

# Documentación del Backend

## 1. Estructura General del Proyecto

El backend está diseñado para garantizar orden, seguridad y escalabilidad. La organización sigue una estructura clara que facilita el mantenimiento y evita conflictos entre variables o métodos.

### **Tools (Constants)**

Definición de constantes globales en métodos estáticos para unificar valores y evitar conflictos.

### **Models**

Definición de las entidades principales y sus relaciones.

### **Migrations**

Migraciones para la creación, modificación y rollback de tablas en la base de datos.

### **Middleware**

Validaciones y seguridad de acceso.

### **JWT**

Gestión de tokens de acceso y refresh.

### **Jobs**

Tareas pendientes para producción (ejemplo: factor de doble autenticación).

### **DTOs (Data Transfer Objects)**

Validación y transferencia de datos.

### **Controllers**

Manejo de rutas y lógica de negocio.

## 2. Models

Los modelos representan las tablas y relaciones en la base de datos.

User

- Campos principales: id, nombre, email (almacenado siempre en minúsculas para evitar duplicados) y password la cual se maneja mediante BCrypt.

UserInterest (Pivot)

- Relación muchos-a-muchos entre usuarios e intereses.

Interest

- Atributos: id, nombre.
- Relación con usuarios a través de UserInterest (pivot table).
- Incluye un método para limpiar el nombre (evita problemas con mayúsculas, tildes y signos).

Event

- Atributos: id, nombre, estado, fecha, categoría, dirección, latitud, longitud y localización (string para registrar manualmente el lugar).
- Representa un evento creado por un usuario.

EventAttendance

- Relación entre usuarios y eventos.
- Estados: Asistir, No asistir, Pendiente de confirmación.

## 3. Migrations

Se utilizan para la creación y modificación de tablas, con opción de rollback.

- create-users - Tabla de usuarios.
- create-interest - Tabla de intereses.
- create-user-interest - Tabla pivot entre usuarios e intereses.
- create-events - Tabla de eventos.
- create-event-attendances - Tabla de asistencias a eventos.
- populate-date - Inserción inicial de intereses disponibles en el registro.

## 4. Middleware

- AdminMiddleware: Gestión de permisos de administrador (pendiente de implementación).
- ApiKeyMiddleware: Verificación de la API Key del servidor.

## 5. JWT (JSON Web Token)

Gestión completa de tokens:

- Creación de access token y refresh token.
- Verificación de validez y expiración.
- Autenticación segura en todas las rutas protegidas.

## 6. Jobs

Implementación pensada para producción.

- Funcionalidad para doble autenticación.
- Actualmente pendiente, no utilizado en entorno local.

## 7. DTOs (Data Transfer Objects)

Garantizan validación y transferencia de datos en cada capa.

- UserDTO: Validación de datos de usuario.
- InterestDTO: Validación y transferencia de intereses.
- UserInterestDTO: Gestión de asignación de intereses a usuarios.
- EventDTO: Validación y transferencia de eventos.
- EventAttendeesDTO: Validación de asistencias a eventos.
- TokenDTO: Transferencia de access y refresh tokens.

## 8. Controllers

Encargados de manejar las rutas y lógica asociada.

UserController:

- Obtener usuarios.
- Devolver detalles de usuario.
- Eliminar usuarios.

JWTUserAuthenticator:

- Verificar token.
- Buscar usuario en base de datos.

InterestsController:

- Listar intereses (ordenados alfabéticamente).
- Crear, actualizar, buscar ,obtener detalles de un interés y eliminar interés.

EventsController:

- Manejo completo de eventos (creación, detalle, actualización, buscar y eliminación).

EventsAttendesController:

- Gestión de asistencias a eventos (confirmar, rechazar o pendiente).

AuthController:

- Registro y autenticación de usuarios.

## 9. Base de Datos

- SQLite: Utilizado actualmente en entorno local
- PostgreSQL: Utilizado en entorno de producción, pensado para futuro despliegue en dominio.

## 10. Seguridad

- Uso de claves privadas fuera del repositorio (no en variables de entorno del proyecto).
- Garantía de que nunca se expondrán claves en GitHub.
- Proyecto en fase de pruebas, pero con el objetivo de mantener siempre la seguridad en despliegue.