

## **Proyecto Final Desarrollo de Software**



Carlos A. Camacho Castaño – 2160331

Kevin Marín – 2160364

Kevin S. Ramírez - 2259371

Harrison I. Valencia – 2159979

Universidad del Valle – Sede Tuluá

Escuela de Ingeniería de Sistemas y Computación

Ing. Juan Pinillos

Tuluá, Valle del Cauca

Junio de 2024

## **Librería Municipal de Tuluá**

### **1. Fase de Inicio - Ciclo de Vida del Desarrollo de Software:**

Se implementará una metodología ágil por su flexibilidad y adaptabilidad, además de tener en cuenta la entrega rápida requerida para este desarrollo.

La metodología elegida fue Scrum porque es un marco ágil que nos permitirá una mayor flexibilidad, nos ayudará a mejorar la colaboración en el equipo y una rápida entrega como lo exige el proyecto, con la posibilidad de poder avanzar en un futuro en el desarrollo del producto más allá de un MVP por su enfoque iterativo e incremental.

#### **Plan de proyecto inicial:**

Para nuestro proyecto final, desarrollaremos un MVP para la librería municipal de Tuluá, en donde se implementará el módulo de préstamo de libro, contrarrestando la problemática de seguir el mecanismo de planillas actual, las cuales son tediosas de llenar, suelen perderse, son difíciles de almacenar y los bibliotecarios no entienden sus caligrafías entre sí.

Para ello contaremos con un login principal con un Admin predeterminado, creación de usuario, catálogo de libros, gestión de préstamos (solicitud y devolución) y lista de préstamos.

Para ello se estiman 5 Sprints, con duración de una semana cada uno, incluyendo el sprint inicial y el sprint final.

Los roles dentro del equipo serán:

Product Owner: Kevin Ramírez

Scrum Master: Carlos Camacho

Desarrolladores: Kevin Marín, Harrison Valencia.

## 2. Fase de Análisis - Análisis de Requerimientos y UML:

Levantamiento de requerimientos:

### Requerimientos Funcionales:

CÓDIGO	RF-01
TÍTULO	Registro de usuarios
DESCRIPCIÓN	El sistema debe permitir a los usuarios registrarse proporcionando un nombre de usuario, dirección de correo electrónico y contraseña.
PRIORIDAD	Alta
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"><li>• El usuario debe poder ingresar su nombre de usuario, dirección de correo electrónico y contraseña en el formulario de registro.</li><li>• El sistema debe verificar que la dirección de correo electrónico sea única en el sistema.</li><li>• Después de un registro exitoso, el usuario debe recibir un correo electrónico de confirmación.</li></ul>
ENTRADAS	Nombre: String Correo: String Contraseña: String

CÓDIGO	RF-02
TÍTULO	Consultar catálogo de libros
DESCRIPCIÓN	El sistema debe permitir a los usuarios consultar el catálogo de libros.
PRIORIDAD	Media
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"><li>• El usuario debe poder consultar los libros por categoría</li><li>• Todos los libros existentes en la biblioteca deben estar en el catalogo</li></ul>
ENTRADAS	Nombre: String Código: Int Estado: String Categoría: String Autor: String Año de lanzamiento: Int

CÓDIGO	RF-03
TITULO	Actualizar catálogo de libros
DESCRIPCIÓN	El sistema debe permitir al Admin actualizar el catálogo de libros.
PRIORIDAD	Media
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"> <li>• El Admin debe poder agregar un libro</li> <li>• El Admin debe poder eliminar un libro</li> <li>• El Admin debe poder editar un libro</li> </ul>
ENTRADAS	Nombre: String Código: Int Estado: String Categoría: String Autor: String Año de lanzamiento: Int

CÓDIGO	RF-04
TITULO	Solicitud de préstamo
DESCRIPCIÓN	El sistema debe permitir al usuario solicitar el préstamo de un libro.
PRIORIDAD	Alta
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"> <li>• El usuario debe poder solicitar el préstamo de un libro</li> <li>• El usuario debe estar logeado</li> </ul>
ENTRADAS	Nombre: String   Código: Int

