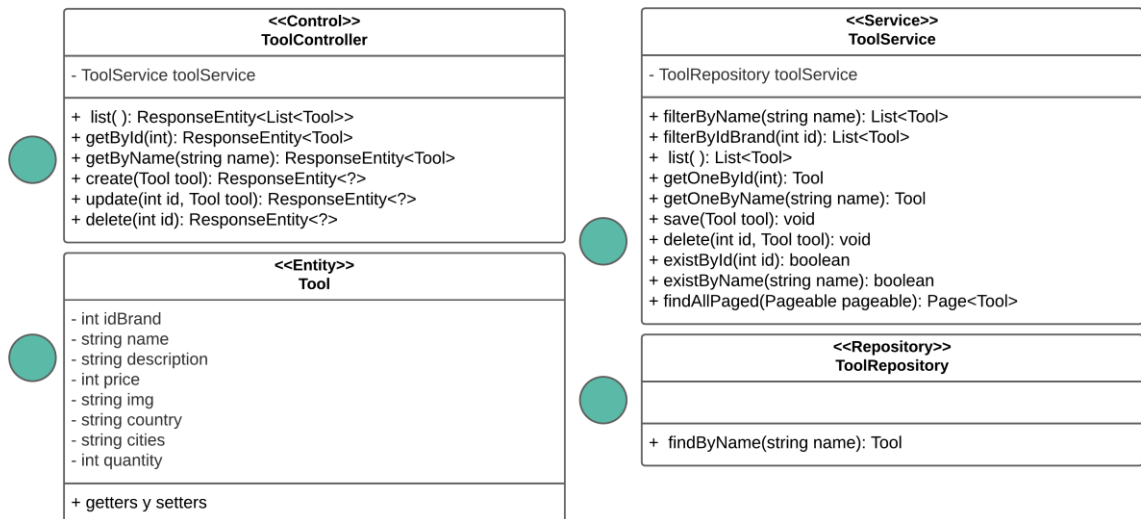
24 de mayo de 2023

## Diagrama de componentes



En este diagrama podemos visualizar las conexiones presentes en el desarrollo del proyecto, y a su vez, se puede representar como están repartidos los componentes y como se comunican entre sí. Además de representar como se debería desplegar cada uno en caso de que su comportamiento se desee independiente entre sí.

### - Componente de Herramientas:



El microservicio "Tool" es un componente que representa la gestión de las herramientas que se utilizan en el aplicativo Web. El código componente consta de varias partes que describen la entidad, su repositorio, su controlador y su servicio.

La entidad "Tool" representa una herramienta y tiene los siguientes atributos:

- id: un entero que representa el ID de la herramienta.
- idBrand: un entero que representa el ID de la marca de la herramienta.
- name: una cadena de texto que representa el nombre de la herramienta.
- description: una cadena de texto que representa la descripción de la herramienta.
- price: un entero que representa el precio de la herramienta.
- img: una cadena de texto que representa la imagen de la herramienta.
- country: una cadena de texto que representa el país de origen de la herramienta.
- cities: una cadena de texto que representa las ciudades en las que se encuentra disponible la herramienta.
- quantity: un entero que representa la cantidad disponible de la herramienta.

La entidad Tool tiene dos constructores: uno sin argumentos y otro con los datos de la herramienta.

El repositorio "ToolRepository" es una interfaz que extiende JpaRepository y proporciona métodos para realizar operaciones de acceso a la base de datos relacionadas con la entidad Tool. Incluye el método "findByName" que busca una herramienta por su nombre.

