

# Documentación Técnica — Sistema de Gestión de Laboratorios Académicos (TEC)

---

**FUTEC**

Fecha: 29/09/25

---

## 1) Arquitectura general

- **Monorepo** con dos aplicaciones:
  - **API (Node + Express + PostgreSQL):** `apps/api`
  - **Móvil (Expo/React Native):** `apps/mobile`
- **Base de datos:** PostgreSQL 16 (Docker). Conexión centralizada desde `src/db/pool.js`.
- **Autenticación:** JWT con `JWT_SECRET` (utilidad en `src/utils/config.js` y controlador `auth.controller.js`).
- **CORS y JSON:** configurados en `src/app.js`.
- **Inicialización DB:** `src/server.js` ejecuta `src/db/schema.sql` de forma idempotente al iniciar.

## 2) Despliegue local (Docker Compose)

1. Copie y ajuste `.env` en la raíz (Postgres, puertos y pgAdmin).
2. Levante servicios:

```
docker compose up -d
```

3. (Opcional) **pgAdmin** en `http://localhost:5050` con credenciales del `.env`.

4. **App móvil (Expo):**

```
cd apps/mobile
npm install|
# Configure el backend:
#   EXPO_PUBLIC_API_URL=http://localhost:8080
npx expo start
```

En emuladores, la app usa fallback automático:

- Android: <http://10.0.2.2:8080/api>
- iOS: <http://localhost:8080/api>

## Variables de entorno (raíz)

- `POSTGRES_USER = labtec`
- `POSTGRES_PASSWORD = labtec`
- `POSTGRES_DB = labtec`
- `API_PORT = 8080`
- `NODE_ENV = development`
- `CORS_ORIGIN = http://localhost:19006,http://localhost:19000,exp:///*`
- `PGADMIN_DEFAULT_EMAIL = admin@tec.ac.cr`
- `PGADMIN_DEFAULT_PASSWORD = admin123`
- `PGADMIN_PORT = 5050`

## Variables de entorno (API)

- `DATABASE_URL = postgresql://labtec:labtec@localhost:5432/labtec`
  - `PORT = 8080`
  - `JWT_SECRET = labtecsecret`
  - `CORS_ORIGIN = http://localhost:19006,http://localhost:19000,exp:///*`
  - `NODE_ENV = development`
- 

## 3) API (Express)

- **Punto base:** <http://<host>:8080/api>
- **Middlewares:** CORS, JSON, `errorHandler/notFound`.
- **Autenticación:** `/api/auth/register` y `/api/auth/login`. Respuesta incluye `token` JWT (12h) y `user` seguro.
- **Protecciones típicas:** validaciones de dominios institucionales (según reglas implementadas en modelos/controladores).

## Endpoints

Módulo (archivo)	Método	Ruta (relativa a <code>/api</code> )
Reportes institucionales	<b>GET</b>	<a href="#">/consumption</a>
Reportes institucionales	<b>GET</b>	<a href="#">/inventory-institutional</a>
Reportes institucionales	<b>GET</b>	<a href="#">/performance</a>
Reportes institucionales	<b>GET</b>	<a href="#">/usage-global</a>
Autenticación	<b>POST</b>	<a href="#">/login</a>
Autenticación	<b>POST</b>	<a href="#">/register</a>
Disponibilidad (slots, suscripciones, bitácora)	<b>GET</b>	<a href="#">/changelog</a>

