



## PP3: Diseño y Arquitectura de Sistemas

### TP Final: Sistema para la Gestión de Préstamos Bibliotecarios

Integrantes:

- Liz Jara
- Camila Sánchez
- Felipe Ferreyra
- Andrés Nuñez

Profesora: Patricia Litovicius

AÑO: 2025

# Índice

## **1. Descripción del Proyecto**

1.1. Contexto y Problema 1.2. Sistema Propuesto 1.3. Actores Clave 1.4. Análisis FODA

## **2. Objetivos del Proyecto**

2.1. Objetivo General 2.2. Objetivos Específicos

## **3. Alcance y Limitaciones**

3.1. Funcionalidades Incluidas 3.2. Exclusiones y Limitaciones

## **4. Análisis de Usuarios y Experiencia de Usuario (UX)**

4.1. Definición de Perfiles de Usuarios 4.2. User Personas 4.3. Historias de Usuario 4.4. Propuesta de Mockups y Diseño Visual 4.5. Principios de Accesibilidad y Diseño Inclusivo

## **5. Modelado del Sistema (UML)**

5.1. Diagrama de Casos de Uso

## **6. Especificación de Requerimientos**

6.1. Requisitos Funcionales 6.2. Requisitos No Funcionales 6.3. Requisitos de Arquitectura 6.4. Matriz de Trazabilidad

## **7. Recursos Disponibles**

7.1. Roles del Equipo 7.2. Material Tecnológico y Herramientas

## **8. Plan de Trabajo**

8.1. Cronograma con Diagrama de Gantt 8.2. Hitos del Proyecto

## **9. Gestión de Riesgos**

9.1. Identificación de Riesgos Técnicos 9.2. Plan de Mitigación y Contingencia

## **10. Diseño Técnico del Sistema**

10.1. Arquitectura Cliente-Servidor (Diagrama de Despliegue) 10.2. Diagramas UML

## **11. Conclusiones Finales**

11.1. Reflexión del Equipo 11.2. Evaluación de Fortalezas, Debilidades y Roadmap Futuro

# 1. Descripción del Proyecto

## 1.1 Contexto Institucional/Empresarial y Problema Detectado

El caso se desarrolla en una biblioteca universitaria de tamaño mediano, con un catálogo de aproximadamente 10.000 libros disponibles. La biblioteca cuenta con computadoras de acceso público para los estudiantes y conexión a internet estable. Actualmente, las reservas de libros se realizan de manera presencial y manual, lo que genera demoras y errores en la gestión.

Los estudiantes deben acudir físicamente a la biblioteca para verificar la disponibilidad de un libro, lo que ocasiona pérdida de tiempo y saturación en el mostrador de atención. Además, no existe un registro automatizado que garantice la correcta asignación de las reservas, generando conflictos cuando varios usuarios solicitan el mismo ejemplar.

## 1.2 Tipo de Sistema Propuesto

Se propone un sistema cliente-servidor compuesto por una aplicación web para la consulta y reserva de libros, y un panel de administración para el personal de la biblioteca.

## 1.3 Actores Clave (Usuarios Involucrados)

- **Usuario principal:** estudiantes y docentes que buscan reservar libros en línea.
- **Usuarios secundarios:** bibliotecarios encargados de administrar las reservas y controlar las entregas.
- **Usuarios administrativos:** personal de gestión que podrá generar reportes de uso del sistema.

## 1.4 Análisis FODA del Proyecto

Este análisis evalúa los factores internos (Fortalezas y Debilidades) y externos (Oportunidades y Amenazas) del proyecto para implementar un nuevo sistema de gestión en la biblioteca universitaria.

### Fortalezas (Factores Internos Positivos)

- **Solución a una Necesidad Real:** El sistema ataca problemas concretos y evidentes del proceso manual actual, como la lentitud, los errores humanos y la falta de información en tiempo real.
- **Optimización de Recursos:** Automatiza tareas repetitivas, liberando tiempo del personal bibliotecario para que puedan dedicarse a actividades de mayor valor.
- **Centralización y Fiabilidad de la Información:** Crea una única fuente de datos confiable para el inventario de libros y el historial de préstamos.
- **Mejora Directa en la Experiencia del Usuario:** Agiliza significativamente el proceso de préstamo y devolución.
- **Apoyo Institucional:** La disposición de la universidad para invertir en el proyecto es un respaldo fundamental.

### Oportunidades (Factores Externos Positivos)

- **Modernización de la Imagen de la Biblioteca:** Posiciona a la biblioteca como un servicio moderno y eficiente.
- **Base para Crecimiento Futuro:** El sistema puede servir como la primera fase para futuras expansiones (reservas en línea, multas, etc.).
- **Toma de Decisiones Basada en Datos:** La información recopilada permitirá generar estadísticas valiosas.
- **Incremento de la Satisfacción General:** Un servicio más rápido y autónomo aumentará la satisfacción de los usuarios.

### Debilidades (Factores Internos Negativos)

- **Dependencia Tecnológica:** Cualquier fallo técnico podría interrumpir el servicio de préstamos.
- **Resistencia al Cambio:** El personal acostumbrado a procesos manuales puede mostrar resistencia.
- **Alcance Limitado en la Primera Fase:** La exclusión de funcionalidades (como reservas en línea en algunos puntos del brief) puede generar expectativas no cubiertas.
- **Curva de Aprendizaje:** El personal necesitará un período de adaptación.

### Amenazas (Factores Externos Negativos)

- **Riesgos de Seguridad Informática:** El sistema manejará datos personales y debe contar con medidas de seguridad robustas.
- **Obsolescencia Tecnológica:** Si no se planifican actualizaciones, el sistema podría quedar obsoleto.
- **Recortes Presupuestarios Futuros:** La falta de presupuesto a futuro podría impedir el correcto mantenimiento.
- **Infraestructura Tecnológica insuficiente:** Si la red o los servidores de la universidad no son estables, el rendimiento se verá afectado.

## **2. Objetivos de Proyecto**

### **2.1 General**

Implementar un sistema cliente-servidor que permita a los estudiantes consultar el catálogo de libros en tiempo real y realizar reservas en línea, mientras los bibliotecarios gestionan de forma centralizada las solicitudes y el estado de los ejemplares.

### **2.2 Específicos**

- Permitir la consulta en línea de la disponibilidad de libros.
- Facilitar la realización de reservas por parte de los estudiantes.
- Permitir la administración y confirmación de reservas por parte de los bibliotecarios.
- Registrar automáticamente las transacciones para generar reportes básicos.
- Implementar un sistema de autenticación de usuarios.

## **3. Alcance y Limitaciones**

### **3.1 Funcionalidades Incluidas**

- Consulta en línea de disponibilidad de libros.
- Realización de reservas por parte de los estudiantes.
- Confirmación y administración de reservas por parte de los bibliotecarios.
- Registro automático de transacciones y generación de reportes básicos.

