

PROYECTO FINAL METODOLOGÍA Y SISTEMAS

Profesores: Leandro Schmidt

Ayudante: Wagner Benjamin

Materia: Metodología de sistemas

Fecha de entrega: 31 de julio

Integrantes del grupo:

- Devaux Franco
- Perrone Eros

Páginas: 15 (no contamos la portada)

ÍNDICE

Proyecto elegido: "Sistema de Gestión de Biblioteca Online"	2
Punto 1----->	2
Punto 2----->	3
Punto 3----->	6
Punto 4----->	7
Punto 5----->	8
Punto 6----->	9
Punto 7----->	10
Punto 8----->	12
Punto 9----->	12
Punto 10----->	13
Punto 11----->	14
Punto 12 ----->	15

Punto 1: Realizar un análisis de requerimientos funcionales y no funcionales del sistema.

Descripción breve del proyecto: El sistema permitirá a una biblioteca gestionar el préstamo de libros de forma online, permitiendo a los usuarios registrarse, consultar catálogos, solicitar préstamos y renovar libros desde su cuenta, mientras que los administradores podrán gestionar el inventario, ver reportes y controlar las devoluciones.

Requerimientos Funcionales:

- Registro e inicio de sesión de usuarios: Los usuarios podrán crear una cuenta con sus datos personales e iniciar sesión para utilizar el sistema.
- Búsqueda y consulta de libros: Los usuarios podrán buscar libros por título, autor o género, consultar disponibilidad y ver la descripción del libro
- Solicitud de préstamo online: Los usuarios podrán solicitar préstamos de libros de forma online, seleccionando el libro deseado y el período de préstamo disponible.
- Renovación de préstamos: Los usuarios podrán renovar un préstamo antes de su vencimiento, siempre que el libro no haya sido solicitado por otro usuario.
- Devolución de libros: El sistema permitirá al usuario registrar la devolución del libro o notificar la intención de devolución para coordinar con la biblioteca.
- Gestión de inventario por administradores: Los administradores podrán agregar, modificar o eliminar libros del catálogo, incluyendo título, autor, ejemplares disponibles, imagen de portada y categoría.
- Notificaciones automáticas: El sistema enviará recordatorios por email antes de la fecha de devolución y notificará sobre renovaciones exitosas o vencimientos.
- Reportes para administración: Generación de reportes de libros más solicitados, préstamos activos y estado del inventario.

Requerimientos No Funcionales:

- Usabilidad: El sistema tendrá una interfaz simple e intuitiva para facilitar su uso por personas de diferentes edades.
- Disponibilidad: El sistema estará disponible las 24 horas, con una disponibilidad mínima del 80%.
- Tiempo de respuesta: El sistema debe responder en menos de 3 segundos por consulta de libro.
- Compatibilidad: Accesible desde dispositivos móviles, tablets y computadoras.
- Seguridad: Los datos de los usuarios deben estar protegidos con autenticación segura.

Los préstamos y devoluciones se registran con fecha y hora exacta para trazabilidad.

El sistema cumplirá con la protección de datos personales.

- Escalabilidad: El sistema podrá ampliarse para más usuarios y libros sin comprometer el rendimiento.
- Mantenibilidad: El sistema será modular para facilitar actualizaciones y correcciones de errores.
- Respaldo de datos: El sistema realizará respaldos automáticos diarios para evitar pérdida de información.

