

Universidad Internacional San Isidro Labrador

Escuela de Ingeniería en Sistemas

Proyecto Final: Sistema de Gestión de Reservas para una Biblioteca Digital

(Isb-27) Ingeniería y Utilización de Software

Profesor: Jason Muñoz Rojas

Estudiantes:

Michel Vargas Lira

Jeudy Eduardo Zuñiga Gonzalez

2025

### Descripción del sistema:

El sistema a desarrollar es una aplicación web para la búsqueda, reserva y gestión de libros, la cual permite al usuario tener una cuenta propia donde puede ver sus libros reservados y administrarlos. Además la aplicación debe tener 2 tipos de perfil de usuario, uno estándar el cual solo permite funciones básicas relacionadas con su biblioteca, y otro perfil de administrador el cual cuenta con opciones de crear nuevos usuarios estándar, agregar y eliminar libros de la aplicación y otras opciones. La arquitectura del sistema sigue el modelo cliente-servidor, con un frontend desarrollado en HTML, CSS y JavaScript, y un backend desarrollado en Java.

### Tecnologías y herramientas utilizadas:

* Java con Spring: Como lenguaje de programación para desarrollar el backend nos decantamos por java ya que este es un lenguaje robusto con numerosas opciones de librerías y herramientas que facilitan el adecuado desarrollo, comunicación con base de datos, pruebas de integridad y seguridad entre muchos otras. Además, es un lenguaje multiplataforma y con gran escalabilidad.
* HTML/CSS/JavaScript: Estas tecnologías son ampliamente utilizadas para el desarrollo de interfaces graficas agradables, intuitivas y funcionales.
* PostgreSQL: En el tema de base de datos optamos por utilizar Postgre ya que es una opción confiable y que se adapta perfectamente al desarrollo junto a Java y Spring, además nos brinda buen rendimiento, escalabilidad y seguridad.
* Postman: Herramienta que utilizamos para hacer pruebas desde el frontend, haciendo solicitudes HTTP al backend, sin necesidad de tener una interfaz gráfica en el navegador.
* IntelliJ: Ide utilizada para manejar Java debido a su gran cantidad de herramientas adaptadas al lenguaje, y su comodidad para trabajar en este.
* Visual Studio Code: Ide utilizada para manejar las tecnologías de frontend, como Html, CSS, Javascript.

### Paradigmas de software:

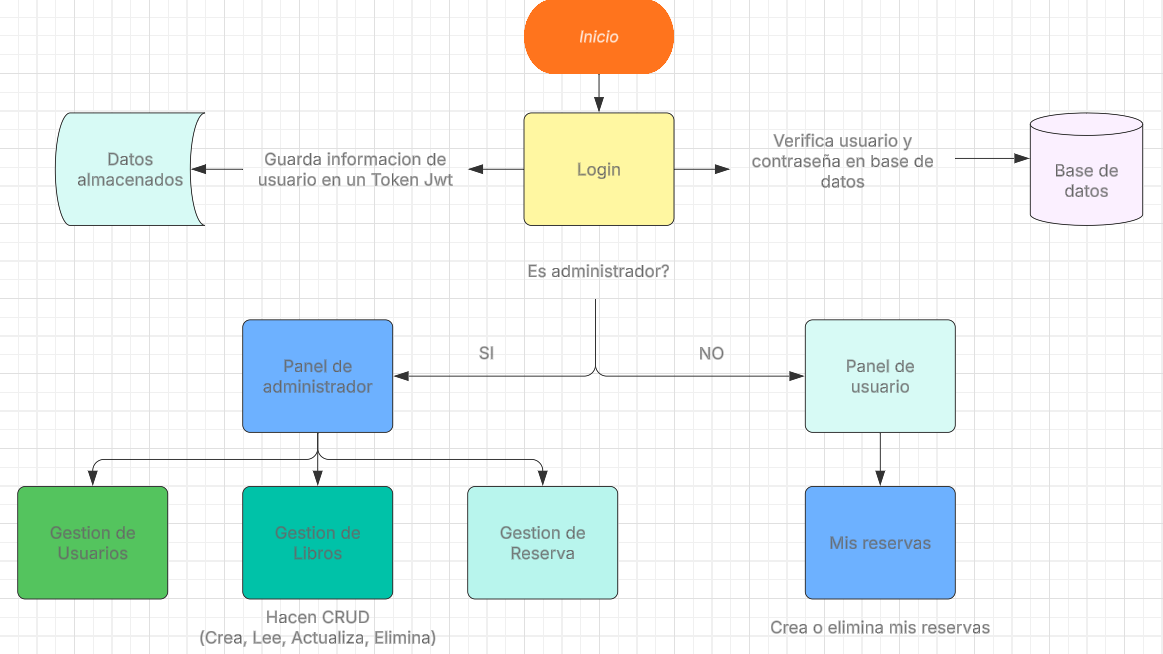
Modelo en espiral: Este paradigma nos permite ser flexibles en el ciclo de desarrollo, permitiéndonos iterar en las diferentes etapas permitiendo modificar y mejorar la aplicación en la marcha tomando en cuenta las revisiones de nosotros como desarrolladores y otros colegas a los cuales podemos mostrar el proyecto en búsqueda de retroalimentación.