CÓDIGO	RF-05
TITULO	Devolver libro
DESCRIPCIÓN	El sistema debe permitir al usuario devolver un libro prestado
PRIORIDAD	Alta
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"> <li>• El usuario debe poder devolver un libro prestado</li> <li>• El usuario debe estar logeado</li> </ul>
ENTRADAS	Nombre: String   Código: Int

CÓDIGO	RF-06
TÍTULO	Gestión de prestamos
DESCRIPCIÓN	El sistema debe permitir al Admin gestionar los préstamos de libros
PRIORIDAD	Alta
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"> <li>• El Admin debe poder registrar un préstamo</li> <li>• El Admin debe poder registrar una devolución</li> <li>• El sistema debe notificar si un libro no se encuentra disponible para préstamo</li> </ul>
ENTRADAS	Nombre: String Código: Int Nombre usuario: String Fecha: String

CÓDIGO	RF-07
TÍTULO	Lista de prestamos
DESCRIPCIÓN	El sistema debe permitirle al Admin visualizar una lista de préstamos activos
PRIORIDAD	Media
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"> <li>• El Admin debe consultar los prestamos activos</li> <li>• Todos los prestamos activos deben estar en la lista</li> <li>• Cuando un préstamo finaliza, es excluido de la lista</li> </ul>
ENTRADAS	Nombre: String Código: Int Nombre Usuario: String

**Requerimientos no funcionales:**

CÓDIGO	RNF - 01
TITULO	Seguridad de los datos
DESCRIPCIÓN	El sistema debe garantizar la seguridad de los datos del usuario mediante el cifrado de la información confidencial y el uso de técnicas de autenticación segura.
PRIORIDAD	Media
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"><li>• Todas las contraseñas de los usuarios deben almacenarse de forma segura utilizando algoritmos de hash.</li><li>• Las comunicaciones entre el cliente y el servidor deben estar cifradas utilizando el protocolo HTTPS.</li><li>• El sistema debe tener políticas de acceso y control de permisos para proteger los datos del usuario contra accesos no autorizados.</li></ul>

CÓDIGO	RNF - 02
TITULO	Identidad de la biblioteca
DESCRIPCIÓN	El sistema debe estar ambientado con los colores municipales
PRIORIDAD	Media
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"><li>• El sistemas debe contener en su paleta de colores principal el color verde, rojo y blanco</li><li>• El sistema debe contener como logo el escudo de Tuluá</li></ul>

CÓDIGO	RNF - 03
TITULO	Interfaz intuitiva
DESCRIPCIÓN	El sistema debe estar intuitivo y fácil de usar.
PRIORIDAD	Alta
CRITERIOS DE ACEPTACION	<ul style="list-style-type: none"><li>• El sistema no debe contener funcionalidades que no se empleen</li><li>• El sistema debe ser conciso y directo</li><li>• El sistema debe visualmente debe basarse en botones</li></ul>

Diagrama de casos de uso:

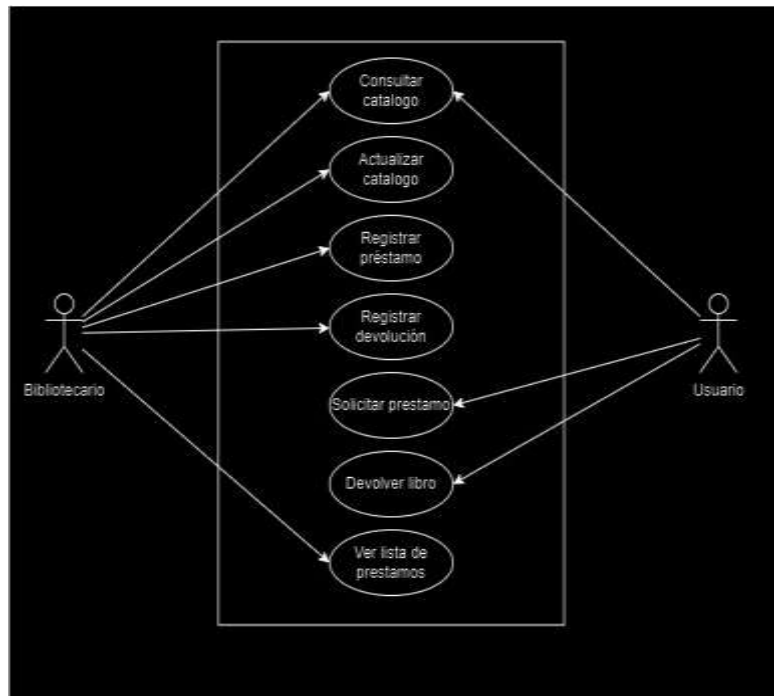
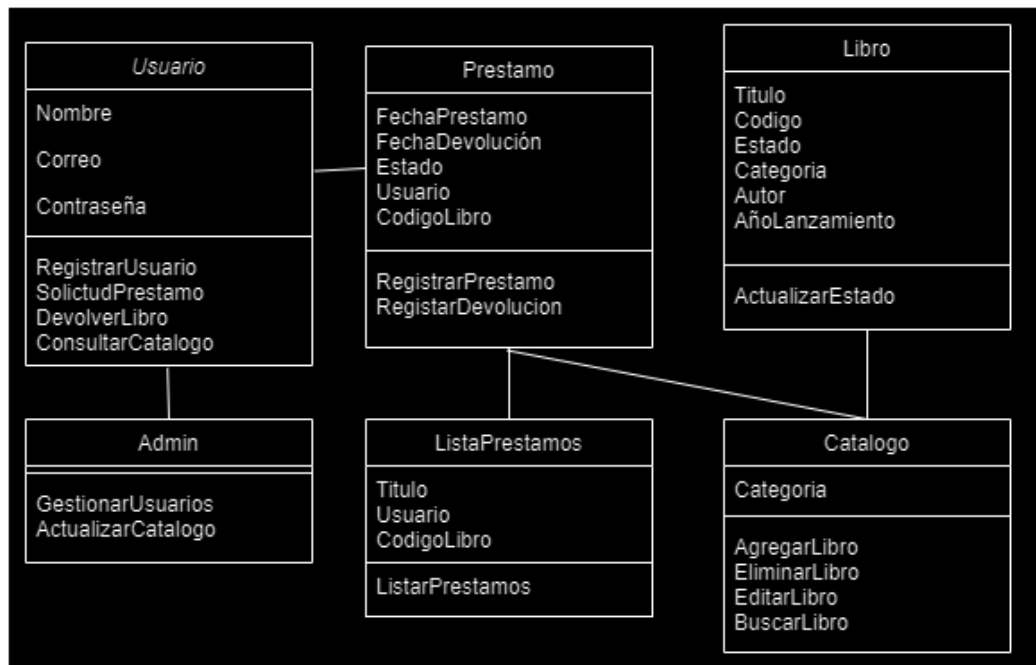


Diagrama de clases:



### 3. Fase de Diseño - Enfoque Orientado a Objetos y UML:

Diagrama de clases (Refinado):