### **3.2 Exclusiones y Limitaciones**

- No cubrirá la gestión física del préstamo y devolución de los libros.
- No administrará multas por retrasos.
- No reemplazará el sistema contable ni de inventario general de la biblioteca.

## 4. Análisis de Usuarios y Experiencia de Usuario (UX)

### 4.1 Definición de Perfiles de Usuarios

- **Estudiante/Docente (Usuario Lector):** Usuario principal. Busca material de estudio, necesita rapidez y confirmación de disponibilidad.
- **Bibliotecario:** Usuario de gestión. Necesita administrar el catálogo, confirmar reservas y gestionar préstamos/devoluciones de forma eficiente.
- **Administrador del Sistema:** Usuario técnico. Asegura el funcionamiento, la seguridad y los respaldos del sistema.

### 4.2 Creación de Personas (User Personas)

#### Bibliotecario:

- **Nombre:** Andres, 26 años.
- **Objetivos:** Gestionar préstamos rápidamente, mantener el catálogo actualizado y reducir errores manuales.
- **Frustraciones:** Errores de inventario y pérdida de fichas manuales.

#### Estudiante/Docente:

- **Nombre:** Liz Jara, 21 años, estudiante de Ingeniería.
- **Objetivos:** Consultar disponibilidad en tiempo real, agilizar préstamos y evitar multas.
- **Frustraciones:** Que el material no esté disponible o la búsqueda sea lenta.

#### Administrador del Sistema:

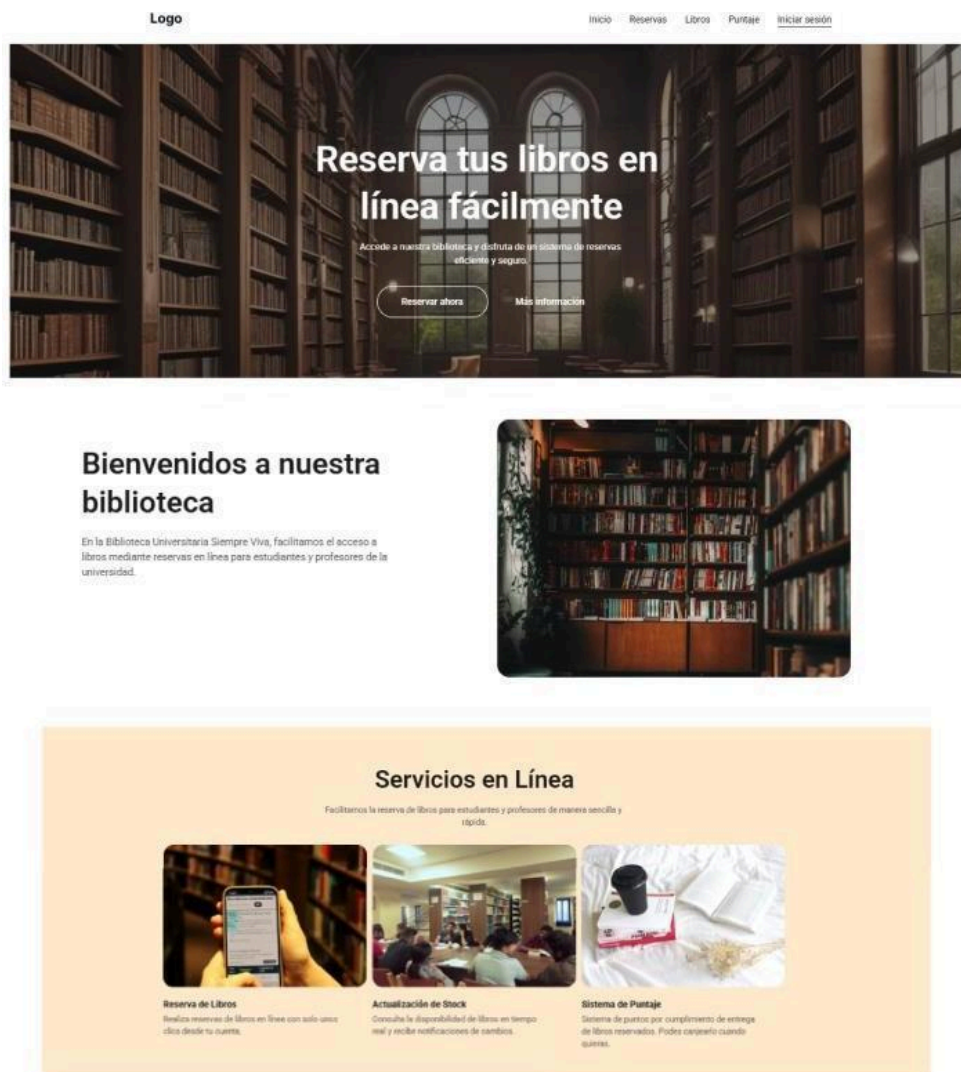
- **Nombre:** Camila Sanchez, 30 años, administradora de sistemas.
- **Objetivos:** Asegurar el funcionamiento continuo, mantener respaldos y seguridad, y escalar el sistema.
- **Frustraciones:** Si el sistema no es modular o escalable.