Modelo de prototipos: En menor medida también pensamos en la utilización de prototipos principalmente en la interfaz grafica buscando crear un vistazo de la aplicación web en búsqueda de opiniones que nos permitan crear una interfaz funcional e intuitiva.

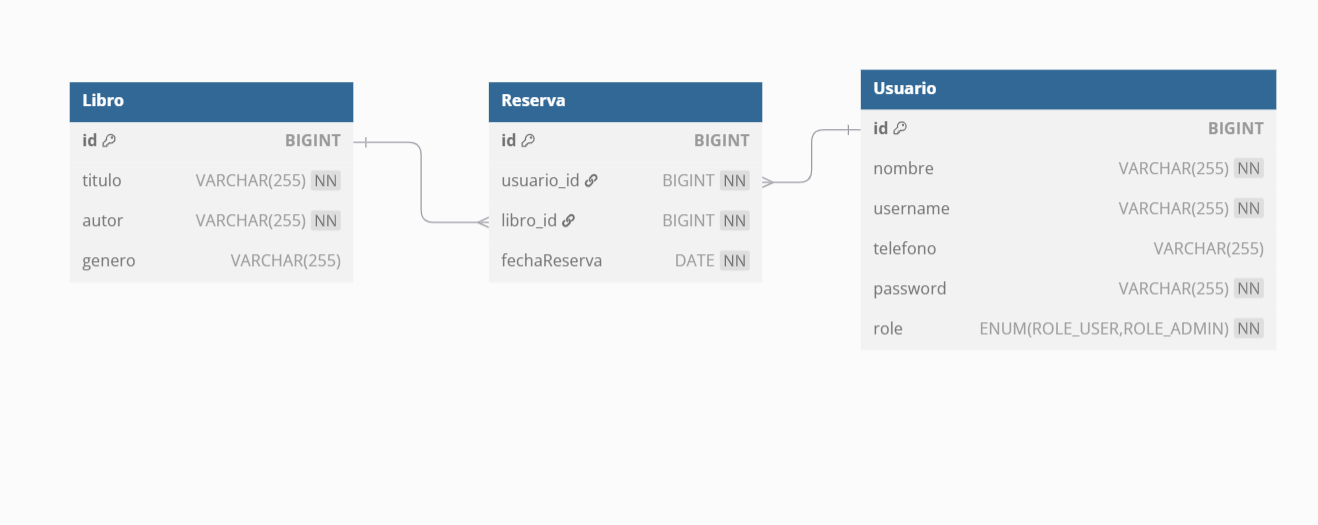
### Funcionalidades principales:

* Búsqueda de libros de diferentes maneras, por nombre, autor o género.
* Reserva de libros por parte de los clientes.
* Gestión de usuarios que permita la creación y administración de diferentes clientes.
* Panel de control de administrador el cual permite agregar o eliminar libros de la plataforma.