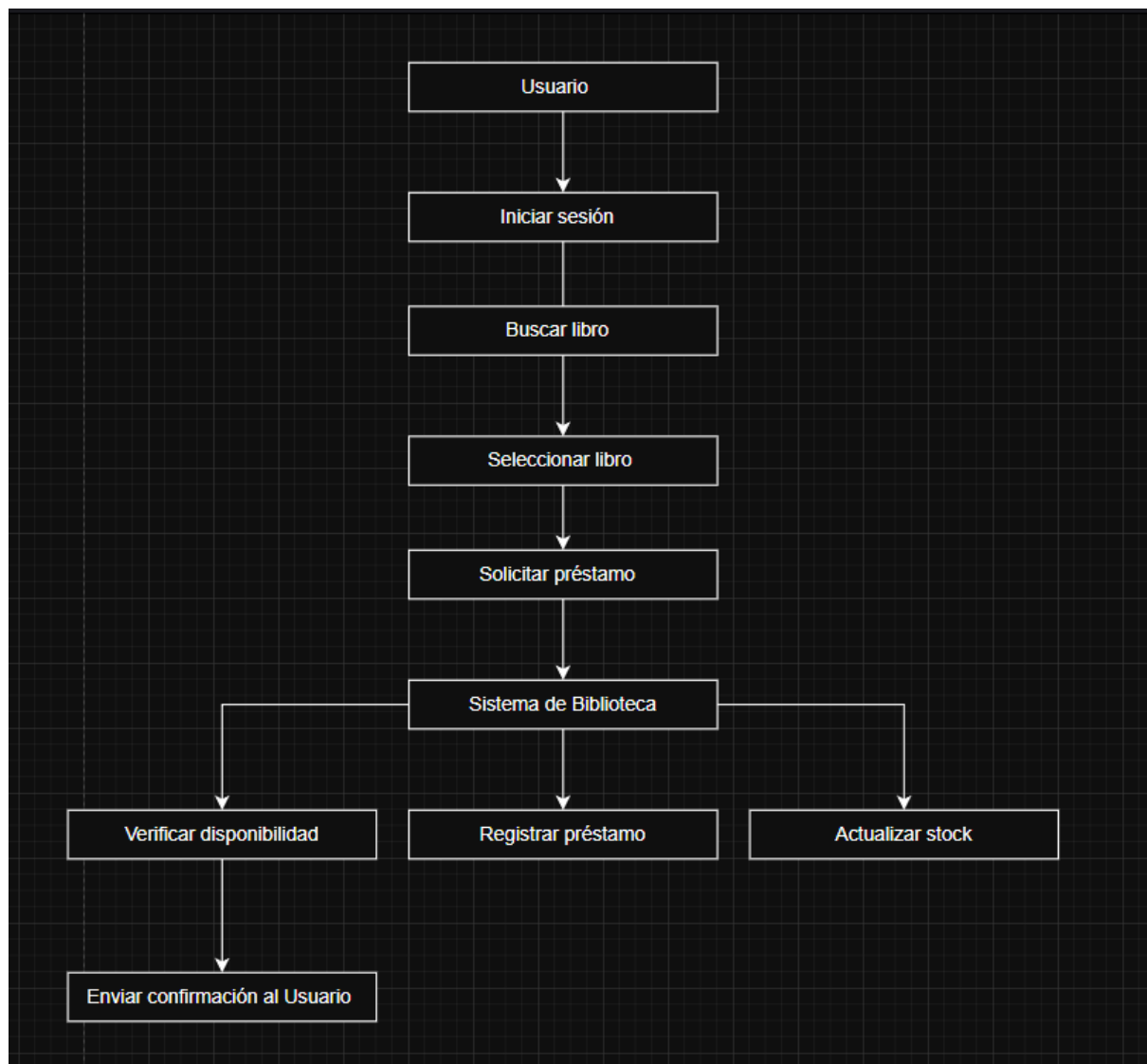
Punto 2: Elaborar al menos 3 diagramas de casos de uso que reflejen las funcionalidades principales. Identificar actores y definir escenarios de interacción.

Caso de Uso 1: Solicitar Préstamo de Libro Online

Actores: Usuario (principal): inicia la solicitud de préstamo. Sistema de Biblioteca (secundario): gestiona la solicitud y actualiza el inventario.

Pasos del diagrama:

1. El Usuario ingresa al sistema (inicia sesión).
2. El Usuario busca y selecciona el libro deseado.
3. El Usuario solicita el préstamo del libro.
4. El Sistema verifica disponibilidad del libro.
5. El Sistema registra el préstamo con fecha de vencimiento.
6. El Sistema actualiza el stock de libros disponibles.
7. El Sistema envía confirmación de préstamo al Usuario.



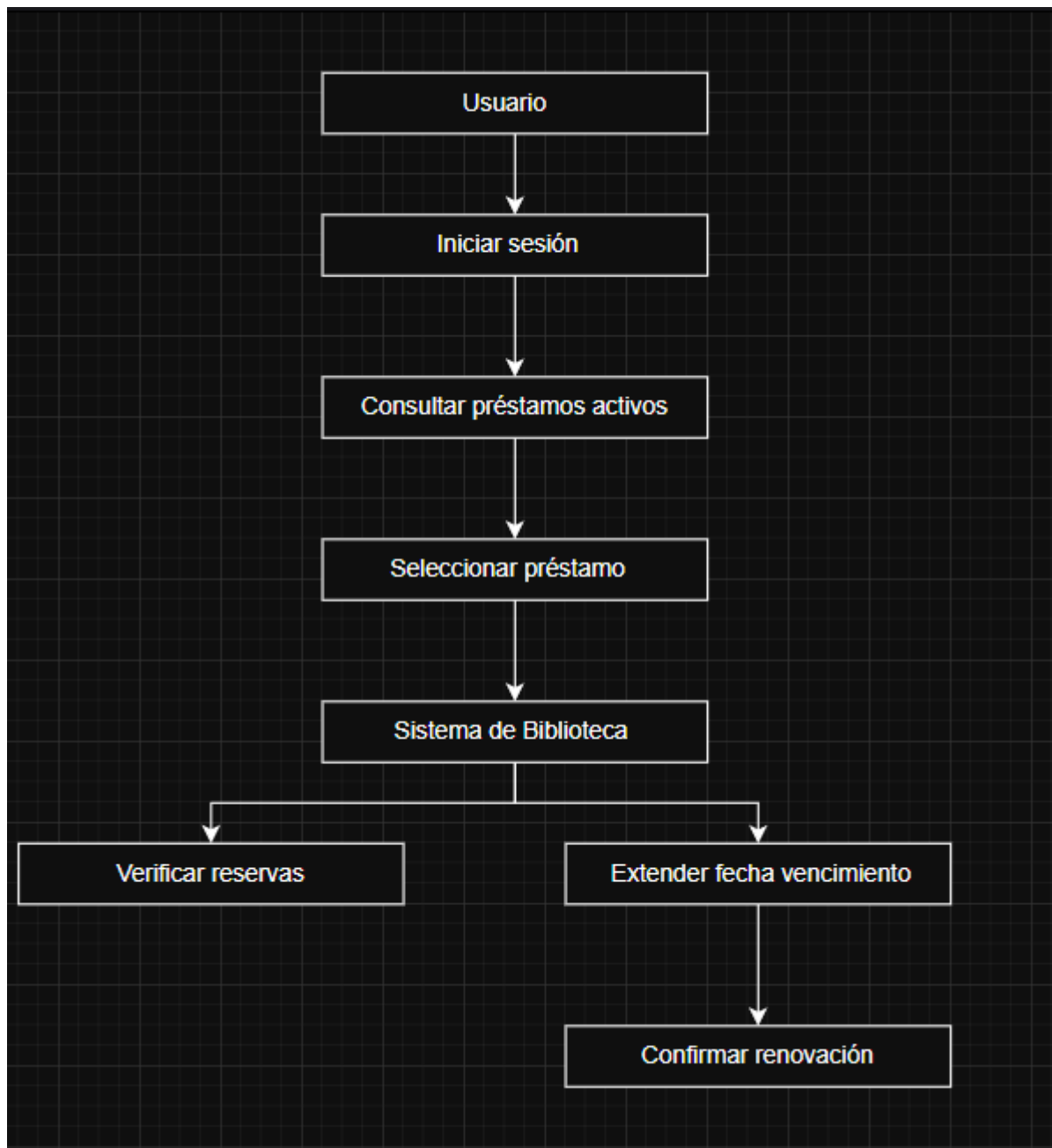
Caso de Uso 2: Renovar Préstamo de Libro