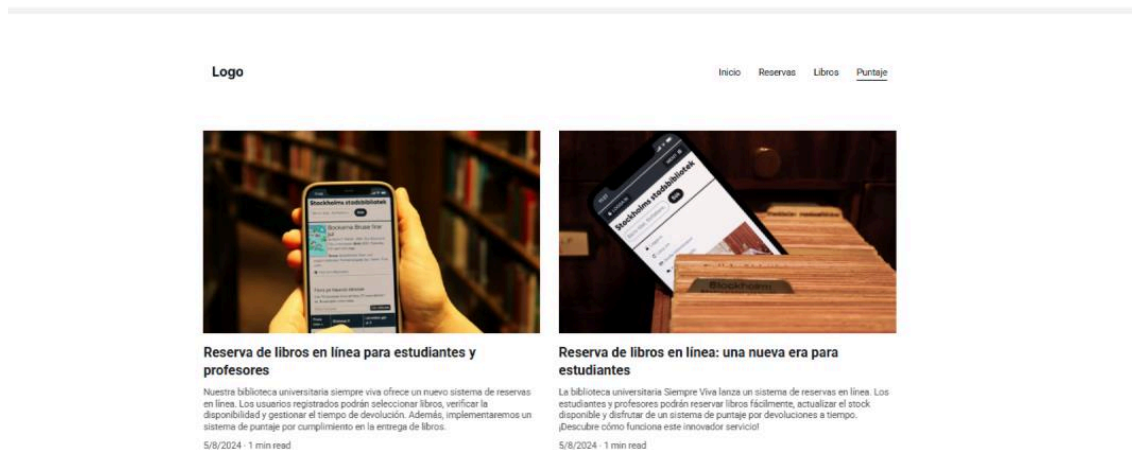
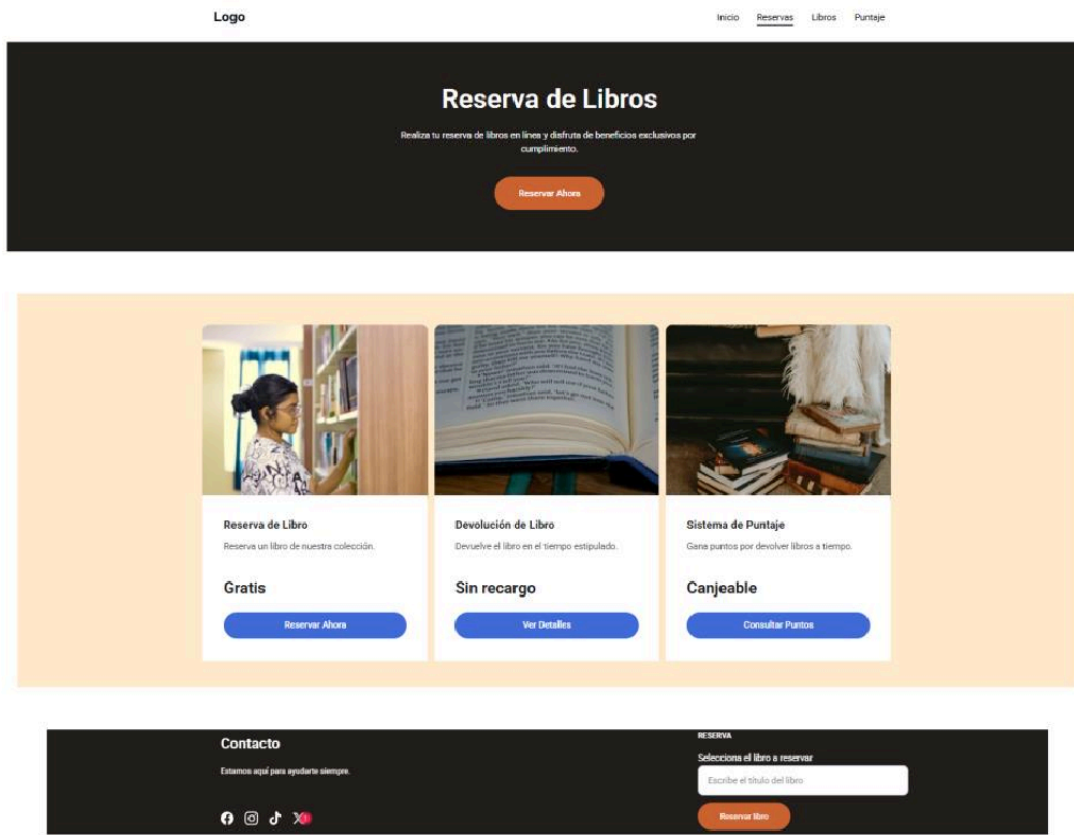
### 4.3 Escenarios de Uso (Historias de Usuario)

- **Docente:** "Como docente universitario, quiero acceder desde la app móvil al catálogo, recibir recomendaciones y gestionar mis préstamos, para enriquecer mis clases y sugerir bibliografía a mis estudiantes".
- **Bibliotecario:** "Como bibliotecario, quiero usar la app para registrar préstamos y devoluciones con códigos QR y consultar reportes, para reducir errores manuales y optimizar la atención".
- **Estudiante:** "Como estudiante, quiero consultar el catálogo desde la app, recibir recordatorios y recomendaciones, para ahorrar tiempo, evitar multas y acceder a material útil".

## 4.4 Propuesta de Wireframes o Mockups Básicos

A continuación, se presentan algunos de los mockups diseñados que representan la estructura y apariencia del sistema de gestión de préstamos de la biblioteca universitaria. Las pantallas muestran las principales secciones del sitio, como el inicio, el catálogo de libros y el área de reserva.





## 4.5 Principios de Accesibilidad y Diseño Inclusivo

- Contraste de color adecuado
- Navegación por teclado
- Etiquetas WAT-ARIA para lectores de pantalla
- Tamaño de fuente legible

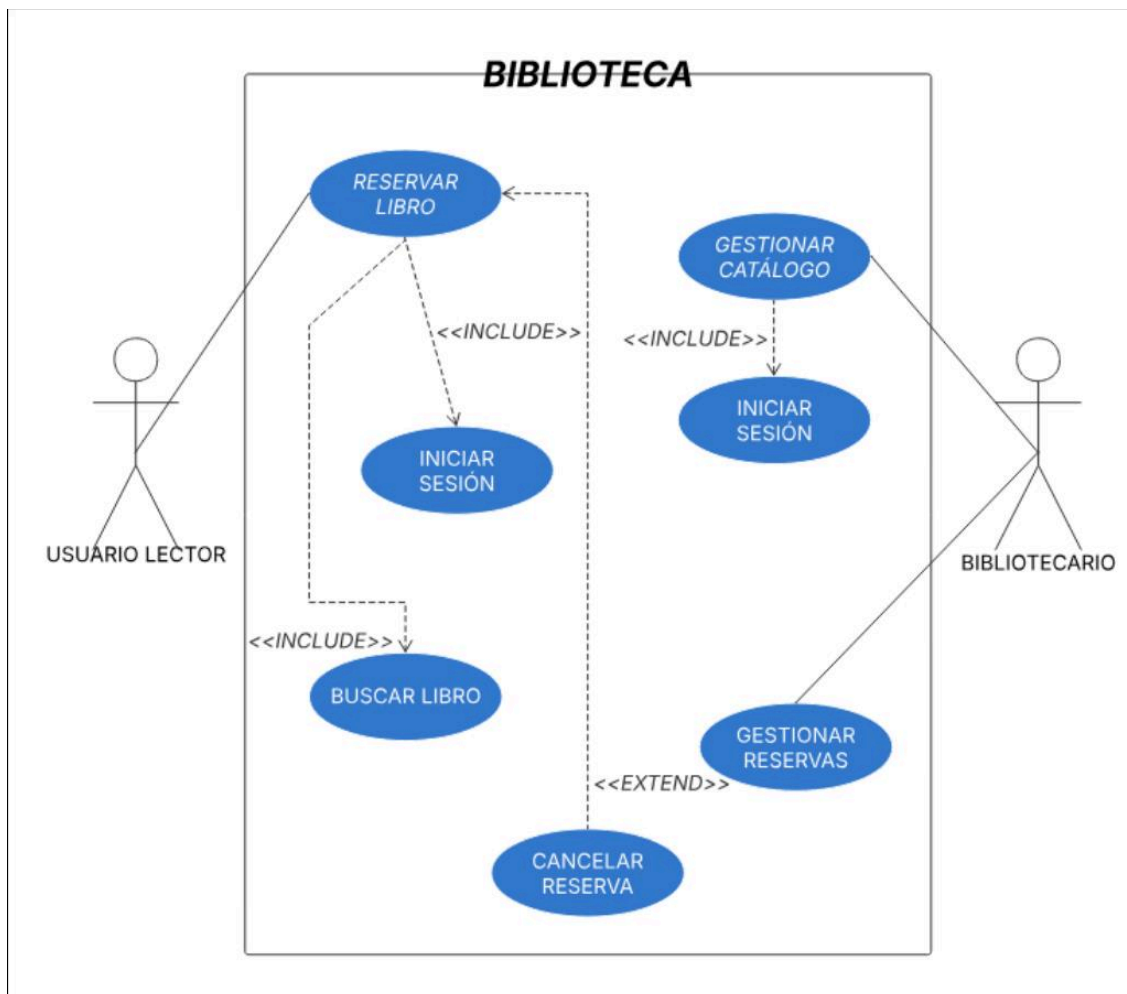


## 5. Casos de Uso (UML)

### 5.1 Diagrama de Casos de Uso

El siguiente diagrama representa los actores principales y las funcionalidades del sistema.