### Diagrama de flujo del programa:

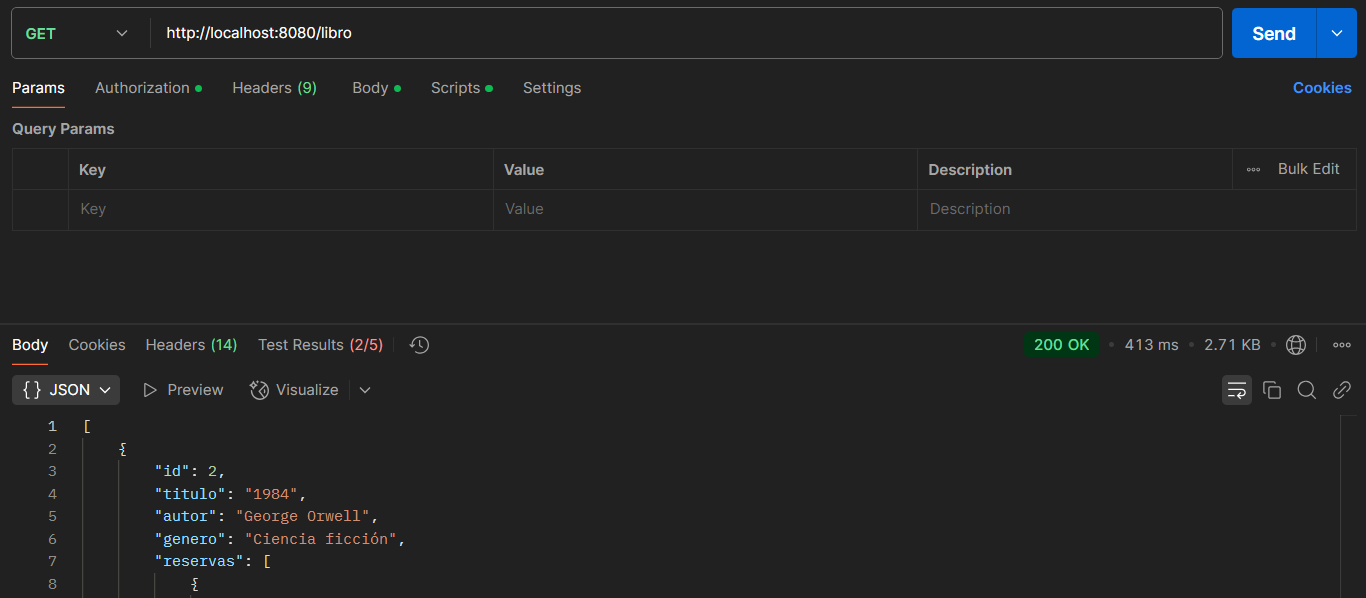