Módulo (archivo)	Método	Ruta (relativa a /api)
Disponibilidad (slots, suscripciones, bitácora)	<b>GET</b>	/labs/:labId/slots
Disponibilidad (slots, suscripciones, bitácora)	<b>POST</b>	/labs/:labId/slots
Disponibilidad (slots, suscripciones, bitácora)	<b>DELETE</b>	/slots/:id
Disponibilidad (slots, suscripciones, bitácora)	<b>PUT</b>	/slots/:id/status
Disponibilidad (slots, suscripciones, bitácora)	<b>GET</b>	/subscriptions
Disponibilidad (slots, suscripciones, bitácora)	<b>POST</b>	/subscriptions
Exploración (labs, recursos, calendario)	<b>GET</b>	/calendar
Exploración (labs, recursos, calendario)	<b>GET</b>	/labs
Exploración (labs, recursos, calendario)	<b>GET</b>	/resources
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>PUT</b>	/assignments/:id/return
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>GET</b>	/reports/usage
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>GET</b>	/requests/:requestId/assignments
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>POST</b>	/requests/:requestId/assignments
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>GET</b>	/requests/:requestId/benefits
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>POST</b>	/requests/:requestId/benefits
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>GET</b>	/requests/:requestId/consumptions
Control (asignaciones, consumos, beneficios, reportes de uso)	<b>POST</b>	/requests/:requestId/consumptions
Escuelas/Departamentos	<b>GET</b>	/
Escuelas/Departamentos	<b>POST</b>	/
Inventario (consumibles, equipos fijos, movimientos)	<b>GET</b>	/consumables
Inventario (consumibles, equipos fijos, movimientos)	<b>POST</b>	/consumables/:id/movements
Inventario (consumibles, equipos fijos, movimientos)	<b>GET</b>	/movements

Módulo (archivo)	Método	Ruta (relativa a /api)
Inventario (consumibles, equipos fijos, movimientos)	<b>PUT</b>	/resources-fixed/:id/status
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/
Laboratorios (perfil, contactos, horarios, recursos)	<b>POST</b>	/
Laboratorios (perfil, contactos, horarios, recursos)	<b>DELETE</b>	/:id
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/:id
Laboratorios (perfil, contactos, horarios, recursos)	<b>PUT</b>	/:id
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/:id/consumables
Laboratorios (perfil, contactos, horarios, recursos)	<b>POST</b>	/:id/consumables
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/:id/contacts
Laboratorios (perfil, contactos, horarios, recursos)	<b>POST</b>	/:id/contacts
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/:id/history
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/:id/hours
Laboratorios (perfil, contactos, horarios, recursos)	<b>PUT</b>	/:id/hours
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/:id/policies
Laboratorios (perfil, contactos, horarios, recursos)	<b>PUT</b>	/:id/policies
Laboratorios (perfil, contactos, horarios, recursos)	<b>GET</b>	/:id/resources-fixed
Laboratorios (perfil, contactos, horarios, recursos)	<b>POST</b>	/:id/resources-fixed
Mantenimientos (ordenes)	<b>GET</b>	/orders
Mantenimientos (ordenes)	<b>POST</b>	/orders
Mantenimientos (ordenes)	<b>GET</b>	/orders/:id
Mantenimientos (ordenes)	<b>PUT</b>	/orders/:id/cancel
Mantenimientos (ordenes)	<b>PUT</b>	/orders/:id/complete
Mantenimientos (ordenes)	<b>PUT</b>	/orders/:id/start
Reportes operativos	<b>GET</b>	/inventory
Reportes operativos	<b>GET</b>	/maintenance
Reportes operativos	<b>GET</b>	/usage
Solicitudes de uso y reservas	<b>GET</b>	/
Solicitudes de uso y reservas	<b>POST</b>	/
Solicitudes de uso y reservas	<b>GET</b>	/:id