Actores: Usuario (principal): realiza la renovación.

Sistema de Biblioteca (secundario): verifica y actualiza el préstamo.

Pasos del diagrama:

1. El Usuario inicia sesión en el sistema.
2. El Usuario consulta su lista de préstamos activos.
3. El Usuario selecciona el préstamo que desea renovar.
4. El Sistema verifica si el libro no tiene reservas de otro usuario.
5. El Sistema extiende la fecha de vencimiento.
6. El Sistema confirma la renovación al Usuario.



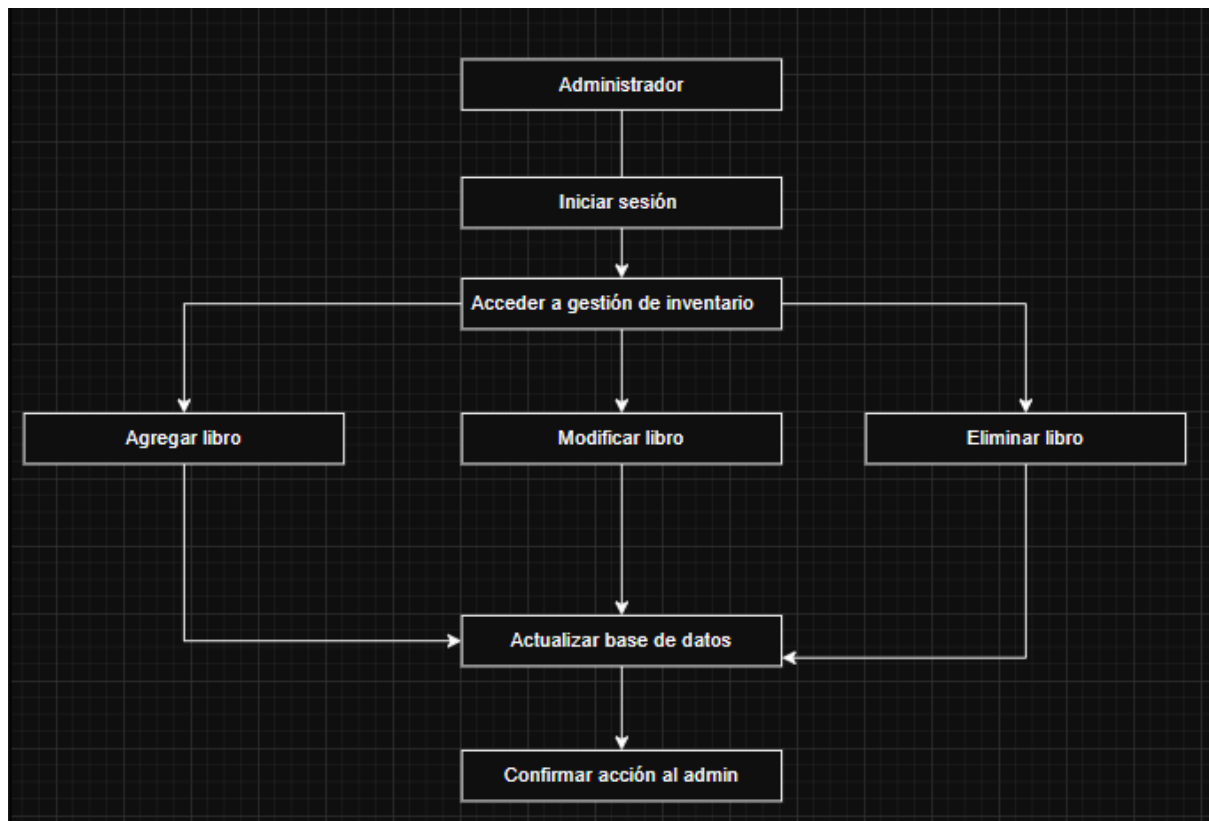
Caso de Uso 3: Gestionar Inventario de Libros

Actores: Administrador (principal): gestiona el catálogo.