### Diagrama de la base de datos:

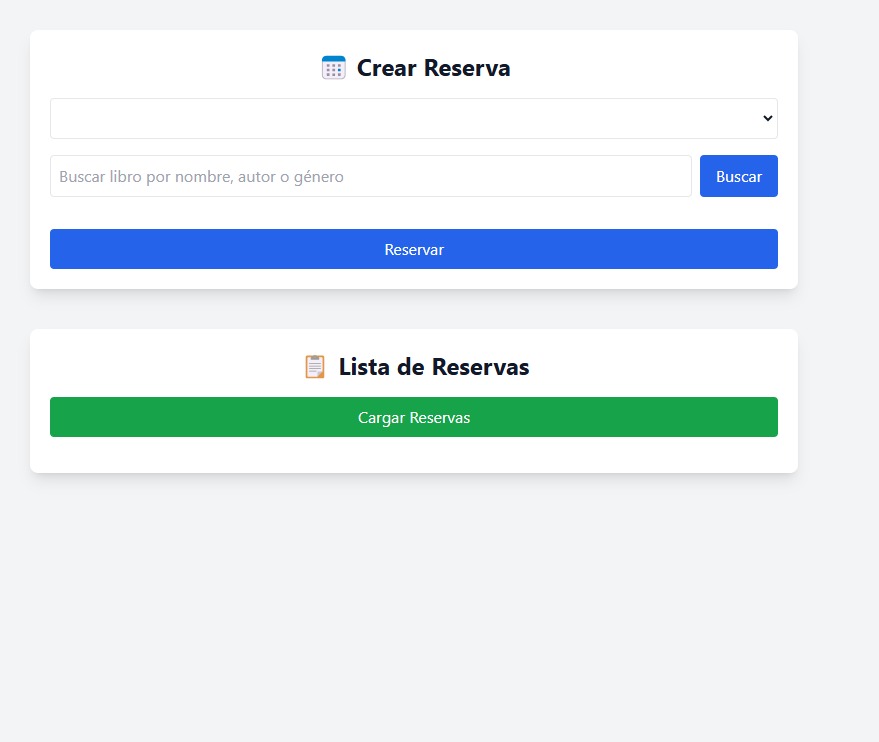
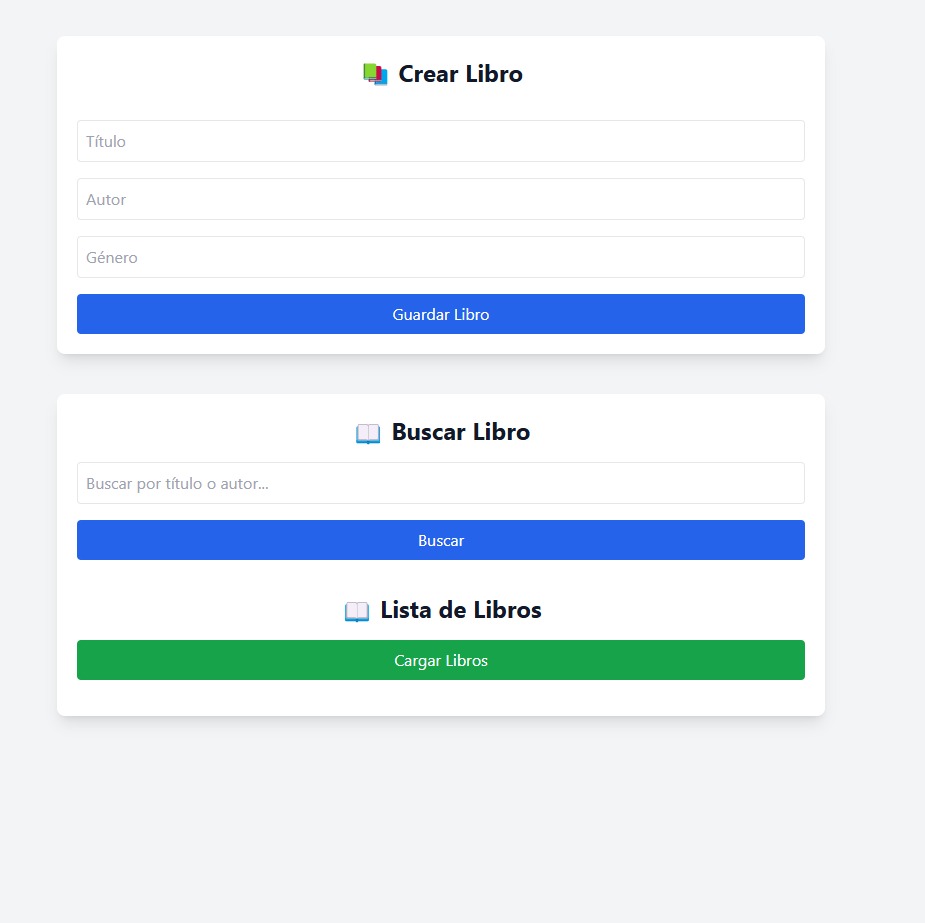