- **Usuario lector:** puede buscar libros en el catálogo y realizar reservas de ejemplares disponibles.
- **Bibliotecario / Administrador:** gestiona las reservas realizadas por los usuarios, actualizando el estado de los libros y confirmando los préstamos.



## **6. Especificación de Requerimientos**

### **6.1 Requisitos Funcionales**

#### **Gestión de Catálogo de Libros**

- RF1.1: Registrar, modificar y dar de baja libros y materiales físicos.
- RF1.2: Consultar la disponibilidad de un libro en tiempo real.
- RF1.3: Mostrar ubicación física del libro dentro de la biblioteca (mapa inteligente).
- RF1.4: Cada ítem del catálogo deberá contar con campos para Título, Autor, e ISBN.

#### **Gestión de Usuarios**

- RF2.1: Registrar, modificar y dar de baja estudiantes, docentes y personal.
- RF2.2: Generar un perfil de lectura dinámico por usuario (temas frecuentes, historial).

#### **Préstamos y Devoluciones**

- RF3.1: Registrar préstamos de libros, asignando fecha de vencimiento automática.
- RF3.2: Registrar devoluciones, actualizando disponibilidad inmediata.
- RF3.3: Notificar al usuario sobre próximos vencimientos vía email o app móvil.
- RF3.4: Permitir escaneo con códigos de barra para agilizar préstamos y devoluciones.

#### **Asistente Virtual y Recomendaciones Inteligentes**

- RF4.1: Implementar un chatbot que responda consultas sobre disponibilidad, vencimientos y búsqueda de materiales.
- RF4.2: Generar recomendaciones personalizadas basadas en el historial de lecturas del usuario.
- RF4.3: Sugerir materiales relacionados a materias o carreras específicas.

#### **Gamificación y Motivación**

- RF5.1: Asignar logros y recompensas a los usuarios por comportamiento.
- RF5.2: Administrar un sistema de puntos que pueda intercambiarse por beneficios internos.

#### **Reportes y Estadísticas Académicas**

- RF6.1: Generar informes sobre libros más prestados, autores más consultados y tendencias académicas.
- RF6.2: Exportar estadísticas para la toma de decisiones de la universidad.
- RF6.3: Registrar auditoría de cambios en catálogos y usuarios.

## 6.2 Requisitos No Funcionales

### Rendimiento y Escalabilidad

- RNF1: El sistema debe responder consultas de disponibilidad en menos de 2 segundos.
- RNF2: Soportar al menos 200 usuarios concurrentes (escalable a más en el futuro).

### Disponibilidad y Confiabilidad

- RNF3: Estar disponible al menos el 99% del tiempo durante horario de biblioteca.
- RNF4: Tener mecanismos de respaldo automático de la base de datos diario.

### Seguridad

- RNF5: Requerir autenticación de usuarios mediante credenciales únicas.
- RNF6: Cifrar la comunicación cliente-servidor para proteger datos personales.
- RNF7: Registrar toda modificación de datos en una bitácora de auditoría.

### Usabilidad

- RNF8: La interfaz debe ser intuitiva y accesible desde PC y dispositivos móviles.
- RNF9: El chatbot debe entender consultas simples en lenguaje natural.

### Mantenibilidad y Portabilidad

- RNF10: El sistema debe estar desarrollado con arquitectura cliente-servidor modular.
- RNF11: Debe ser compatible con navegadores modernos y adaptable a app móvil.

### Innovación y Adaptabilidad

- RNF12: El sistema debe incorporar mecanismos de análisis de datos (learning analytics).
- RNF13: El motor de recomendaciones debe poder mejorarse con algoritmos de IA en futuras versiones.

## 6.3 Requisitos de Arquitectura

### Modelo Cliente-Servidor:

- Cliente (Navegador web y aplicación dedicada)
- Servidor (Servidor de aplicaciones/API REST)
- Base de Datos (PostgreSQL/MySQL).

### Posibles Servicios Externos (APIs):

- API de correo para notificaciones
- API de IA para el chatbot

## 6.4 Matriz de Trazabilidad

Finalmente, se incluye una tabla de trazabilidad que conecta los casos de uso con las actividades del sistema y las pantallas de la interfaz de usuario.

Actor	Caso de Uso	Actividad del Sistema	Pantalla Asociada UX/UI
Estudiante / Docente	Iniciar sesión	Validar credenciales y permitir acceso	Pantalla Login
Estudiante / Docente	Registrarse	Crear cuenta nueva y almacenar datos	Pantalla Registro de Usuario
Estudiante / Docente	Consultar catálogo	Mostrar listado de libros disponibles	Pantalla Libros
Estudiante / Docente	Reservar libro	Registrar la reserva del libro	Pantalla Reservar
Estudiante / Docente	Consultar puntaje	Mostrar puntos acumulados y beneficios	Pantalla Puntaje
Bibliotecario	Registrar préstamo	Registrar la salida del libro	Pantalla Reservar /
Bibliotecario	Registrar devolución	Actualizar disponibilidad del libro	Pantalla Reservar /
Administrador	Gestionar usuarios	Alta, baja o modificación de usuarios	Pantalla Registro / de Gestión
Administrador	Gestionar catálogo	Alta, baja o modificación de libros	Pantalla Libros / de Gestión
Sistema	Notificar vencimiento	Enviar correo o alerta sobre devolución	— (Proceso automatizado)