Sistema de Biblioteca (secundario): guarda y actualiza datos

Pasos del diagrama:

1. El Administrador inicia sesión en el sistema.
2. El Administrador selecciona la opción de gestionar inventario.
3. El Administrador puede:
4. Agregar un nuevo libro al catálogo.
5. Modificar información de un libro existente.
6. Eliminar un libro del catálogo.
7. El Sistema actualiza la base de datos con los cambios realizados.
8. El Sistema confirma la acción al Administrador.



Punto 3: Crear al menos 2 diagramas de actividad para modelar procesos y flujos de trabajo.

Diagrama de Actividad 1: Solicitar Préstamo de Libro Online

Descripción del proceso:

Representa los pasos que realiza un usuario para solicitar un préstamo online, desde el inicio de sesión hasta la confirmación del préstamo.

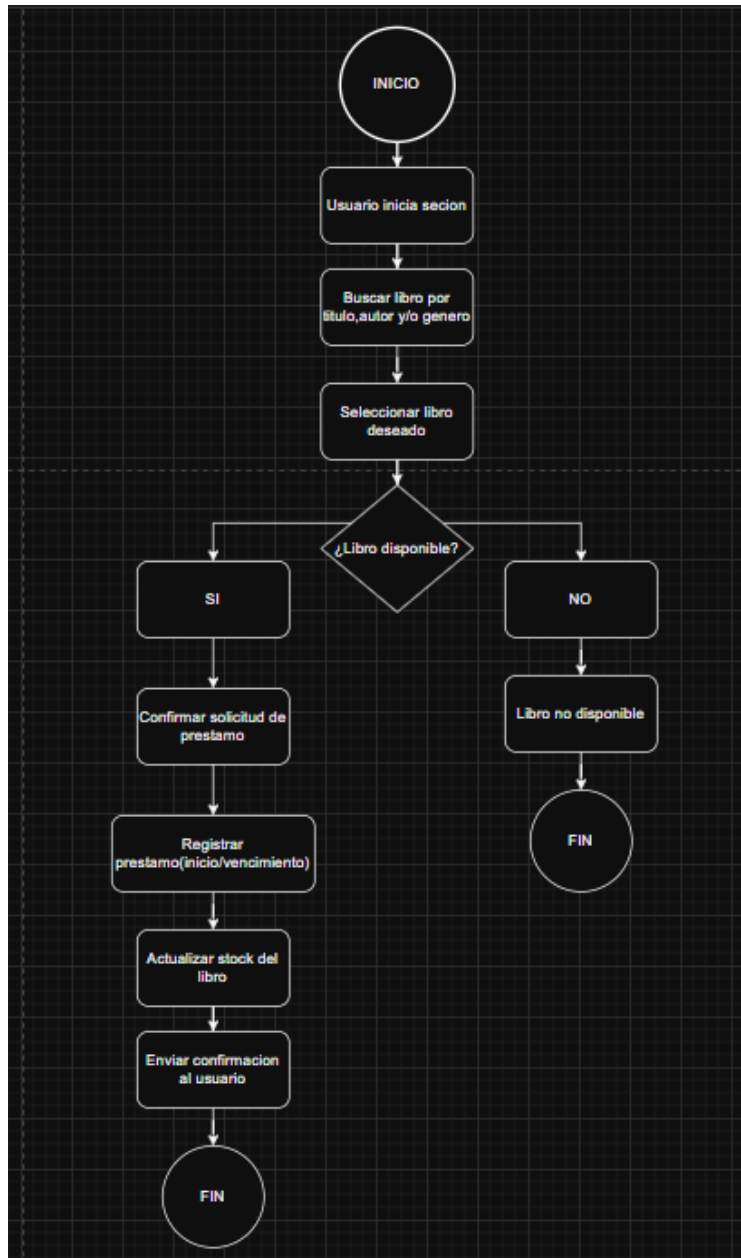
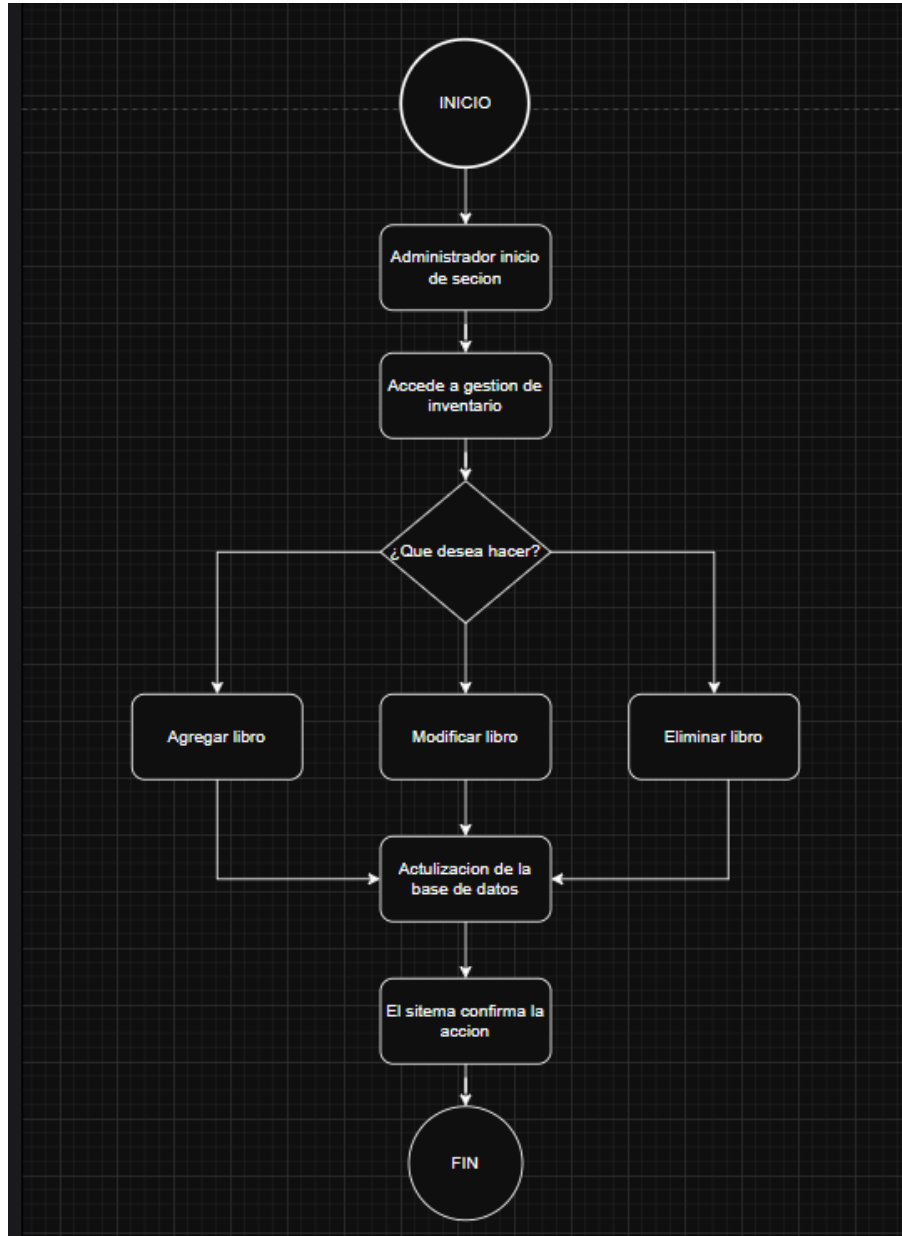


