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.

Endpoints

V3:

- Rutas:
    - `/rutas` : Get All
    - `/rutas/{id}`: Get By Id
    - `/rutas/add`: POST para crear rutas
    - `/rutas/update`: PUT para editar rutas
    - `/rutas/upload`: PUT para añadir una foto a la ruta
    - `/rutas/delete/{id}`: DELETE para borrar rutas

-- El resto de Endpoints siguen el mismo formato --

Excepciones:

- Rutas Favoritas:
    - `/rutas_favoritas/{id}`: encuentra las rutas favoritas del usuario con esa `id`
    - `/rutas_favoritas/populares`: da una lista con las rutas que tienene más de 5 favoritos
    - `/rutas_favoritas/add`: añade una ruta favorita a partir de la `id` de ruta y la de usuario
    - `/rutas_favoritas/delete`: elimina una ruta favorita a partir de la `id` de ruta y de usuario

V2:

- Los usuarios solo pueden hacer `Get All`, y `GetByI` **Excepciones**:
    - `/usuarios` que solo se puede obtener el registrado, su propio usuario.

Cambios:

- Los usuarios solo pueden hacer un `PUT` y `DELETE` de las fotos de las rutas / faunas / floras que ellos hayan creado y no estén verificadas.
- Los usuarios solo pueden ejecutar cambios en las rutas favoritas si es sobre el mismo.

V1:

- Auth
  - `/login`: inicia sesión con los datos de un usuario registrado y verificado.
  - `/auth/register`: registra un usuario con un nombre y correo no existente en la bbdd, y le manda un correo de verificación.
  - `/confirmación/correo=&token=`: recibe los datos del correo y el token de verificación en la url y verifica al usuario.

Fotos:

- `/imagenes`: accede a las imágenes según la categoría.
  - `/fauna/nombre.extension`
  - `/flora/nombre.extension`
  - `/usuario/nombre.extension`
  - `/ruta/nombre.extension`
- `nombre.extension`: es como se guarda el campo foto en la bbdd.

Seguridad

- Autenticación con token JWT.
- Control de acceso por rol (`ROLE_ADMIN`, `ROLE_USER`).
- Filtros personalizados (`JwtFilter`, `SecurityConfig`).
- Verificación de cuentas por correo (`Mail Sender`).

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	Co
es.iespuertodelacruz.mp.canarytrails.controller.v2		78 %
es.iespuertodelacruz.mp.canarytrails.mapper		92 %
es.iespuertodelacruz.mp.canarytrails.service		90 %
es.iespuertodelacruz.mp.canarytrails.controller.v3		90 %
es.iespuertodelacruz.mp.canarytrails.entities		83 %
es.iespuertodelacruz.mp.canarytrails.dto.ruta		77 %
es.iespuertodelacruz.mp.canarytrails.dto.usuario		81 %
es.iespuertodelacruz.mp.canarytrails.dto.comentario		86 %
es.iespuertodelacruz.mp.canarytrails.dto.municipio		92 %
es.iespuertodelacruz.mp.canarytrails.dto.coordenada		86 %
es.iespuertodelacruz.mp.canarytrails.dto.zona		86 %
es.iespuertodelacruz.mp.canarytrails.dto.fauna		98 %
es.iespuertodelacruz.mp.canarytrails.security		99 %
es.iespuertodelacruz.mp.canarytrails.config		100 %
es.iespuertodelacruz.mp.canarytrails.dto.flora		100 %
es.iespuertodelacruz.mp.canarytrails.controller.v1		100 %
es.iespuertodelacruz.mp.canarytrails.dto.usuario.auth		100 %
es.iespuertodelacruz.mp.canarytrails.dto.rutafavorita		100 %
es.iespuertodelacruz.mp.canarytrails		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.
- `expo-file-system` y `async-storage` → Modo offline.

📲 Pantallas principales

- Registro / Login
- Explorar rutas
- Información de ruta
- Buscar ruta
- Traking en tiempo real (configurable)
- Crear ruta / fauna / flora
- Flora / Fauna asociada
- Informacion Perfil
- Rutas Favoritas
- Creaciones

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 exponen las fotos desde local a diferentes endpoints mediante un webconfig y `Resources Handler`
- 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
```

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/`