## 7. Recursos Disponibles

### 7.1 Roles del Equipo

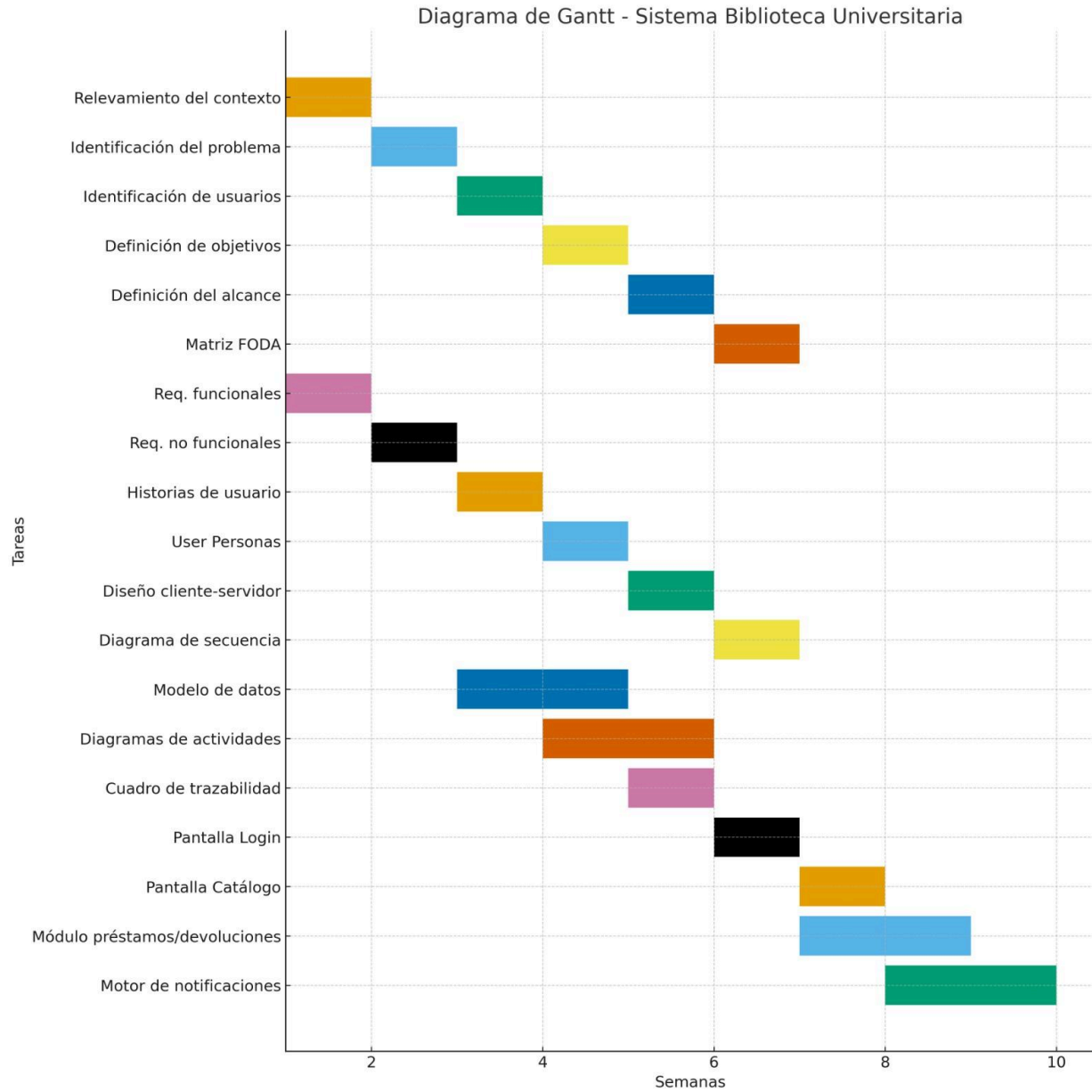
- Liz Jara: Project Manager / Analista
- Camila Sánchez: Diseñadora UX/UI / Tester
- Felipe Ferreyra: Arquitecto de Software / Programador Backend
- Andrés Nuñez: Programador Frontend / Analista

### 7.2 Material Tecnológico y Herramientas

- Software: Figma (Mockups), Visual Studio Code (IDE), Git (Control de versiones), Node.js (Backend), React (Frontend), PostgreSQL (Base de datos)
- Hardware: Computadoras personales de los miembros del equipo
- Servidores: Despliegue en un servicio en la nube como Vercel (Frontend) y Render (Backend) para el prototipo

## 8. Plan de Trabajo

- 8.1 Cronograma con Diagrama de Gantt



## **8.2 Hitos: análisis, diseño, desarrollo, pruebas, entrega**

### **Hito 1: Análisis y Requerimientos (Fecha: 18/08/2025)**

- Relevamiento del contexto
- Identificación del problema
- Identificación de usuarios
- Definición de objetivos
- Definición del alcance
- Matriz FODA
- Requisitos funcionales y Requisitos no funcionales
- Historias de usuario
- User Personas

### **Hito 2: Diseño de UX/UI y Arquitectura (Fecha: 22/09/2025)**

- Diseño cliente-servidor
- Modelo de datos
- Diagramas de actividades
- Diagrama de secuencia
- Cuadro de trazabilidad
- Diseño de Pantalla Login
- Diseño de Pantalla Catálogo

### **Hito 3: Desarrollo (Sprint 1 - Funcionalidades clave) (Fecha: 17/03/2026)**

- Infraestructura y Base de Datos
- Implementación del Módulo de préstamos/devoluciones
- Desarrollo de API de Autenticación
- Desarrollo de API de Gestión de Catálogo
- Desarrollo del Motor de notificaciones

### **Hito 4: Pruebas y QA (Fecha: 18/05/2026)**

- Ejecución de Plan de Pruebas
- Reporte de Bugs e Incidencias
- Pruebas de Integración

### **Hito 5: Entrega final y Presentación (Fecha: 30/06/2026)**

- Integración final
- Cierre del Proyecto.
- Defensa Oral

## **9. Gestión de Riesgos**

### **9.1 Riesgos Técnicos Identificados**

Al analizar el sistema, hemos encontrado que es posible que se presenten los siguientes inconvenientes:

- Falla del servidor
- Corte de energía
- Pérdida de datos
- Ataque informático
- Error humano
- Pérdida de red
- Daño físico (incendio, robo, etc)

### **9.2 Propuestas de Mitigación y Contingencia**

Por lo tanto, hemos diseñado el siguiente plan de contingencias:

- Reemplazar el servidor y restaurar la copia de seguridad
- Usar UPS y retomar el servicio al restablecer la energía
- Restaurar la última copia de seguridad disponible
- Aislar el sistema, limpiar archivos dañados y restaurar backup
- Recuperar datos desde la copia de respaldo o registros de auditoría
- Revisar el cableado, router o switch y reiniciar servicios
- Activar plan de recuperación en otro equipo o servidor