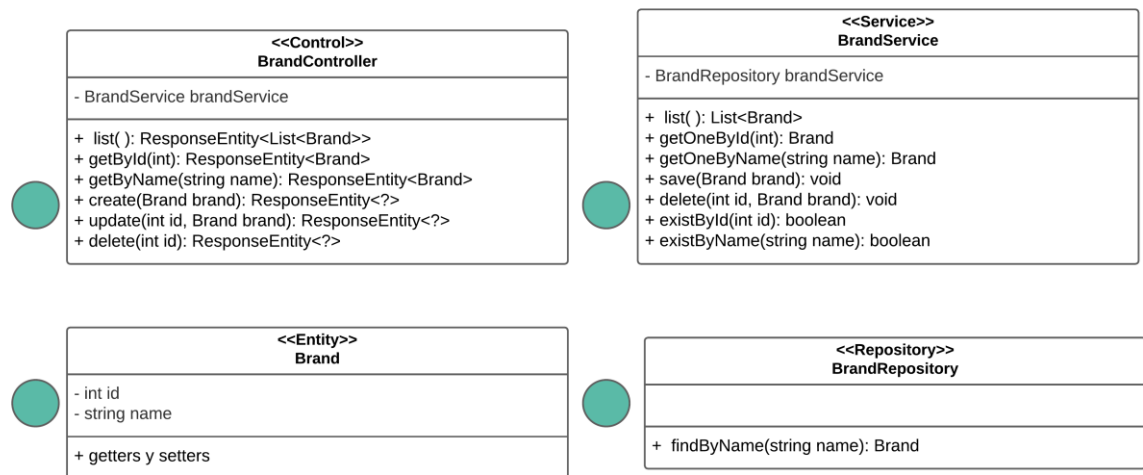
El controlador "ToolController" es responsable de manejar las solicitudes HTTP relacionadas con las herramientas. Contiene varios métodos de punto final que realizan operaciones CRUD (crear, leer, actualizar y eliminar) en las herramientas:

- "list": un método que devuelve una lista de todas las herramientas.
- "getById": un método que obtiene una herramienta por su ID.
- "getByName": un método que obtiene una herramienta por su nombre.
- "create": un método que crea una nueva herramienta.
- "update": un método que actualiza una herramienta existente.
- "delete": un método que elimina una herramienta por su ID.
- "filterByName": un método que filtra las herramientas por nombre.
- "filterByBrandId": un método que filtra las herramientas por ID de marca.
- "searchPaged": un método que obtiene una lista paginada de herramientas.

El servicio "ToolService" implementa la lógica de negocio relacionada con las herramientas. Proporciona métodos para realizar operaciones como filtrar herramientas por nombre y obtener una lista paginada de herramientas.

En resumen, el microservicio "Tool" se encarga de gestionar las herramientas, proporcionando operaciones CRUD, filtrado y paginación a través de su controlador y utilizando el repositorio y el servicio para acceder y manipular los datos de la herramienta en la base de datos.

### Componente de Marcas:



El microservicio "Brand" es una parte de un sistema más grande que se encarga de gestionar y administrar marcas. Proporciona funcionalidades para crear, actualizar, eliminar y obtener información sobre marcas específicas. A continuación, se describe cada componente del microservicio:

#### 1. Controlador (BrandController):

El controlador es responsable de manejar las solicitudes HTTP relacionadas con las marcas. Proporciona los siguientes puntos finales (endpoints) para interactuar con el microservicio:

- GET /brand/: Un punto final de ejemplo que devuelve un saludo.
- GET /brand/list: Obtiene la lista de todas las marcas existentes.
- GET /brand/detail/{id}: Obtiene los detalles de una marca según su ID.
- GET /brand/detailname/{name}: Obtiene los detalles de una marca según su nombre.
- POST /brand/create: Crea una nueva marca.
- PUT /brand/update/{id}: Actualiza los detalles de una marca existente.
- DELETE /brand/delete/{id}: Elimina una marca existente.

#### 2. Repositorio (BrandRepository):

El repositorio define las operaciones de persistencia para la entidad "Brand". Proporciona métodos para buscar y recuperar marcas de la base de datos, en este caso utilizando Spring Data JPA.

