

---

# API REST – Backend (Spring Boot)

**Lenguaje:** Java 17

**Framework:** Spring Boot

**Seguridad:** JWT

**Base de datos:** MySQL

**ORM:** Spring Data JPA

**Documentación:** Swagger

## Paquetes principales

- **controller**: Controladores REST.
- **entities**: Entidades JPA del modelo de datos.
- **dto**: Data Transfer Objects para entrada/salida.
- **repository**: Interfaces para acceso a la base de datos.
- **service**: Lógica de negocio (implementa **IServiceGeneric**).
- **mapper**: Conversores entre entidades y DTOs (MapStruct).
- **security**: Configuración de autenticación/autorización con JWT.

## Seguridad

- Autenticación con tokens JWT.
- Control de acceso por rol (**ROLE\_ADMIN**, **ROLE\_USER**).
- Filtros personalizados (**JwtFilter**, **SecurityConfig**).

## Testing

- JUnit 5 y Mockito para tests de:
  - Repositorios.
  - Servicios.
  - Controladores.
  - Mappers.

Cuenta con una cobertura de código del 89% generada por JaCoCo

# CanaryTrails

Element	Missed Instructions	Cov
<a href="#">es.iespuertodelacruz.mp.canarytrails.controller.v2</a>	<div><div></div></div>	78 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.mapper</a>	<div><div></div></div>	92 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.service</a>	<div><div></div></div>	90 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.controller.v3</a>	<div><div></div></div>	90 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.entities</a>	<div><div></div></div>	83 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.ruta</a>	<div><div></div></div>	77 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.usuario</a>	<div><div></div></div>	81 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.comentario</a>	<div><div></div></div>	86 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.municipio</a>	<div><div></div></div>	92 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.coordenada</a>	<div><div></div></div>	86 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.zona</a>	<div><div></div></div>	86 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.fauna</a>	<div><div></div></div>	98 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.security</a>	<div><div></div></div>	99 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.config</a>	<div><div></div></div>	100 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.flora</a>	<div><div></div></div>	100 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.controller.v1</a>	<div><div></div></div>	100 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.usuario.auth</a>	<div><div></div></div>	100 %
<a href="#">es.iespuertodelacruz.mp.canarytrails.dto.rutafavorita</a>	<div><div></div></div>	100 %
<a href="#">es.iespuertodelacruz.mp.canarytrails</a>	<div><div></div></div>	100 %
Total	1.243 of 11.447	89 %

## App Móvil – React Native

**Lenguaje:** JavaScript / TypeScript

**Framework:** React Native

**Librerías clave:**

- `react-navigation` → Navegación por pantallas.
- `axios` → Llamadas a la API.
- `react-native-maps` → Visualización de rutas.
- `redux` (opcional) → Gestión de estado.
- `i18n` → Multilenguaje.
- `expo-file-system` y `async-storage` → Modo offline.

### Pantallas principales

- Registro / Login
- Explorar rutas
- Ficha de ruta
- Mapa GPS en tiempo real
- Subir ruta
- Flora / Fauna asociada
- Perfil

# Sistema Administrativo – Dolibarr

**Propósito:** Gestión de la empresa que comercializa la app.

**Configuraciones básicas:**

- Gestión de clientes y proveedores.
- Productos/servicios (ej. planes premium o merchandising).
- Presupuestos, facturación y pedidos.
- Logotipo y diseño personalizado.

**Ruta recomendada de acceso:**

<http://localhost/dolibarr/> o [/var/www/html/dolibarr/](http://var/www/html/dolibarr/)

---

## Pruebas y Calidad

- Cobertura de código con JaCoCo.
  - Validación de DTOs con [javax.validation](#).
  - Tests para control de errores ([@ControllerAdvice](#)).
  - CI/CD opcional con GitHub Actions (configurable).
- 

## Despliegue

En el futuro:

- **Backend:** Puede ser desplegado DigitalOcean.
  - **App:** Compilar como APK para Android o IPA para iOS.
  - **Admin:** Dolibarr puede instalarse en servidor web Apache o Nginx.
- 

## Observaciones Técnicas

- Se utiliza una tabla polimórfica [fotos](#) para manejar imágenes de rutas, fauna y flora.
  - Todas las relaciones están correctamente modeladas con claves foráneas y tablas intermedias.
  - Controladores REST siguen convención RESTful: [GET](#), [POST](#), [PUT](#), [DELETE](#).
- 

## Enlaces útiles

- [Manual de uso](#)
  - [Manual de usuario](#)
  - [Diseño UI/UX](#)
- 

## App Móvil – React Native

**Lenguaje:** JavaScript / TypeScript

**Framework:** React Native

**Gestor de paquetes:** npm

**Ecosistema:** Android

---

## Estructura de Carpetas Principal

```
AppCanaryTrails/
├── src/
│   ├── api/ → Configuración de Axios y peticiones a la API
│   ├── components/ → Componentes reutilizables (cards, botones, etc.)
│   ├── constants/ → Estilos globales, colores, iconos, traducciones
│   ├── navigation/ → Stack, tab y drawer navigators
│   ├── screens/ → Pantallas principales (Inicio, Ruta, Perfil, etc.)
│   ├── store/ → Gestión de estado (Redux, Context API u otra)
│   └── utils/ → Funciones auxiliares, validaciones, helpers
├── assets/ → Imágenes, iconos, fuentes
├── App.js → Punto de entrada de la app
├── .env → Variables de entorno
└── package.json
```

---

## Comunicación con el Backend

- Se usa **Axios** como cliente HTTP (`/src/api/api.js`).
- Las rutas de la API están centralizadas en un archivo para facilitar su gestión y multientorno.
- La variable `API_BASE_URL` se carga desde `.env`.

```
import axios from "axios";
const API = axios.create({
  baseURL: process.env.API_BASE_URL || "http://localhost:8080/api/",
});
```