

## All Classes and Interfaces

All Classes and Interfaces	Interfaces	Classes	Enum Classes	Exception Classes
<b>Class</b>				
<b>Description</b>				
<b>AccessDeniedException</b>				
Excepción para acceso denegado (falta de permisos).				
<b>Activity</b>				
Entidad que representa una actividad en un itinerario de viaje.				
<b>ActivityController</b>				
Controlador REST para gestionar las actividades del itinerario de un viaje.				
<b>ActivityDTO</b>				
DTO para la entidad Activity.				
<b>ActivityRepository</b>				
Repositorio para la entidad Activity.				
<b>ActivityService</b>				
Servicio para gestionar las operaciones de las actividades.				
<b>ActivityType</b>				
Enum que representa los tipos de actividades disponibles en un itinerario.				
<b>AddMemberRequestDTO</b>				
DTO para agregar un nuevo miembro a un viaje.				
<b>BackendApplication</b>				
Clase principal para iniciar la aplicación Spring Boot.				
<b>ChangeMemberRoleRequestDTO</b>				
DTO para cambiar el rol de un miembro en un viaje.				
<b>CorsConfig</b>				
Configuración de CORS (Cross-Origin Resource Sharing) Permite que el frontend pueda hacer solicitudes al backend.				
<b>CreateActivityRequestDTO</b>				
DTO para crear una nueva actividad.				
<b>CreateSplitRequestDTO</b>				
DTO para crear una nueva división de gasto.				
<b>CustomUserDetailsService</b>				
Servicio para cargar detalles de usuario desde la base de datos.				
<b>DatabaseSeeder</b>				
Executes the SQL statements contained in <code>data.sql</code> when the database is empty.				
<b>DuplicateResourceException</b>				
Excepción para operaciones duplicadas (ej: usuario ya existe).				
<b>ErrorResponseDTO</b>				
DTO para respuestas de error de la API.				
<b>Expense</b>				
Entidad que representa un gasto en un viaje.				
<b>ExpenseController</b>				
Controlador REST para gestionar gastos en viajes.				
<b>ExpenseCreateRequestDTO</b>				
DTO para crear un nuevo gasto.				

**ExpenseDTO**

DTO para información completa de un gasto.

**ExpenseRepository****ExpenseService****ExpenseSplit**

Entidad que representa la división de un gasto entre un usuario.

**ExpenseSplitController**

Controlador REST para gestionar divisiones de gastos.

**ExpenseSplitDTO**

DTO para una división de gasto.

**ExpenseSplitRepository****ExpenseSplitService**

Servicio para gestionar las divisiones de gastos.

**ExpenseSplitType**

Enum que define cómo se divide un gasto entre los participantes.

**GlobalExceptionHandler**

Manejador global de excepciones para la API REST.

**HealthController**

Controlador para verificar el estado de los servicios del backend Usado por la página de estado para comprobar que el backend está disponible

**JwtAuthenticationFilter****JwtTokenProvider****LoginRequestDTO**

DTO para solicitud de login.

**LoginResponseDTO**

DTO para respuesta de login (contiene el token JWT y datos del usuario).

**OpenApiConfig**

Configuración de OpenAPI/Swagger para la API REST.

**ResourceNotFoundException**

Excepción base para errores de recursos no encontrados.

**SecurityConfig**

Configuración de Spring Security con JWT y CORS

**Trip**

Entidad que representa un viaje colaborativo.

**TripController**

Controlador REST para gestionar viajes.

**TripCreateRequestDTO**

DTO para crear un nuevo viaje.

**TripDTO**

DTO para información completa de un viaje.

**TripMember**

Entidad que representa la relación entre un Usuario y un Viaje.

**TripMemberController**

Controlador REST para gestionar miembros de viajes.

#### **TripMemberDTO**

DTO para información de un miembro de un viaje.

#### **TripMemberRepository**

#### **TripMemberRole**

Enumeración que define los roles de un usuario dentro de un viaje.

#### **TripMemberService**

Servicio para gestionar la membresía de usuarios en viajes.

#### **TripRepository**

#### **TripService**

#### **UpdateActivityRequestDTO**

DTO para actualizar una actividad existente.

#### **User**

Entidad que representa un usuario en el sistema.

#### **UserController**

Controlador REST para gestionar usuarios.

#### **UserCreateRequestDTO**

DTO para crear/registrar un nuevo usuario.

#### **UserDTO**

DTO para información pública de un usuario.

#### **UserRepository**

#### **UserRole**

Enum que define los roles de los usuarios en el sistema.

#### **UserService**

#### **UserUpdateRequestDTO**

DTO para actualizar datos de un usuario existente.

#### **ValidationException**

Excepción para errores de negocio (ej: viaje con fechas inválidas).

#### **WebConfig**

Configuración Web - CORS Global Este filtro CORS se ejecuta ANTES de todos los filtros de seguridad, asegurando que el preflight (OPTIONS) sea procesado correctamente.