### 3. Entidad (Brand):

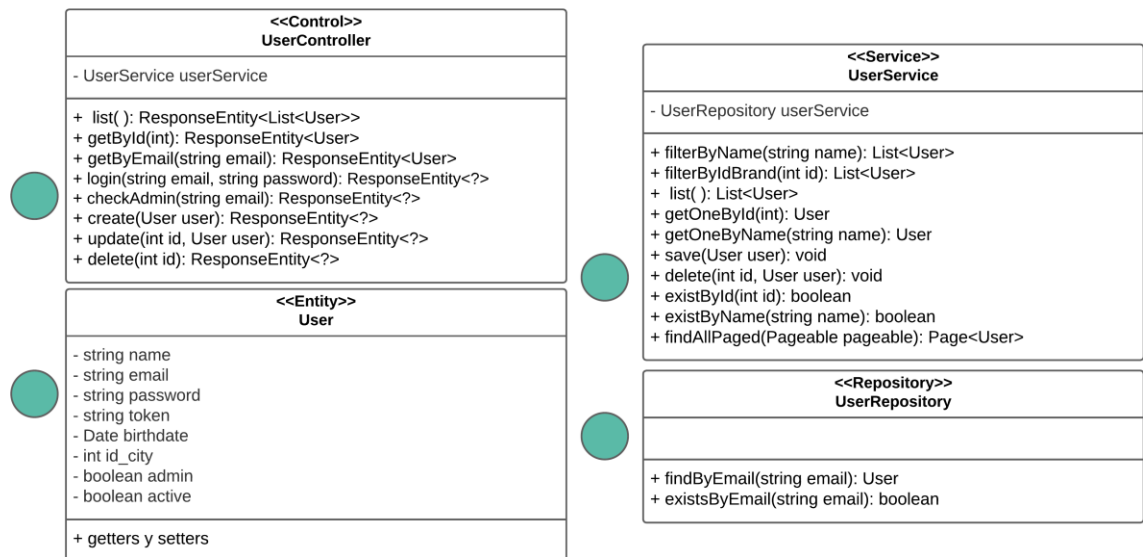
La entidad "Brand" representa una marca y se mapea a una tabla en la base de datos. Tiene atributos como id (identificador único) y name (nombre de la marca). La anotación @Entity indica que es una entidad JPA y la anotación @Table especifica el nombre de la tabla asociada.

### 4. Servicio (BrandService):

El servicio encapsula la lógica de negocio relacionada con las marcas. Proporciona métodos para realizar operaciones como listar marcas, obtener una marca por su ID o nombre, guardar una nueva marca, eliminar una marca existente, y verificar si una marca existe por su ID o nombre. Utiliza el repositorio de marcas para interactuar con la capa de persistencia.

En resumen, el microservicio "Brand" permite realizar operaciones CRUD (crear, leer, actualizar, eliminar) en marcas. Proporciona una API RESTful para interactuar con el servicio y utiliza Spring Boot, JPA y una base de datos para almacenar y recuperar datos de marcas.

## - Componente de Usuarios:



El microservicio de Usuarios es un sistema amplio que se encarga de gestionar y manejar las operaciones relacionadas con los usuarios de la aplicación Web de herramientas. Proporciona funcionalidades para el registro, autenticación, administración y gestión de usuarios dentro del sistema.

Este microservicio está diseñado para interactuar con otros componentes del sistema a través de una API RESTful. Utiliza tecnologías como Spring Boot y JPA (Java Persistence API) para implementar la lógica de negocio y persistir los datos en una base de datos.

La funcionalidad principal del microservicio de Usuarios incluye:

1. Registro de usuarios: Permite a los usuarios registrarse en el sistema proporcionando información como nombre, correo electrónico, contraseña, fecha de nacimiento, ciudad, etc.
2. Autenticación de usuarios: Proporciona un mecanismo de inicio de sesión para que los usuarios puedan acceder a su cuenta. Verifica las credenciales del usuario (correo electrónico y contraseña) y devuelve un token de acceso para las futuras solicitudes.
3. Gestión de perfiles de usuario: Permite a los usuarios actualizar su información de perfil, como nombre, correo electrónico, contraseña, fecha de nacimiento, ciudad, etc.
4. Consulta de usuarios: Proporciona operaciones para obtener información de usuarios específicos, ya sea por su ID o su dirección de correo electrónico.
5. Gestión de usuarios administradores: Permite verificar si un usuario tiene privilegios de administrador y realizar acciones específicas para ellos, como acceso a ciertas funcionalidades adicionales.
6. Listado de usuarios: Permite obtener una lista paginada de usuarios registrados en el sistema.
7. Desactivación de usuarios: Proporciona la capacidad de desactivar usuarios, lo que implica que no podrán iniciar sesión ni realizar acciones dentro del sistema.

Este microservicio utiliza un modelo de datos que incluye la entidad "User" (Usuario) con atributos como nombre, correo electrónico, contraseña, token, fecha de nacimiento, ciudad, indicador de administrador y estado de activación.

En resumen, el microservicio de Usuarios se encarga de la gestión de usuarios en un sistema, proporcionando funcionalidades esenciales como registro, autenticación, consulta y actualización de perfiles, gestión de privilegios administrativos y más.