

Documentación General del Proyecto

Unilab+

Sistema de gestión de préstamos y reservas de materiales universitarios

Integrante:

Alejandro Ortiz Vargas

Fredman Santiago Plazas Artunduaga

Docente: Jesus Ariel

Curso: Programacion Movable

Fecha: 19 de mayo de 2025

Universidad Corporación Colporativa Corhuila

Facultad de ingeniería

Carrera de ingeniería en sistemas

Documentación general del Proyecto Unilab+

1. Presentación Técnica

Unilab+ es una plataforma web desarrollada para facilitar la gestión de préstamos y reservas de materiales en un entorno universitario.

El sistema permite que los usuarios se registren, reserven materiales y gestionen sus reservas, mientras que los administradores pueden controlar usuarios, materiales y estados del sistema.

El proyecto está dividido en dos componentes principales:

- Backend: Desarrollado en Java con Spring Boot, implementa una arquitectura en capas (Controlador, Models, DTO, Servicio, Repositorio, “Seguridad”).
- Frontend: Desarrollado con Angular, ofreciendo una interfaz moderna y responsive.
- Aplicación Móvil (APK), Funcionamiento con Dev Tunnels:

Inicialmente, al generar el APK de la aplicación móvil con Ionic Angular, se presentó una problemática relacionada con la conectividad hacia el backend. Aunque el APK podía instalarse y ejecutarse correctamente en dispositivos Android, no permitía funcionalidades clave como el registro, inicio de sesión o reservas, debido a la falta de conexión con los servicios REST del backend.

- Solución Implementada


Para resolver esta limitación, se implementó un mecanismo de túnel público usando la herramienta Dev Tunnels de Microsoft. Esta herramienta permite exponer de manera segura el servidor backend local (Spring Boot) a través de una URL pública accesible desde el dispositivo móvil donde se instala el APK.

- Configuración

1. Puerto backend Spring Boot: 8080
2. URL pública generada: <https://w6x4dvhq-8080.use.devtnnls.ms/>
3. Esta URL se ha configurado como base para las peticiones HTTP en todos los servicios del frontend Angular.

- Cambios en el código frontend

Anjunto imagen:



```
environment.ts M
src > environments > environment.ts > ...
1  export const environment = {
2    production: true,
3    apiUrl: 'https://w6x4dvhq-8080.use.devtnnls.ms/api/auth',
4  };
5
```

Los servicios (user.service.ts, material.service.ts, reserva.service.ts, etc.) consumen esta apiUrl para asegurar la conectividad desde el APK hacia el backend.

- Resultado

Gracias a esta configuración, el APK ahora puede:

1. Registrar usuarios.
2. Autenticar mediante login.
3. Mostrar y reservar materiales.
4. Consultar historial de reservas.

Ejecutar todas las funcionalidades previstas, desde un dispositivo Android real conectado a Internet.

Este enfoque asegura una experiencia funcional completa sin necesidad de desplegar el backend en un servidor externo, ideal para entornos de pruebas, desarrollo o presentaciones.

2. Requerimientos

Funcionales:

- Registro y autenticación de usuarios.
- CRUD de materiales.

- Gestión de reservas.
- Roles diferenciados (usuario y administrador).
- Notificación de estados de reserva.

No Funcionales:

- Seguridad en el acceso mediante JWT.
- Estructura modular y mantenible.
- Compatible con dispositivos móviles.
- Base de datos relacional.

3. Arquitectura General

Backend (Spring Boot):

- Controladores REST
- Servicios con lógica de negocio
- Repositorios con Spring Data JPA
- Seguridad con Spring Security

Frontend (Angular):

- Módulos por páginas (login, register, home, admin, reservas, materiales,)
- Ruteo y consumo de APIs
- Componentes reutilizables
- Comunicación: HTTP/HTTPS con autenticación vía “JWT”.

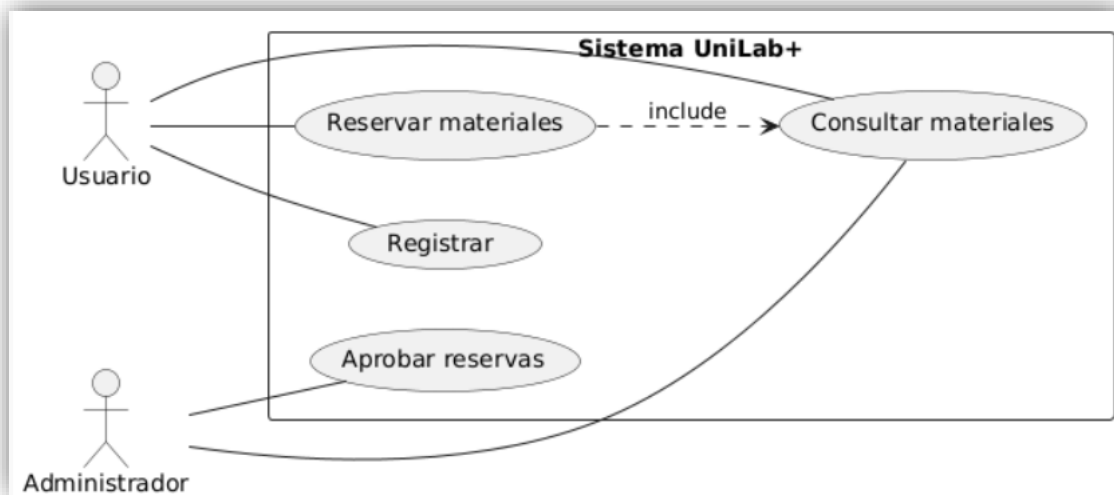
4. Diagramas UML, MER y MR

UML:

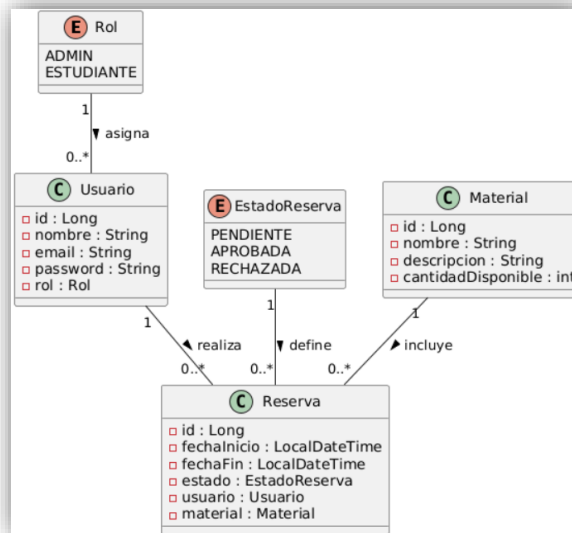
- Diagrama de Casos de Uso:

Actores principales: Usuario y Administrador. Casos: Registrar, Reservar, Aprobar, Consultar materiales.

Anjunto imagen:



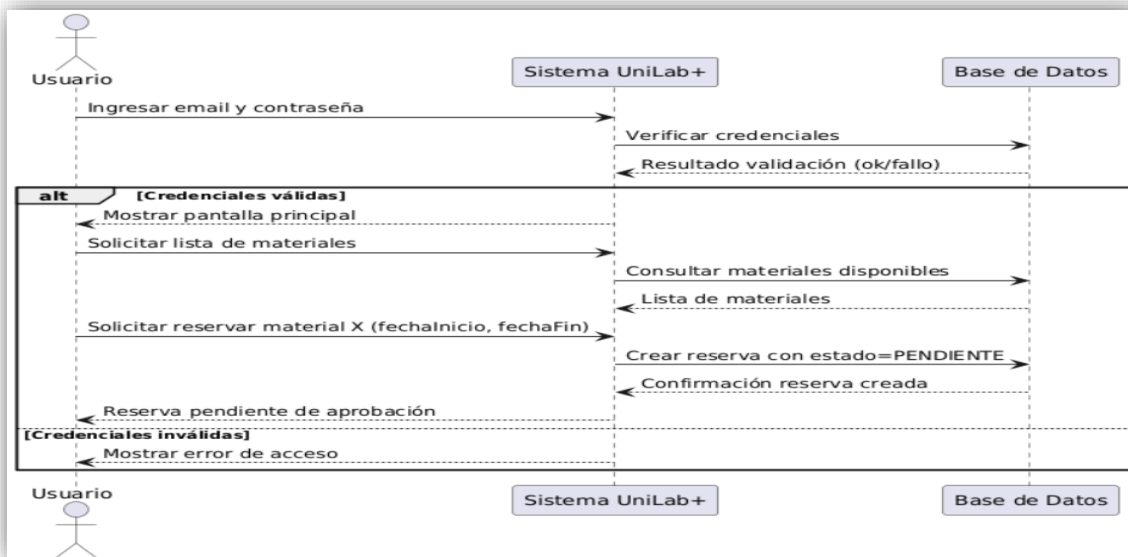
- Diagrama de Clases: Usuario, Material, Reserva, Rol, EstadoReserva.



Anjunto imagen:

- Diagrama de Secuencia: Flujo típico de autenticación y reserva.

Anjunto imagen:



MER:

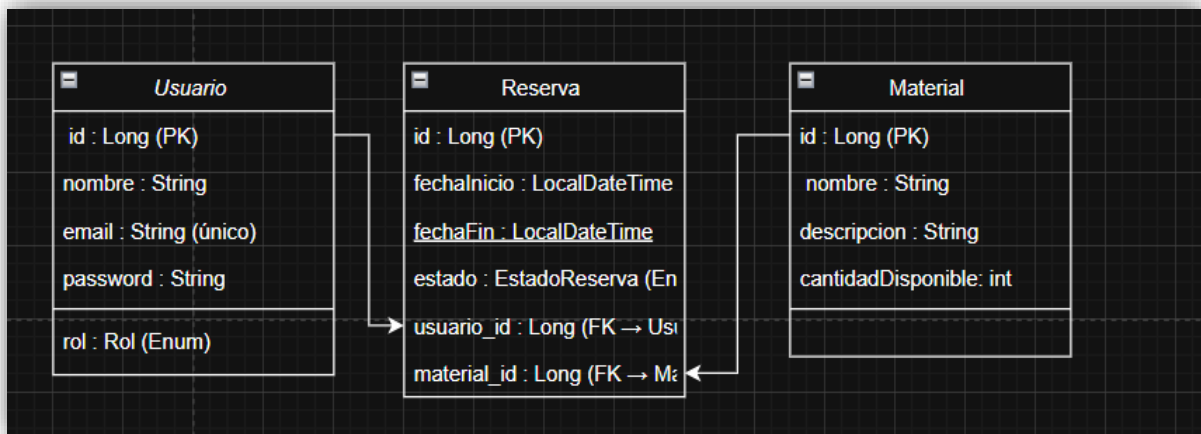
- Usuario (id, nombre, email, contraseña, rol_id)
- Material (id, nombre, descripción, cantidad)