Diagrama de Actividad 2: Gestionar Inventario de Libros

Descripción del proceso:

Este proceso representa las acciones que puede realizar un administrador para gestionar el inventario (agregar, modificar o eliminar libros).



Punto 4: Definir el alcance, los objetivos y los entregables del proyecto.

Alcance del proyecto

El proyecto consiste en desarrollar un Sistema de Gestión de Biblioteca Online que permita a los usuarios consultar el catálogo, solicitar préstamos, renovar préstamos y gestionar devoluciones de forma digital, mientras que los administradores podrán gestionar el inventario y consultar reportes de préstamos.

El sistema será accesible desde computadoras y dispositivos móviles y contará con notificaciones automáticas para recordar a los usuarios las fechas de devolución de libros.

Objetivos del proyecto

Objetivo general:

- Facilitar la gestión de préstamos de libros de una biblioteca mediante un sistema online, reduciendo tiempos de espera.

Objetivos específicos:

- Permitir que los usuarios puedan consultar el catálogo y disponibilidad de libros de forma online.
- Brindar la opción de solicitar y renovar préstamos sin necesidad de ir físicamente a la biblioteca.
- Permitir a los administradores gestionar el inventario de forma eficiente.
- Enviar notificaciones automáticas a los usuarios sobre vencimientos y renovaciones.
- Generar reportes para la administración sobre los préstamos y el estado del inventario.

Entregables del proyecto

Los entregables definidos para el proyecto serán:

1. Documento de requerimientos: Incluyendo los funcionales y no funcionales, con diagramas de casos de uso y de actividad.
2. Cronograma de desarrollo: Planificación semanal del proyecto con tareas asignadas al equipo.
3. Diagrama entidad-relación de la base de datos.
4. Métricas de avance y calidad: Documento con las métricas elegidas y su justificación.
5. Documento final del proyecto: Con todo el análisis, diagramas, métricas y conclusiones para entrega al profesor.

Punto 5: Elaborar un cronograma utilizando diagramas de Gantt o metodologías ágiles como Scrum.

Metodología seleccionada: Scrum

Para este proyecto se utilizará Scrum por su flexibilidad y adaptabilidad, permitiendo realizar entregas parciales de funcionalidades y recibir retroalimentación continua, asegurando calidad y avance constante.