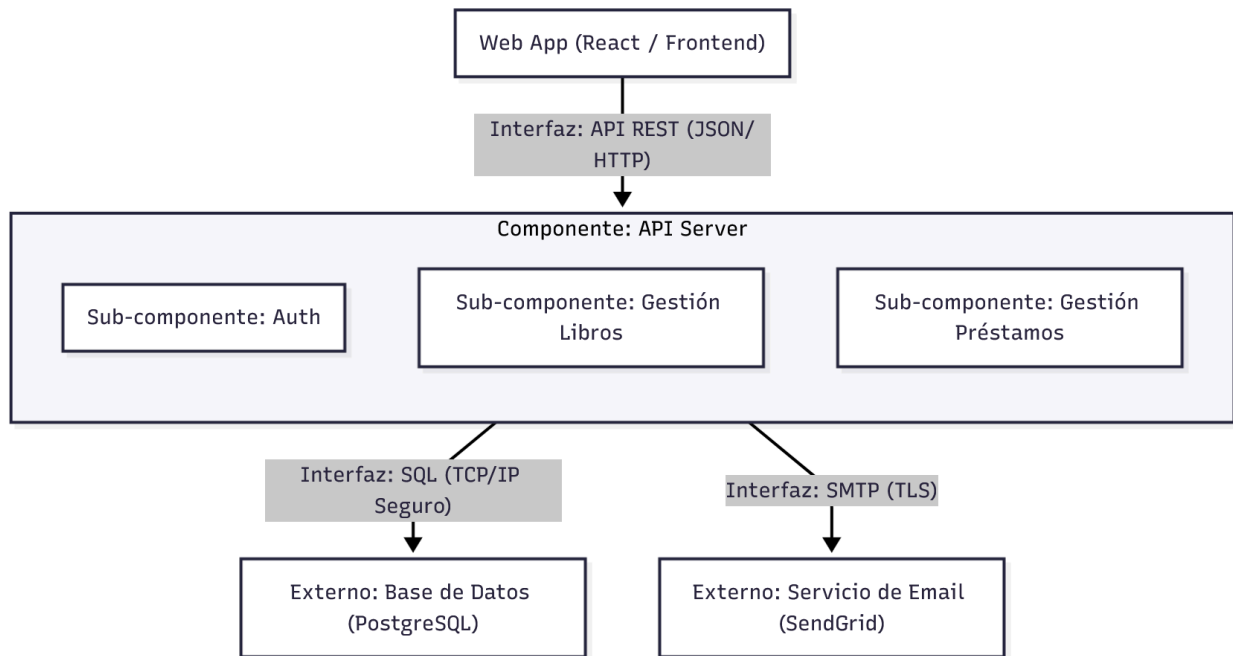
Sin embargo, para prevenir dichas contingencias, hemos establecido las siguientes medidas:

- Realizar copias de seguridad diarias.
- Mantener actualizado el antivirus y firewall.
- Capacitar al personal en uso correcto del sistema.
- Tener una UPS o sistema de respaldo eléctrico.
- Guardar copias de seguridad fuera del edificio.
- Actualizar el software periódicamente.

## 10. Diseño del Sistema

### 10.1 Arquitectura Cliente-Servidor

- Representación Gráfica (Diagrama de Despliegue)



- Justificación de la Arquitectura Elegida

- Se eligió una arquitectura de microservicios modular basada en una API REST por su escalabilidad, separación de responsabilidades entre frontend y backend, y facilidad para conectar múltiples clientes (web, app móvil) a futuro.

