

Proyecto Fase 3 Backend:

Primero me he creado el proyecto con laravel 12, después he seguido con la estructura de datos, para ello primero.

He diseñado y migrado las tablas: Creé las migraciones para users, events, types, registrations y registration_status.

He poblado el sistema (Seeders): He creado Seeders para tener datos reales desde el principio: tipos de eventos (Cultura, Deporte...) y los estados de inscripción (Pendiente, Aprobada...).

Y he personalizado las tablas: Modifiqué la tabla de usuarios para añadir el campo rol.

He definido los modelos y las relaciones para ello he vinculado mis modelos en los archivos app/Models, y queda de la siguiente manera:

- **app/Models/Events.php:** Define que un evento pertenece a un tipo(Type) y tiene muchas inscripciones.
- **app/Models/Registrations.php:** El modelo intermedio que conecta usuarios con eventos y gestiona el estado.
- **app/Models/Type && RegistrationStatus.php** Modelos para las tablas de referencia(Types y Status).
- **app/Models/User** Configurado con HasApiTokens para permitir el login mediante tokens de Sanctum.

He habilitado las Rutas de api ya que no estaba activo, para eso ejecute el comando para crear el archivo routes/api.php y separar mi backend de la parte web.

Me he creado los controladores de la siguiente manera:

- **app/Http/Controllers/Api/AuthController:** Gestiona el inicio de sesión y la generación de tokens de seguridad.
- **app/Http/Controllers/Api/EventController:**
 - **Index():** Obtiene todos los eventos con su tipo asociado y los devuelve en formato JSON
 - **Store():** Permite al administrador crear nuevos eventos y valida la capacidad.
- **app/Http/Controllers/Api/RegistrationController.php:**
 - **store():** Gestiona las inscripciones validando que no haya duplicados y que el evento tenga cupo disponible.
 - **index() & updateStatus():** Permite al admin ver todas las solicitudes y aprobarlas/rechazarlas.
- **app/Http/Controllers/Api/UserController:**
 - **Profile():** Devuelve los datos del usuario logueado y su historial de eventos.

Tras esto, he configurado los endpoints por los cuales haré la conexión cuando tenga el frontend hecho.

En routes/api.php he definido que rutas son públicas (Login y Ver eventos) y cuales están protegidas con el token (Inscribirse, perfil y crear eventos).
Y he comprobado que las peticiones a la api son correctas usando postman