Semana	Sprint	Actividades Principales	Responsable
--------	--------	-------------------------	-------------

Semana 1	Sprint 1	<ul style="list-style-type: none"> • Reunión de planificación • Análisis de requerimientos (funcionales y no funcionales) • Identificación de actores y casos de uso • Definición de alcance y objetivos • Asignación de roles 	Todo el equipo
Semana 2	Sprint 2	<ul style="list-style-type: none"> • Diagramas de casos de uso • Diagramas de actividad • Diagrama entidad-relación • Revisión de diagramas en equipo 	Eros (Dev) Franco y Titi (revisión)
Semana 3	Sprint 3	<ul style="list-style-type: none"> • Diseño de mockups o interfaz (Figma/papel) • Estructura de secciones (usuario y admin) • Revisión del flujo del sistema 	Eros Franco y Titi (feedback)
Semana 4	Sprint 4	<ul style="list-style-type: none"> • Definición de métricas de calidad y avance • Planificación de pruebas (unitarias/funcionales) • Organización de documentación • Validación de consistencia en los diagramas 	Titi
Semana 5	Sprint 5	<ul style="list-style-type: none"> • Revisión general del documento final • Aplicación de ajustes sugeridos • Preparación para entrega y simulacro de defensa 	Franco (Scrum Master) Eros y Titi (apoyo)

Barra De Avance:

Sprint	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
Sprint 1	████████				
Sprint 2		████████			
Sprint 3			████████		
Sprint 4				████████	
Sprint 5					████████

Punto 6: Identificar los recursos necesarios y gestionar el equipo de trabajo, describiendo quiénes son los integrantes del equipo, su roles y responsabilidades.

Recursos necesarios para el proyecto

Hardware:

- 3 computadoras portátiles (una por integrante).
- Acceso a internet estable para trabajo colaborativo.

Software y herramientas:

- Google Drive: para almacenar y compartir documentos del proyecto.
- draw.io : para diseñar diagramas de casos de uso, actividad y ER.
- Google Docs / Word: para redactar el documento final.
- Canva (opcional): para diseño de portada y presentación.
- Zoom / Google Meet: para reuniones de planificación semanal.
- Trello / Google Sheets: para gestionar tareas y avances de sprints.
- Email: para comunicación y envío de avances.

Gestión del equipo de trabajo

El proyecto se desarrollará en equipo de 3 integrantes, con roles claros para facilitar la organización y responsabilidad de cada tarea.

Integrantes del equipo:

Franco Devaux – Scrum Master

- ❖ Planifica y organiza las reuniones semanales.
- ❖ Supervisa el avance de cada sprint.
- ❖ Facilita la comunicación entre integrantes.
- ❖ Coordina la entrega del proyecto.

Eros Perrone – Developer (Diseño y diagramas)

- ❖ Encargada del diseño de diagramas de casos de uso, actividad y ER.
- ❖ Colabora en la creación del prototipo de interfaz.
- ❖ Apoya en la redacción de las secciones técnicas del proyecto.

Titi Lapuerta – Tester y Documentación

- ❖ Encargado de la planificación de métricas y pruebas.
- ❖ Revisa la coherencia de los diagramas con los requerimientos.
- ❖ Redacta y organiza el documento final del proyecto para su entrega.

Plan de trabajo en equipo:

- Se realizan reuniones de planificación y revisión al inicio y fin de cada sprint para controlar el progreso.
- Cada integrante notificará al resto del equipo sobre avances y posibles bloqueos.
- Se utilizará Trello / Google Sheets para asignar tareas específicas a cada integrante y marcar el estado de avance.
- El Scrum Master (Franco) se encargará de facilitar las reuniones, controlar tiempos y asegurar que se cumpla con la planificación.

Punto 7: Seleccionar y describir las metodologías de proceso que se aplicarán (ejemplo: modelos en cascada, iterativo, ágil).

Metodología seleccionada: Metodología Ágil (Scrum)

- Para el desarrollo del Sistema de Gestión de Biblioteca Online, se selecciona la Metodología Ágil, específicamente el marco de trabajo Scrum, ya que permite:

- Adaptarse rápidamente a cambios en los requerimientos.
- Entregar avances funcionales de forma parcial y progresiva.
- Favorecer la comunicación y el trabajo colaborativo.
- Monitorear el progreso de manera continua.

¿Por qué Scrum?

- Permite entregas tempranas y frecuentes de partes del sistema funcional.
- El cliente puede revisar y dar feedback en cada sprint, asegurando que el producto cumpla con las necesidades reales.
- Favorece la detección temprana de errores y corrección rápida.
- Ayuda a gestionar el tiempo de forma eficiente dividiendo el trabajo en sprints de duración fija.
- Fomenta el trabajo en equipo con roles claros.

Descripción de Scrum aplicado al proyecto

Roles definidos:

- **Scrum Master** (Franco): organiza reuniones, remueve impedimentos y asegura que se cumplan los tiempos.
- **Developer** (Eros): se encarga del diseño de diagramas y estructura del sistema.
- **Tester y Documentación** (Titi): planifica métricas y pruebas, redacta el TP.

Artefactos utilizados:

- **Product Backlog**: lista de funcionalidades a desarrollar (casos de uso, diagramas, pruebas, prototipo).
- **Sprint Backlog**: selección de tareas a realizar en cada sprint semanal.
- **Incremento**: avance funcional al finalizar cada sprint (por ejemplo: diagramas listos, métricas definidas).

Eventos en el equipo:

- **Sprint Planning**: reunión al inicio de cada sprint para planificar tareas.
- **Daily Scrum** (informal): breve seguimiento en cada reunión online.
- **Sprint Review**: revisión de los avances al final de cada sprint.
- **Sprint Retrospective**: reflexión sobre qué se hizo bien y qué se puede mejorar.