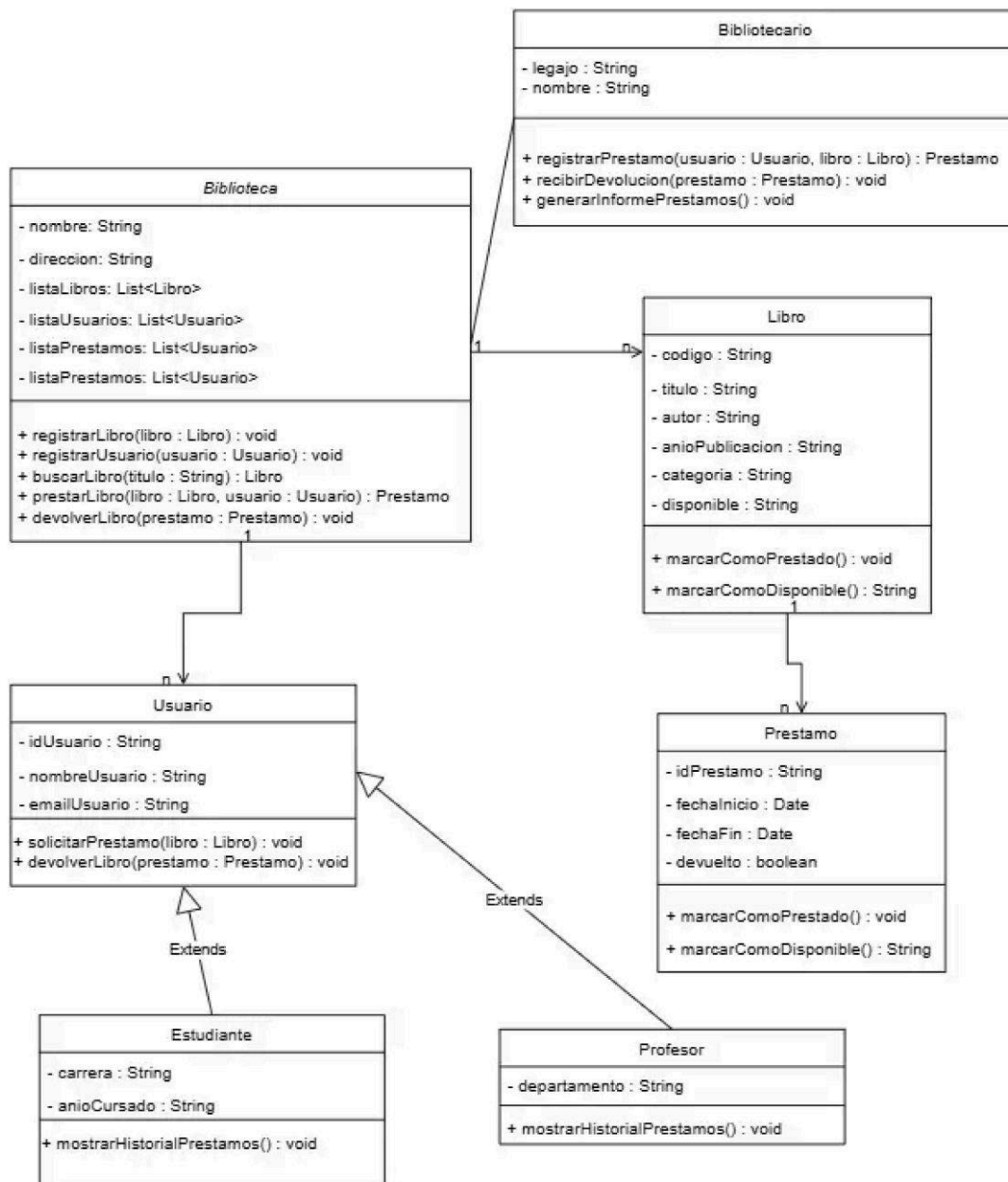


## Diagramas UML

- Diagrama de Clases

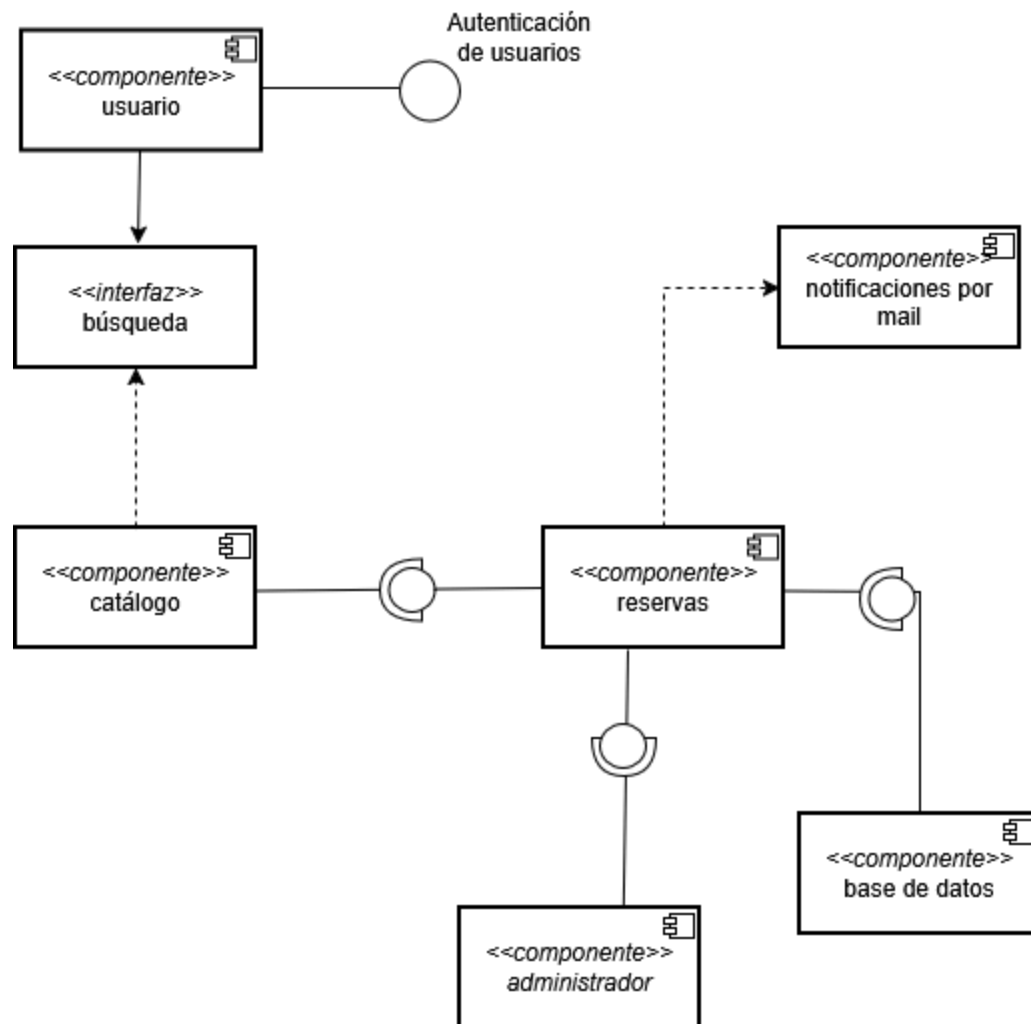
Este diagrama modela la estructura estática del sistema. Define las clases principales, sus atributos y métodos.