Módulo (archivo)	Método	Ruta (relativa a /api)
Solicitudes de uso y reservas	<b>POST</b>	/:id/approve
Solicitudes de uso y reservas	<b>PUT</b>	/:id/cancel
Solicitudes de uso y reservas	<b>GET</b>	/:id/messages
Solicitudes de uso y reservas	<b>POST</b>	/:id/messages
Solicitudes de uso y reservas	<b>POST</b>	/:id/need-info
Solicitudes de uso y reservas	<b>POST</b>	/:id/reject
Solicitudes de uso y reservas	<b>PUT</b>	/:id/status
Solicitudes de uso y reservas	<b>POST</b>	/preview
Tipos de recurso	<b>GET</b>	/
Tipos de recurso	<b>POST</b>	/
Recursos (creación, edición, fotos)	<b>GET</b>	/
Recursos (creación, edición, fotos)	<b>POST</b>	/
Recursos (creación, edición, fotos)	<b>DELETE</b>	/:id
Recursos (creación, edición, fotos)	<b>GET</b>	/:id
Recursos (creación, edición, fotos)	<b>PUT</b>	/:id
Recursos (creación, edición, fotos)	<b>GET</b>	/:id/photos
Recursos (creación, edición, fotos)	<b>POST</b>	/:id/photos
Recursos (creación, edición, fotos)	<b>PUT</b>	/:id/request-url
Técnicos (gestión operativa)	<b>GET</b>	/assignments
Técnicos (gestión operativa)	<b>POST</b>	/assignments
Técnicos (gestión operativa)	<b>PUT</b>	/assignments/:id/damaged
Técnicos (gestión operativa)	<b>PUT</b>	/assignments/:id/lost
Técnicos (gestión operativa)	<b>PUT</b>	/assignments/:id/return
Técnicos (gestión operativa)	<b>GET</b>	/requests/:id/precheck
Técnicos (gestión operativa)	<b>GET</b>	/requests/approved
Usuarios (perfil e historial)	<b>GET</b>	/me
Usuarios (perfil e historial)	<b>PUT</b>	/me
Usuarios (perfil e historial)	<b>GET</b>	/me/history
Usuarios (perfil e historial)	<b>GET</b>	/me/lab-requirements
Usuarios (perfil e historial)	<b>GET</b>	/me/trainings

Módulo (archivo)	Método	Ruta (relativa a /api)
Usuarios (perfil e historial)	POST	/me/trainings

Algunos routers exponen rutas en raíz (/) y otros bajo prefijos (/labs, /reports, etc.).

### Ejemplos (cURL)

```
# Registro
curl -X POST http://localhost:8080/api/auth/register -H "Content-Type: application/json" -d '{"email":"usuario@estudiante.tec.ac.cr","password":"Xx_2025","full_name":"Nombre Apellido"}'

# Login
curl -X POST http://localhost:8080/api/auth/login -H "Content-Type: application/json" -d '{"email":"usuario@estudiante.tec.ac.cr","password":"Xx_2025"}'

# Listar laboratorios
curl http://localhost:8080/api/labs

# Crear slot de disponibilidad
curl -X POST http://localhost:8080/api/labs/1/slots -H "Content-Type: application/json" -d '{"starts_at":"2025-10-01T08:00:00","ends_at":"2025-10-01T10:00:00","status":"DISPONIBLE","title":"Práctica 1"}'
```

## 4) Base de datos (PostgreSQL)

- **Pool:** `src/db/pool.js` (usa `DATABASE_URL` y `ssl:false` por defecto).
- **Bootstrap:** `src/server.js` lee y ejecuta `src/db/schema.sql` (migración idempotente).

### Tablas por módulo (principales)

- **1.1 Gestión de Perfiles de Laboratorio:** `departments`, `labs`, `lab_contacts`, `lab_hours`, `resources_fixed`, `consumables`, `lab_history`
- **1.2 Publicación de Disponibilidad y Recursos:** `resources`, `calendar_slots`, `availability_subscriptions`, `changelog`
- **1.3 Gestión de Solicitudes y Reservas:** `requests`, `request_items`, `request_messages`
- **2.2 Gestión de Inventario:** `inventory_movements`
- **2.3 Gestión de Mantenimientos:** `maintenance_orders`
- **3.1/3.4 Usuarios e Histórial:** `users`, `trainings_catalog`, `user_trainings`
- **3.5 Notificaciones y Mensajería:** `notifications`, `changelog`, `request_messages`
- **4.x Administración y Auditoría:** `system_settings`, `role_permissions`

En `src/db/schema.sql` se encuentran las llaves foráneas, restricciones (`CHECK`), índices y bitácoras (`changelog`, `audit_log`).