Duración de sprints en este proyecto:

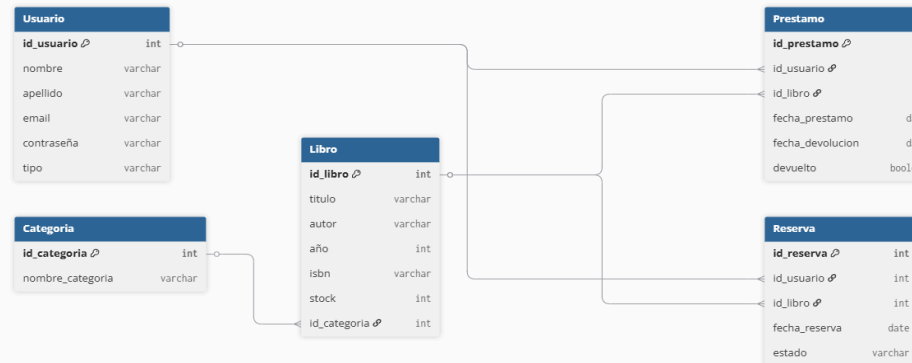
Se trabajará con 5 sprints de 1 semana cada uno, permitiendo cumplir con el cronograma y revisar avances de forma continua.

Conclusión de la metodología seleccionada:

Utilizar Scrum en este proyecto permite trabajar de forma ordenada, adaptable y colaborativa, asegurando entregas parciales de calidad y fomentando la mejora continua, con entregables claros en cada sprint para facilitar la aprobación directa del TP.

Punto 8: Desarrollar diagramas entidad-relación para la base de datos, si fuera aplicable.

```
1 Table Usuario {
2   id_usuario int [pk]
3   nombre varchar
4   apellido varchar
5   email varchar
6   contraseña varchar
7   tipo varchar
8 }
9
10 Table Libro {
11   id_libro int [pk]
12   titulo varchar
13   autor varchar
14   año int
15   isbn varchar
16   stock int
17   id_categoria int [ref: > Categoria.id_categoria]
18 }
19
20 Table Prestamo {
21   id_prestamo int [pk]
22   id_usuario int [ref: > Usuario.id_usuario]
23   id_libro int [ref: > Libro.id_libro]
24   fecha_prestamo date
25   fecha_devolucion date
26   devuelto boolean
27 }
28
29 Table Reserva {
30   id_reserva int [pk]
31   id_usuario int [ref: > Usuario.id_usuario]
32   id_libro int [ref: > Libro.id_libro]
33   fecha_reserva date
34   estado varchar
35 }
36
37 Table Categoria {
38   id_categoria int [pk]
39   nombre_categoria varchar
40 }
```



Punto 9: Definir al menos 3 métricas para medir el avance y la eficiencia del proceso (ejemplo: cantidad de historias de usuario completadas / historias totales, velocidad del equipo por iteración, etc). Explicar cómo se recopilarán y analizarán los datos de esas métricas seleccionadas.

Durante el desarrollo del proyecto aplicamos la metodología ágil Scrum, por lo tanto definimos las siguientes métricas para medir cómo avanza el equipo y cuán eficiente es en cada sprint.

Métrica 1: Porcentaje de historias de usuario completadas

Fórmula: $(\text{historias completadas} / \text{historias totales}) \times 100$

Objetivo: medir el avance general del proyecto sprint a sprint.

Ejemplo: si hay 10 historias y se completaron 8, se logró un 80% de avance.

Cómo se recopila:

- Se anota cada historia de usuario planificada en una hoja de cálculo (Google Sheets o Trello).
- Cada integrante marca como "completada" cuando termina su parte.

Cómo se analiza:

- Al finalizar cada sprint se calcula el porcentaje.
- Si hay bajo avance, se revisan los bloqueos o errores.

Métrica 2: Velocidad del equipo (story points por sprint)

Definición: cantidad de puntos (story points) completados por sprint.

Objetivo: medir la capacidad del equipo para entregar valor semana a semana.

Ejemplo: si en la semana se completaron 12 puntos, esa fue la velocidad.

Cómo se recopila:

- Cada historia tiene un valor estimado en puntos (por dificultad).
- Al final del sprint se suman los puntos completados.

Cómo se analiza:

- Se observa si la velocidad se mantiene o mejora en cada sprint.
- Si baja, se analiza si hubo tareas mal estimadas o interrupciones.

Métrica 3: Tareas retrasadas por sprint

Definición: cantidad de tareas planificadas que no se completaron en el sprint.

Objetivo: identificar cuellos de botella y posibles sobrecargas.

Ejemplo: si había 8 tareas y quedaron 3 sin hacer, hay un 37,5% de retraso.

Cómo se recopila:

- Se compara lo planificado al inicio del sprint con lo que se marcó como hecho.

Cómo se analiza:

- Se revisa si las tareas no completadas se debieron a mala planificación, falta de tiempo o dependencia externa.
- Sirve para ajustar mejor el próximo sprint.

Herramienta usada para recopilar métricas:

- Se usará una hoja de cálculo en Google Sheets con tres columnas:
- Historia o tarea
- Estado (pendiente / en progreso / completada)
- Puntos asignados
- Cada integrante del equipo actualiza su estado y se hacen cálculos automáticos al final de cada semana.

Punto 10: Proponer métricas para evaluar la calidad del producto final (ejemplo: cantidad de errores detectados, cobertura de pruebas, etc).