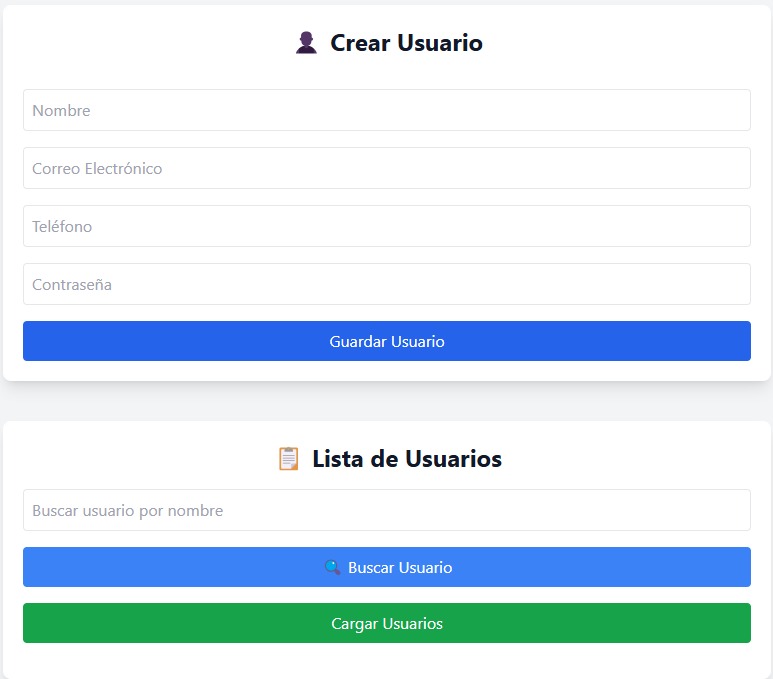
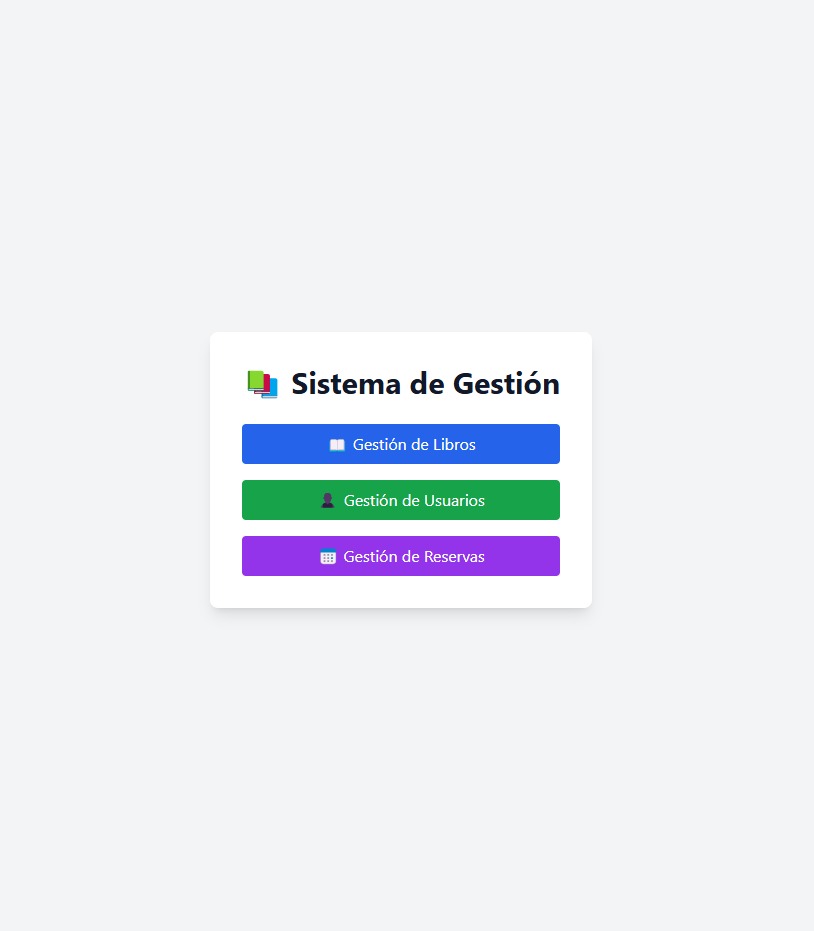