## Semillas

- `src/db/seed_users.sql`: crea usuarios demo (ADMIN, DOCENTE, ESTUDIANTE, TECNICO) con `pgcrypto/crypt()`.
- 

## 5) Seguridad y permisos

- **JWT** con `JWT_SECRET` (12h). No comparta el secreto.
  - **Roles**: ADMIN, TECNICO, DOCENTE, ESTUDIANTE. Permisos sugeridos en `role_permissions`.
  - **CORS**: permitido desde orígenes configurados (`CORS_ORIGIN`) o \* por defecto.
  - **Validaciones**: dominios institucionales y requisitos de laboratorio (capacitaciones).
- 

## 6) Manejo de errores y bitácoras

- **Errores HTTP**: `middlewares/error.js` gestiona `notFound` y `errorHandler`.
  - **Errores PG**: varios controladores traducen códigos (ej. 23503 violación de FK) a mensajes legibles.
  - **Changelog/Auditoría**: tablas `changelog` y `audit_log` con filtros y exportación.
- 

## 7) Móvil (Expo/React Native)

- **Base de API**: `services/api.ts` usa `EXPO_PUBLIC_API_URL` o fallbacks por plataforma.
  - **Rutas de UI**: carpeta `app/` organizada por features (auth, browse, requests, tech, admin, reports).
  - **Servicios HTTP**: carpeta `services/` separa clientes por dominio (labs, requests, inventory, etc.).
  - **Tipos y hooks**: carpetas `types/` y `hooks/` para estado y utilidades.
- 

## 8) Pruebas manuales (sugeridas)

1. Crear **departamento y laboratorio** → agregar **contactos y horarios**.
  2. Publicar **recursos** y crear **slots** de disponibilidad.
  3. Registrar **usuario docente/estudiante y enviar solicitud**.
  4. Desde **técnico**, entregar recursos (assignments) y registrar consumos.
  5. Crear **orden de mantenimiento** y completarla.
  6. Generar **reportes** y exportar a PDF/Excel.
- 

## 9) Despliegue/Producción (pautas)

- Definir `DATABASE_URL` con SSL y `JWT_SECRET` robustos.
  - Endurecer CORS con orígenes específicos.
  - Configurar backups de DB y rotación de logs.
  - Habilitar métricas/monitorización externa si aplica.
- 

## Librerías utilizadas

### API (Node.js):

- bcryptjs
- cors
- dotenv
- express
- helmet
- json2csv
- jsonwebtoken
- morgan
- pg

**API (Dev):**

- nodemon

**Móvil (Expo/React Native):**

- @expo/vector-icons
- @react-navigation/bottom-tabs
- @react-navigation/elements
- @react-navigation/native
- axios
- expo
- expo-blur
- expo-constants
- expo-font
- expo-haptics
- expo-image
- expo-linking
- expo-router
- expo-splash-screen
- expo-status-bar
- expo-symbols
- expo-system-ui
- expo-web-browser
- react
- react-dom
- react-native
- react-native-gesture-handler
- react-native-reanimated
- react-native-safe-area-context
- react-native-screens
- react-native-web
- react-native-webview

**Móvil (Dev):**

- @babel/core
- @types/react

- babel-plugin-module-resolver
- eslint
- eslint-config-expo
- typescript

## Tablas de la base de datos

- academic\_benefits
- availability\_subscriptions
- calendar\_slots
- changelog
- consumables
- departments
- inventory\_movements
- lab\_contacts
- lab\_history
- lab\_hours
- lab\_policies
- lab\_training\_requirements
- labs
- maintenance\_orders
- material\_consumptions
- notifications
- request\_items
- request\_messages
- requests
- resource\_assignments
- resource\_photos
- resource\_types
- resources
- resources\_fixed
- role\_permissions
- system\_settings
- trainings\_catalog
- user\_trainings
- users