

SPORT RESERVE MOBILE

MANUAL REALIZADO POR
JUAN CAMILO NEUTA SUNCE

MANUAL DE PROGRAMADOR
FUNDACIÓN ESCUELA TECNOLOGICA DE NEIVA “JESÚS OVIEDO PEREZ”
PROYECTO DE GRADO – CICLO “TECNOLOGÍA EN DESARROLLO DE
SISTEMAS DE INFORMACIÓN Y REDES”
RIVERA – HUILA
2025

Tabla de contenido

Descripción general del sistema	3
Requerimientos técnicos.....	3
Estructura del proyecto.....	3
Configuración Inicial	4
1. Crear proyecto Laravel con el comando:.....	4
2. Configurar archivo .env.....	4
3. Instalar Sanctum:.....	4
Implementación de módulos.....	4
- Autenticación:.....	4
- Canchas.....	4
Middleware personalizado.....	5
Base de datos	5
Entidades principales:.....	5
Pruebas en Postman	5
Flujo de pruebas:	5
Errores comunes y soluciones	5
Próximas fases	5

Descripción general del sistema

El backend de SportReserve Mobile fue desarrollado en Laravel 12 con PostgreSQL y Laravel Sanctum. Gestiona la autenticación de usuarios, manejo e canchas deportivas y sistemas de reservas, sirviendo como capa API REST para la aplicación móvil desarrollada en Flutter.

Requerimientos técnicos

- Sistema operativo: Windows 10 o superior
- PHP 8.2+
- Composer 2.6+
- PostgreSQL 14 o superior
- Visual Studio Code
- Postman
- Laravel Sanctum

Estructura del proyecto

```
sportreserve_mobile_backend/
├── app/
│   ├── Http/
│   │   ├── Controllers/
│   │   ├── Middleware/
│   │   └── Models/
│   └── routes/
│       └── api.php
└── database/
    └── migrations/
├── .env
└── composer.json
```

Configuración Inicial

1. Crear proyecto Laravel con el comando:

```
composer create-project laravel/laravel sportreserve_mobile_backend
```

2. Configurar archivo .env

```
DB_CONNECTION=pgsql  
DB_HOST=127.0.0.1  
DB_PORT=5432  
DB_DATABASE=sportreserve_mobile_db  
DB_USERNAME=postgres
```

3. Instalar Sanctum:

```
composer require laravel/sanctum  
php artisan migrate
```

Implementación de módulos

- Autenticación:

```
POST /api/register - Registro de usuario  
POST /api/login - Inicio de sesión  
POST /api/logout - Cierre de sesión  
GET /api/profile - Perfil del usuario autenticado
```

- Canchas

```
GET /api/canchas - Listar canchas  
POST /api/canchas - Crear cancha  
PUT /api/canchas/{id} - Editar cancha  
DELETE /api/canchas/{id} - Eliminar cancha
```

Middleware personalizado

Se creó ‘Ensureapiauthenticated.php’ para manejar respuestas JSON en lugar de redirecciones web. Registrado en ‘bootstrap/app.php’ como alias ‘auth.api’

Base de datos

Entidades principales:

- Users: usuarios registrados
- Canchas: canchas deportivas
- Reservas (pendiente) registro de reservas por usuario

Pruebas en Postman

Flujo de pruebas:

1. Registrar usuario
2. Iniciar sesión
3. Copiar token y probar */profile* y */canchas*
4. Cerrar sesión
5. Intentar acceder sin token → respuesta: ‘No autenticado’

Errores comunes y soluciones

- Route [login] not defined → se corrigió con middleware ‘auth.api’
- Error 500 en profile → faltaba ‘HasApiTokens’ en modelo User
- Error MassAssignment → faltaba \$fillable en modelo

Próximas fases

- Implementar módulo de reservas
- Agregar subida de imágenes
- Integrar con Flutter
- Documentar API en Swagger o Postman