- Reserva (id, usuario_id, material_id, fecha_inicio, fecha_fin, estado_id)
- EstadoReserva (id, nombre)
- Rol (id, nombre)

MR:

- Relaciones 1:N entre Usuario → Reserva, Material → Reserva, Rol → Usuario, EstadoReserva → Reserva

Ajunto imagen :



5. Backlog General del Proyecto Unilab+

Este backlog define las funcionalidades esenciales del sistema Unilab+, priorizadas según su impacto en la experiencia del usuario y la operación del sistema. Cada ítem corresponde a una historia de usuario documentada, incluye el objetivo funcional, criterios de aceptación clave, prioridad y estimación de esfuerzo (story points).

1. HU-01: Registro de Usuario

Objetivo: Permitir el registro de estudiantes y docentes en Unilab+.

Backend:

- Endpoint para crear usuario con validaciones (correo único, contraseña segura).
- Envío opcional de correo de verificación.
- Guardado en base de datos y gestión de errores.

Criterios de aceptación:

- Validar datos obligatorios.
- Rechazar correos duplicados.
- Confirmación visual y/o por correo.

Prioridad: Alta

Estimación: 4 puntos

Rutas asociadas: POST /api/usuarios/register

2. HU-02: Inicio de Sesión

Objetivo: Autenticar usuarios para el acceso seguro al sistema.

Backend:

- Seguridad en contraseñas (hashing).
- Endpoint de Autenticación

Criterios de aceptación:

- Validar existencia del correo antes de contraseña.
- Mostrar feedback claro.
- Bloquear después de 3 intentos fallidos.

Prioridad: Alta.

Estimación: 3 puntos.

Rutas asociadas: POST /api/auth/login

3. HU-03: Visualización de Materiales

Objetivo: Mostrar los materiales disponibles para consulta.

Backend:

- Endpoint para listar materiales con paginación y filtros.
- Incluir detalles de disponibilidad y fechas de devolución.

Criterios de aceptación:

- Mostrar nombre, estado, descripción.
- Resaltar disponibilidad inmediata.
- Optimización de carga.

Prioridad: Media

Estimación: 5 puntos

Rutas asociadas: GET /api/materiales, GET /api/materiales/{id}

4. HU-04: Reserva de Material

Objetivo: Permitir a los usuarios reservar materiales.

Backend:

- Endpoint para crear reservas con validación de disponibilidad.
- Gestión de fechas y conflictos.
- Cancelación de reservas.

Criterios de aceptación:

- Validar disponibilidad por fecha y duración.
- Confirmación de éxito.

Prioridad: Media**Estimación:** 6 puntos**Rutas asociadas:** POST /api/reservas, GET /api/reservas, DELETE /api/reservas/{id}**5. HU-05: Gestión de Materiales (Administrador)**

Objetivo: Permitir a administradores crear, modificar o eliminar materiales.

Backend:

- CRUD completo de materiales con autenticación de rol.
- Registro de cambios (bitácora opcional).

Criterios de aceptación:

- Crear, editar, eliminar materiales.
- Seguridad basada en roles (solo admin).

Prioridad: Alta**Estimación:** 5 puntos**Rutas asociadas:** POST /api/admin/materiales, PUT /api/admin/materiales/{id}, DELETE /api/admin/materiales/{id}**6. Consideraciones Técnicas Generales**

- Base de datos: Relacional (MySQL), con entidades Usuario, Material, Reserva.
- Validaciones: En backend y frontend (duplicados, campos vacíos, etc.).
- Optimización: Paginación, uso de DTOs, manejo eficiente de errores.

Tablero: Se recomienda usar Trello o GitHub Projects con las siguientes columnas:

- HU (Historias de Usuario)
- Backlog/Ideas y Mejoras Futuras
- Por Hacer
- En Desarrollo
- En Revisión
- Aprobado
- Completado

6. Manual de Usuario

Inicio de Sesión: Accede al sistema con correo y contraseña.

Reservar Material:

1. Ingresa a la sección de materiales.
2. Selecciona el material y elige fechas.
3. Confirma la reserva.

Administrar (admin):

1. Ve a la sección "Reservas".
2. Acepta o rechaza solicitudes.
3. Consulta historial de reservas y usuarios.

Responsivo: El sistema se adapta a dispositivos móviles.