Para asegurar la calidad del sistema desarrollado, se utilizarán las siguientes métricas, que permiten medir distintos aspectos del producto en su versión final:

1. Cantidad de errores detectados en pruebas

- Se contabiliza cuántos errores aparecen durante la fase de pruebas (unitarias, de integración, de aceptación).
- Un número alto puede indicar baja calidad o necesidad de mejorar las validaciones.
- Un número bajo, junto con buena cobertura de pruebas, indica estabilidad.

Ejemplo: Si durante las pruebas finales se detectan más de 10 errores funcionales, se revisan los módulos involucrados.

2. Cobertura de pruebas

- Mide qué porcentaje del código fue ejecutado durante las pruebas.
- Se busca alcanzar al menos un 80% de cobertura para asegurar que la mayoría de los caminos lógicos fueron verificados.

Ejemplo: Si la cobertura es del 92%, significa que la mayoría del sistema fue puesto a prueba y tiene menor riesgo de fallos ocultos.

3. Tiempo promedio de respuesta del sistema

- Evalúa la rapidez con la que el sistema responde a una acción del usuario (como buscar un libro o reservar un turno).
- Un buen sistema debe responder en menos de 2 segundos para las operaciones comunes.

Ejemplo: Si reservar un turno médico demora más de 3 segundos, se considera que hay que optimizar consultas a la base de datos.

4. Satisfacción del usuario (opcional si lo presentás)

- Se puede realizar una breve encuesta a los usuarios (reales o simulados) luego de usar el sistema.
- Preguntas simples como: ¿fácil de usar? ¿Te pareció útil? ¿Hubo algo confuso?
- Se mide en escala del 1 al 5 o con "Sí / No".

Punto 11: Justificar las métricas elegidas en función de los requisitos del proyecto.

Las métricas seleccionadas fueron definidas en función de los objetivos del sistema y los requerimientos funcionales y no funcionales establecidos en el proyecto. A continuación se explica por qué cada una fue elegida:

1. Cantidad de errores detectados en pruebas

Justificación:

- Esta métrica está directamente relacionada con el requerimiento de que el sistema sea confiable y funcional.
- Permite medir si las funcionalidades como solicitar préstamo, renovar libros o consultar el catálogo funcionan correctamente antes de liberar el sistema.

Está alineada con los objetivos de entregar un producto estable y sin errores graves en producción.

2. Cobertura de pruebas

Justificación:

- Este sistema tiene varios caminos posibles (por ejemplo: libro disponible o no, renovación permitida o rechazada, etc.).
- La cobertura de pruebas permite verificar que todos esos escenarios hayan sido considerados durante las pruebas.

Asegura que los requerimientos funcionales estén bien implementados y testeados.

3. Tiempo promedio de respuesta del sistema

Justificación:

- Uno de los requerimientos no funcionales del sistema es que sea rápido y cómodo para el usuario.
- Medir el tiempo de respuesta ayuda a garantizar una buena experiencia y a detectar posibles cuellos de botella.

Está directamente vinculado con el rendimiento del sistema y la satisfacción del usuario.

4. Satisfacción del usuario (opcional)

Justificación:

- Aunque no es una métrica técnica, medir la experiencia del usuario (por ejemplo con una encuesta) permite saber si el sistema cumple su objetivo de ser accesible, útil y fácil de usar.

Está relacionada con los requerimientos no funcionales de usabilidad y experiencia general.

Punto 12: Sugerir un plan para garantizar la mejora continua basada en los resultados de las métricas.

Para asegurar que el sistema evolucione y mejore con el tiempo, se plantea un plan de mejora continua enfocado en el análisis de las métricas definidas y en la incorporación de aprendizajes en cada ciclo de desarrollo.

1. Revisión periódica de métricas

- Se analizarán los resultados de las métricas (errores detectados, cobertura de pruebas, tiempos de respuesta, satisfacción del usuario) al finalizar cada sprint o iteración.
- Se dejará registro de estos valores en una planilla compartida por el equipo (por ejemplo, en Google Sheets).

Objetivo: detectar tendencias, repeticiones de errores o áreas que requieren atención.

2. Reunión de retrospectiva al finalizar cada sprint

- El equipo realizará una retrospectiva corta al finalizar cada sprint (una vez por semana).
- En esta reunión se discutirán:
 - Qué salió bien.
 - Qué se puede mejorar.
 - Qué acciones tomar para el próximo sprint.

Esto permite mejorar tanto el producto como la forma en que se trabaja en equipo.

3. Revisión y mejora del código o funcionalidades problemáticas

- Si una funcionalidad presenta muchos errores, se priorizará su revisión y posible refactorización.
- Si una sección del sistema tiene baja cobertura de pruebas, se escribirán nuevos casos de prueba.
- Si el tiempo de respuesta supera lo esperado, se analizarán las consultas a base de datos o procesos innecesarios.

Esto permite mantener la calidad técnica del sistema.

4. Registro de decisiones y aprendizajes

- Se documentarán los cambios aplicados a partir de los resultados de las métricas.
- Esto servirá como historial del proyecto y como referencia para futuros desarrollos o nuevas versiones.

Facilita aprender de la experiencia y no repetir errores.

5. Encuestas de satisfacción post-entrega

- Si el sistema se entrega a usuarios reales, se propondrá una encuesta de feedback corto.
-
- Esto permitirá recoger opiniones útiles para futuras mejoras.

La retroalimentación directa del usuario final es clave para evolucionar el sistema