**Clases incluidas:** Biblioteca, Bibliotecario, Libro, Usuario, Préstamo, Estudiante y Profesor.



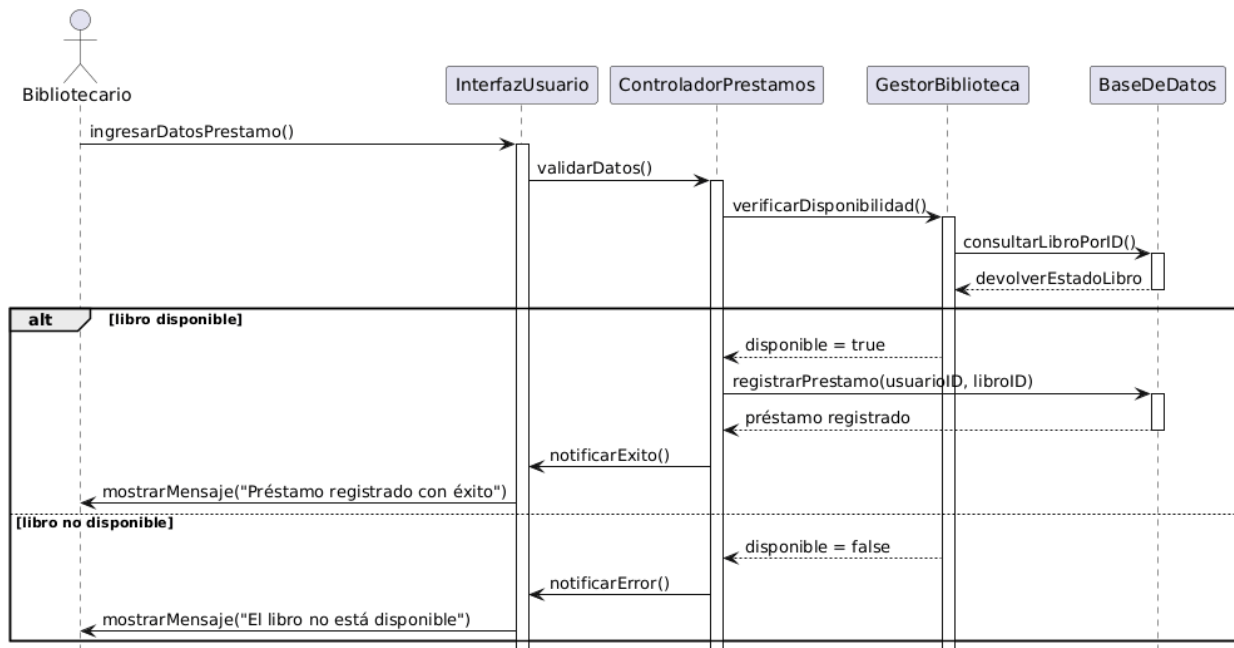
- **Diagrama de Componentes**

A continuación se presenta el diagrama de componentes del Sistema



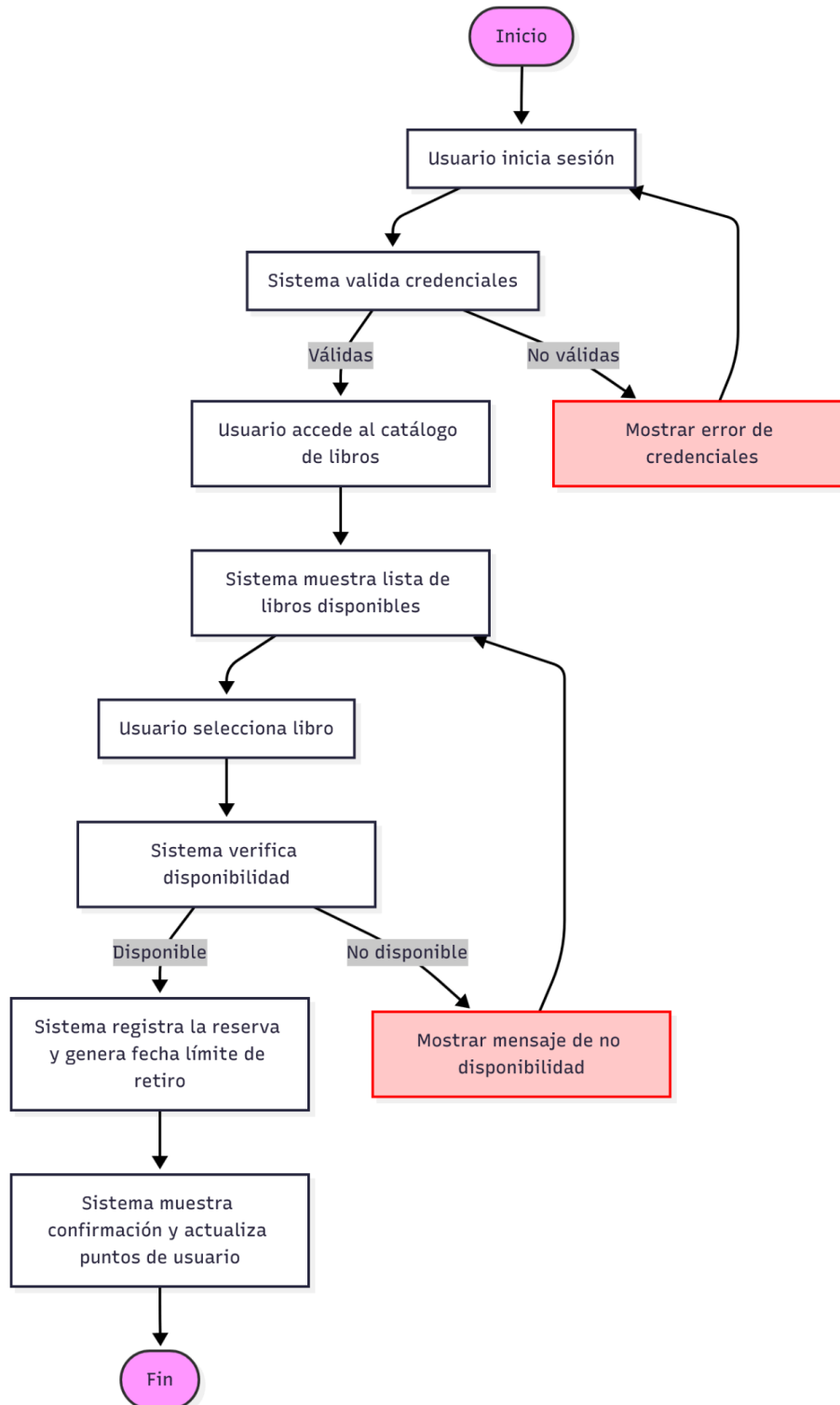
- **Diagrama de Secuencia: Gestión de Préstamos**

Este diagrama detalla el flujo de interacciones para el caso de uso "Gestión de préstamos de libros". Muestra la secuencia de mensajes entre el Bibliotecario, la InterfazUsuario, el ControladorPrestamos, el GestorBiblioteca y la BaseDeDatos.



- **Diagrama de Actividades**

- Descripción del flujo: El estudiante inicia sesión, consulta el catálogo, selecciona un libro y realiza la reserva. El sistema valida la disponibilidad, registra la reserva y confirma al usuario.



## Interfaces de Usuario (UI)

[Mockup de las pantallas del sistema \(Figma\)](#)

# 11. Conclusiones Finales

## • 11.1 Reflexión del Equipo Sobre el Aprendizaje Logrado

El desarrollo del "Sistema para la Gestión de Préstamos Bibliotecarios" ha representado para el equipo una oportunidad para aplicar y consolidar los conocimientos adquiridos durante la cursada de Práctica Profesionalizante III.

A través de este proyecto, hemos logrado:

- **Integrar el Ciclo de Vida del Software:** Comprendimos la importancia de no saltar directamente a la programación, valorando las etapas previas de análisis de requerimientos y modelado UML (Casos de Uso, Clases, Secuencia) como herramientas fundamentales para evitar errores costosos en el desarrollo.
- **Transición de lo Manual a lo Digital:** Enfrentamos el desafío real de digitalizar un proceso completamente manual, entendiendo cómo la tecnología puede transformar la experiencia del usuario (estudiantes y bibliotecarios) y optimizar tiempos operativos.
- **Trabajo Colaborativo y Roles:** La división de roles (Backend, Frontend, UX/UI, Gestión) nos permitió simular un entorno laboral realista, aprendiendo a coordinar las entregas y a mantener la coherencia entre la interfaz visual y la lógica de negocio.

## • 11.2 Evaluación de Fortalezas, Debilidades y Mejoras Futuras del Sistema

Tras finalizar el análisis y diseño del prototipo, realizamos la siguiente evaluación técnica y funcional del producto entregado:

- **Fortalezas del Sistema:**
  - **Experiencia de Usuario (UX) Intuitiva:** Gracias al diseño centrado en el usuario y el uso de interfaces limpias, hemos reducido significativamente la curva de aprendizaje para los bibliotecarios que venían de un sistema manual.
  - **Integridad de Datos:** La arquitectura de base de datos relacional garantiza que no existan duplicidades ni conflictos en las reservas, solucionando el problema principal de la biblioteca.
  - **Escalabilidad:** La arquitectura cliente-servidor separada (Frontend en React / Backend en Node.js) permite que el sistema crezca en tráfico sin necesidad de reestructurar todo el código.

- **Debilidades y Limitaciones:**
  - **Gestión de Multas y Sanciones:** El sistema no automatiza el cobro o gestión de multas por retrasos, requiriendo aún intervención manual para estos casos.
  - **Dependencia de Conectividad:** Al ser una aplicación web, su funcionamiento depende estrictamente de la conexión a internet de la universidad; una caída de red inhabilita las operaciones de préstamo en tiempo real.
- **Mejoras Futuras (Roadmap):**
  - **Implementación de IA para Recomendaciones:** Integrar algoritmos de aprendizaje automático para sugerir libros a los estudiantes basándose en su historial real de lectura.
  - **App Móvil Nativa:** Desarrollar una aplicación dedicada (Android/iOS) que permita notificaciones push nativas y escaneo de códigos QR utilizando la cámara del celular de forma más fluida.
  - **Integración con Sistemas Académicos:** Conectar la base de datos de usuarios con el sistema de gestión de alumnos de la universidad para automatizar completamente las altas y bajas de estudiantes cada semestre.