# Versiones:

**Versiones V1:** Es la primera versión del programa en la cual se crea la base de datos, se realiza el API que va conectar nuestro Frontend con el Backend, en esta versión no se cuenta con una interfaz gráfica, solo se realizan cruds básicos a la base de datos, sin implementar seguridad y división de roles. Para realizar pruebas de funcionamiento se utiliza la aplicación de escritorio Postman.



**Versiones v2:** Se busca crear una interfaz gráfica básica la cual nos permita mostrar el programa visualmente, conseguimos crearlo utilizando Html, Javascript y CSS. En esta etapa del proyecto solo contamos con la interfaz que va funcionar como panel de administrador, no se implementa seguridad ni división de Roles.



**Versiones V3:** Se crea el entorno necesario para utilizar Jason web token, el cual nos permite brindarle seguridad a la aplicación con un sistema de autentificación versátil y seguro, el cual consiste en que con cada inicio de sesión se crea un token el cual incluye información de seguridad del usuario y su rol.  
Además se hacen cambios en la apariencia de la aplicación, manteniéndola simple para cualquier usuario que la utiliza, pero agregándole colores menos agresivos para la vista.

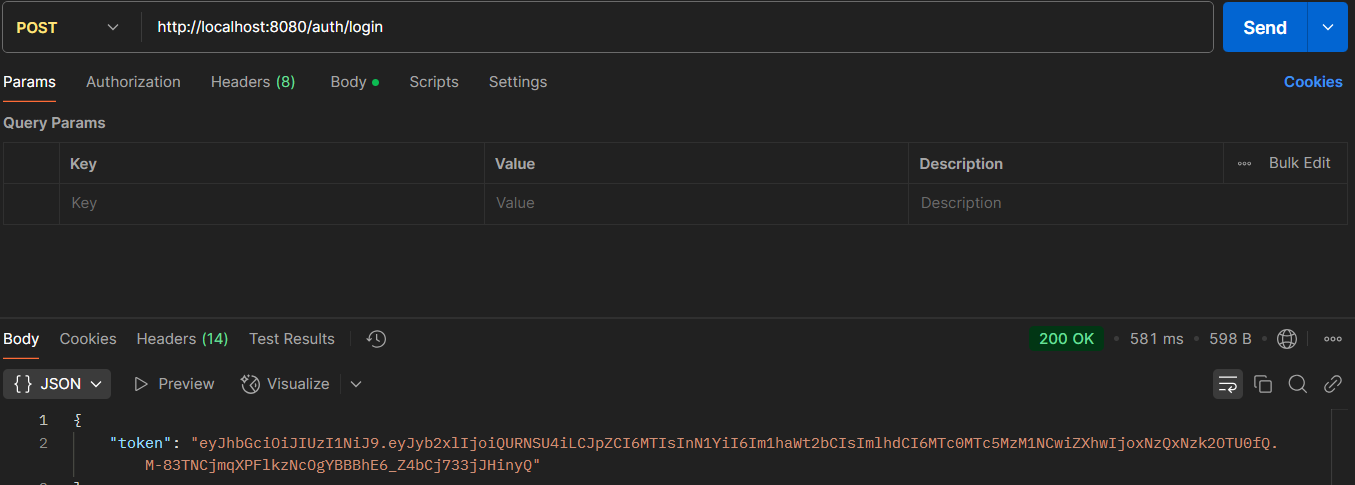


# Pruebas:

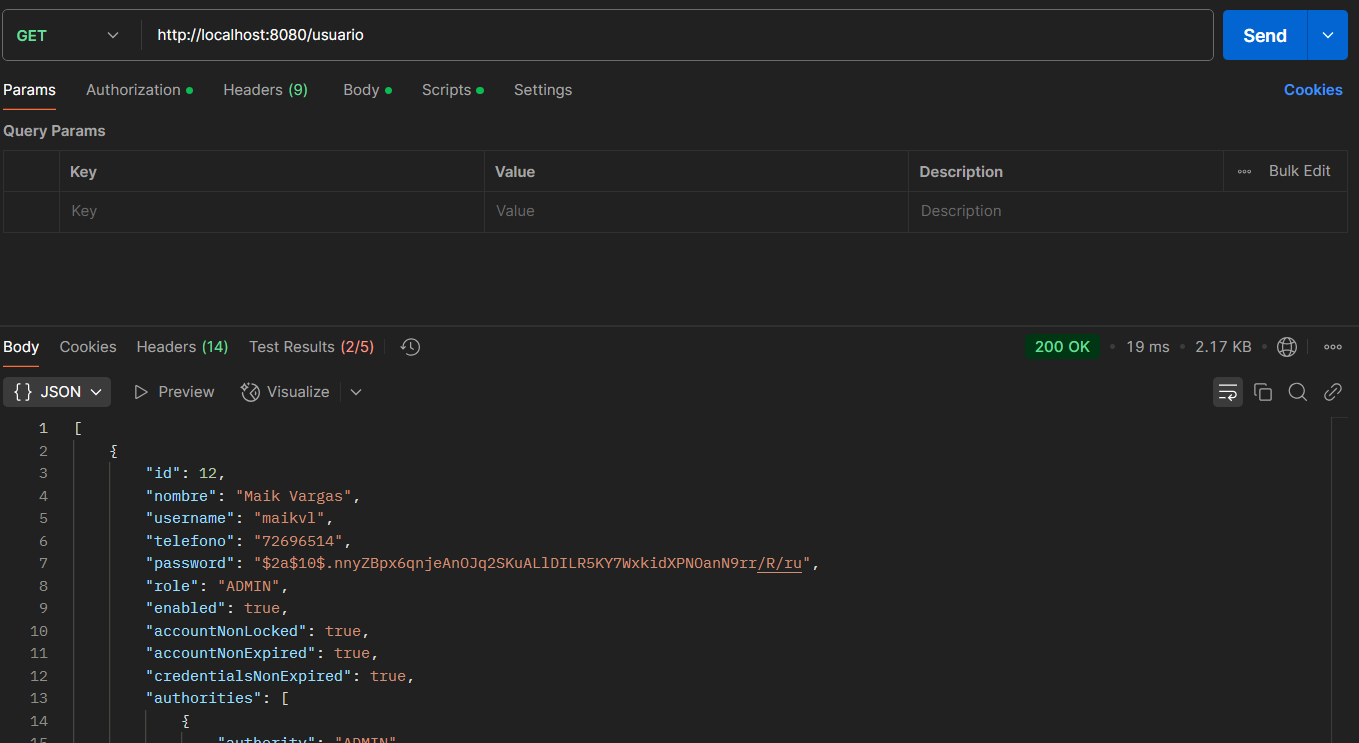
**Pruebas manuales:** Una vez terminamos el desarrollo los dos programadores encargados realizamos diversas pruebas de manera manual, encontrando algunos errores y cuestionándonos entre el equipo de que manera podemos solucionar y simplificar algunos procesos.

**Pruebas por medio de Postman:** Nos fue de mucha utilidad esta herramienta ya que nos permite crear todos los endpoints de la aplicación y comprobar diversos temas como la debida autentificación y creación del token, verificación de la duración del token sin caducar, comprobar la adecuada carga de los datos provenientes de la base de datos.

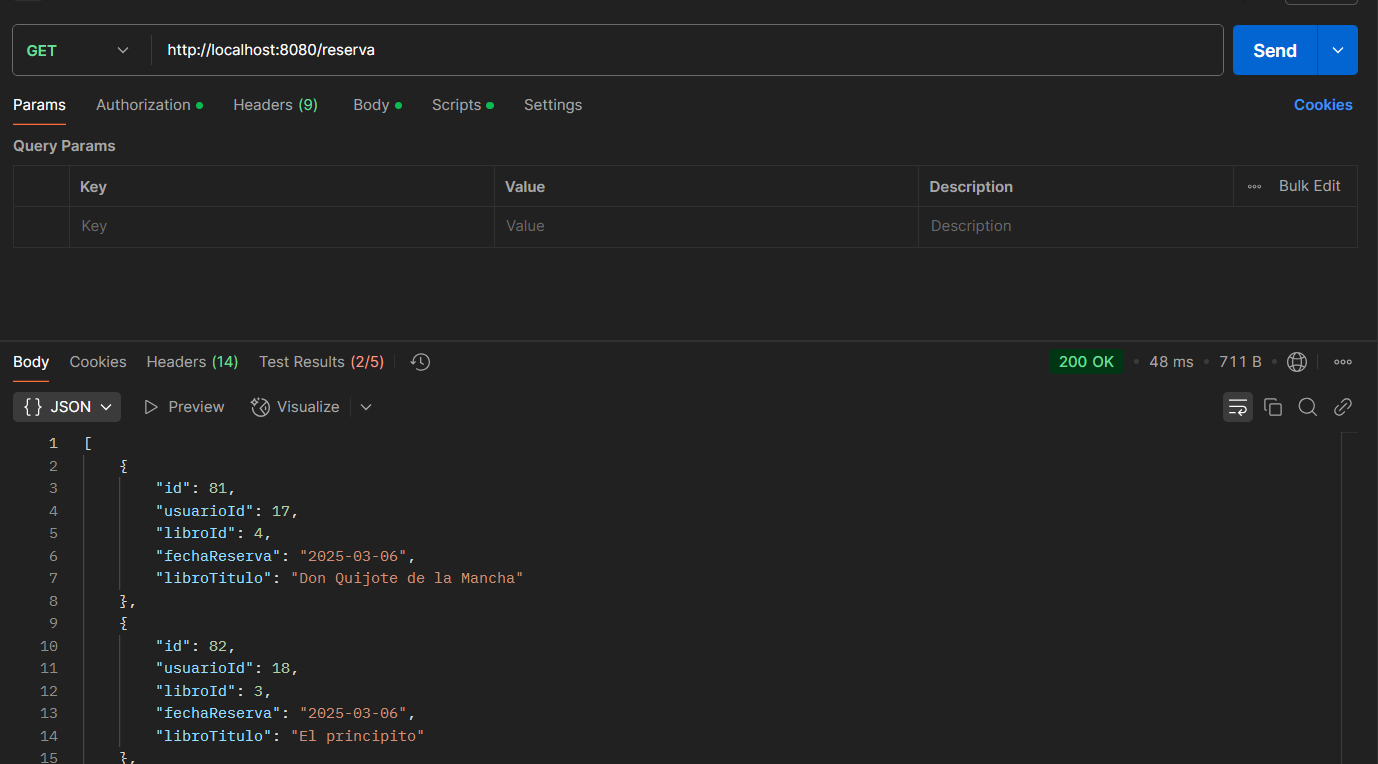
Autentificación y creación del token.



Prueba de los controladores de Libro:

Prueba de controladores de usuario:

Prueba de controladores de reserva:



**Prueba de Usuarios externos:**

Usuario: Alexander Soto Brown, miembro del equipo de informática de la compañía Coopesana.

**Tareas realizadas:**

Prueba de consultas para búsqueda de libros y usuarios.

Creación de libros y usuarios en el sistema.

**Comentarios:**

Rendimiento:

"Estuve probando las diferentes consultas y creando libros y usuarios, el funcionamiento parece estar bien con un rendimiento aceptable. No hubo retrasos notables, y las consultas se ejecutaron rápidamente."

Usabilidad y Diseño:

"El funcionamiento general de la aplicación es correcto, pero sugiero realizar algunos ajustes en el diseño. Los colores de fondo son algo claros y pueden dificultar la visibilidad, especialmente cuando se usan por largos períodos de tiempo. Cambiar algunos tonos a colores menos brillantes o más oscuros podría mejorar la experiencia visual y reducir la fatiga ocular."

Resultado de la prueba:

El usuario probó las funcionalidades principales de la aplicación, incluyendo las consultas y la creación de registros, sin encontrar problemas de rendimiento ni errores técnicos.

Acciones tomadas: Se tomaron en cuenta las sugerencias sobre el color para realizar ajustes en el tema visual de la interfaz, considerando un esquema de colores más amigable para la visibilidad en uso prolongado.