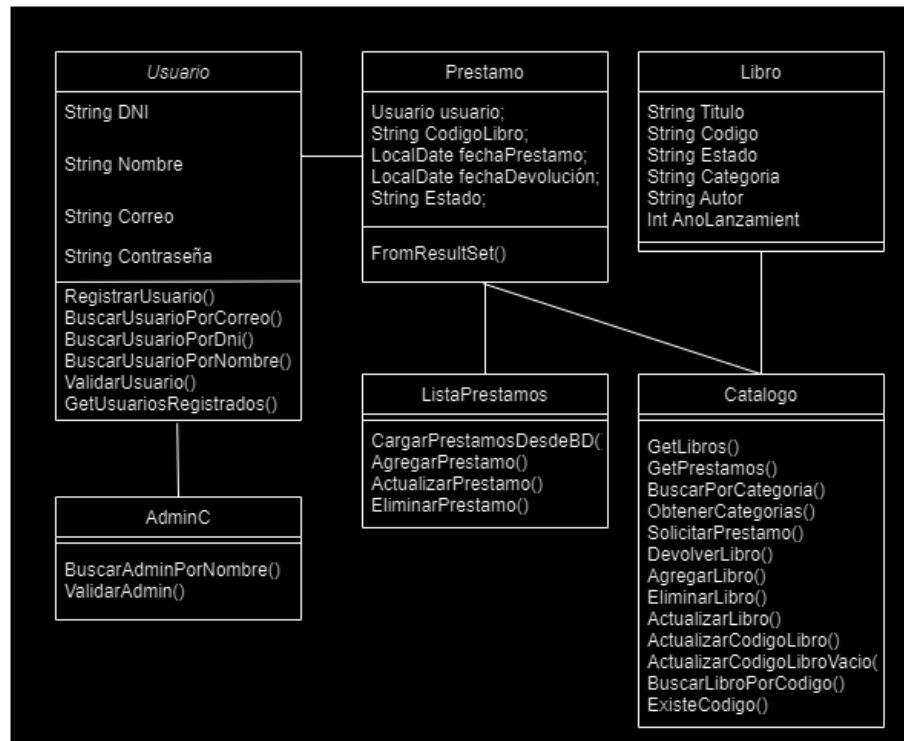


Diagrama de secuencia:

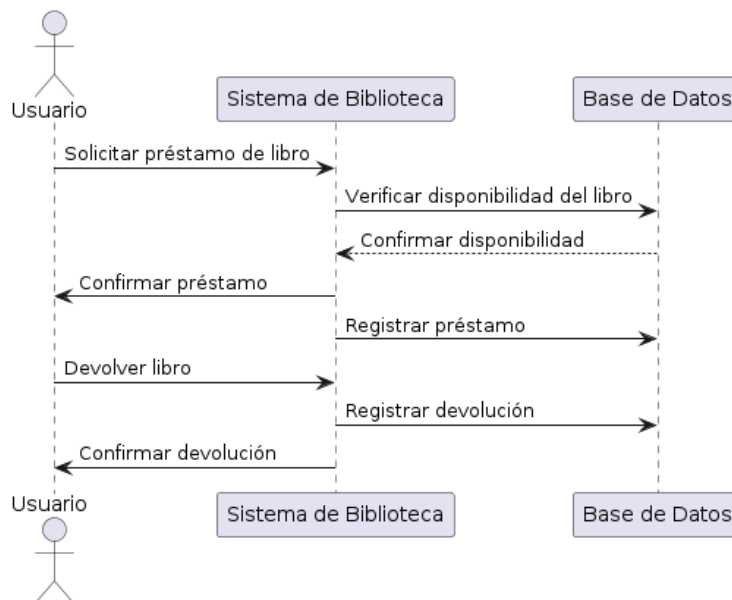




Diagrama de despliegue (App local):

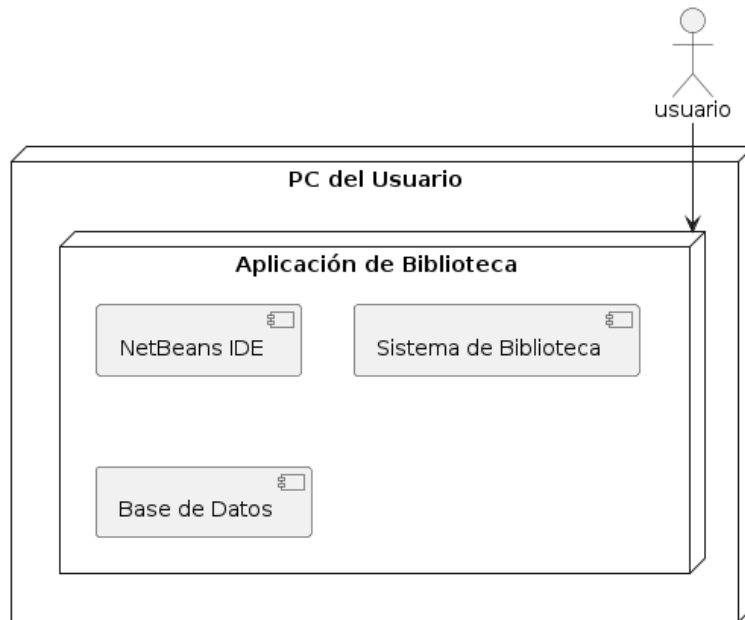


Diagrama de actividades:

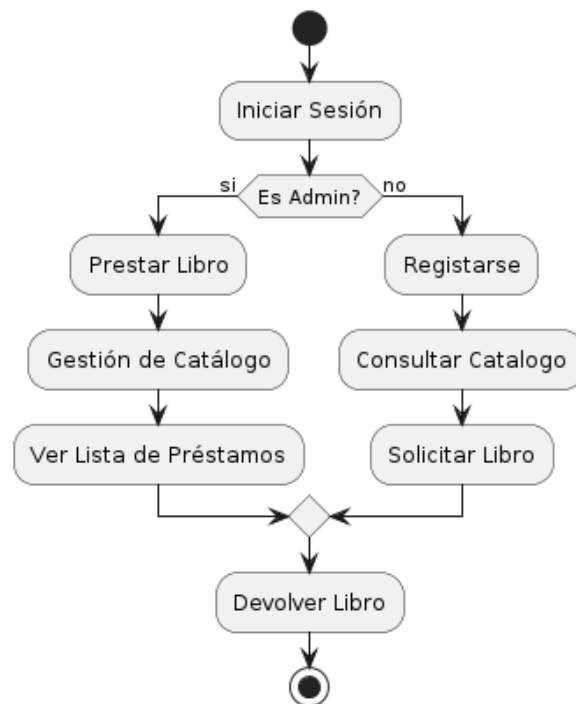
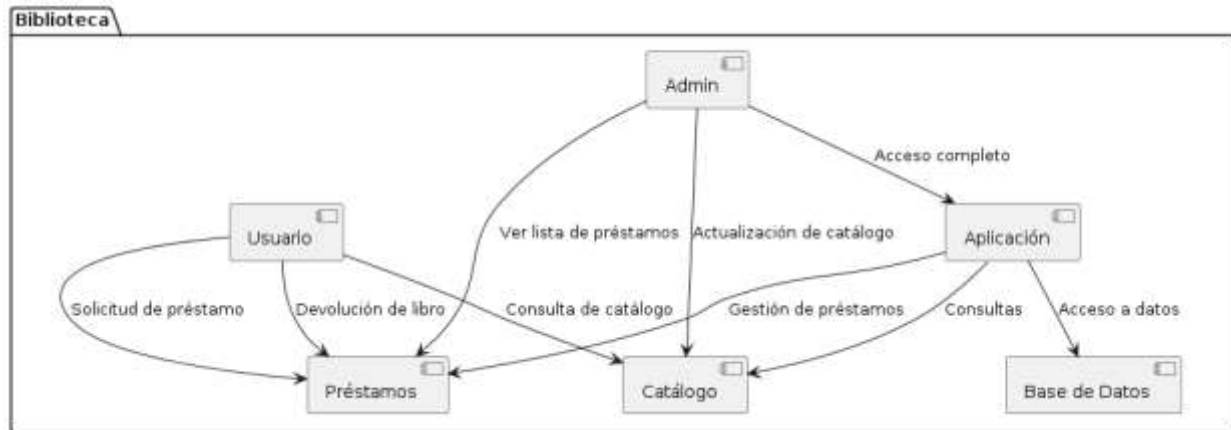


Diagrama de componentes:



#### 4. Fase de Construcción - Metodología RUP y Desarrollo Orientado a Objetos:

Se desarrolla el MVP en la herramienta Apache Netbeans en el lenguaje de programación Java por su enfoque orientado a objetos y todas las posibilidades de extensión que nos brinda con sus diferentes Frameworks.

Ya contamos con unos requisitos funcionales y no funcionales, un diseño por medio de diagramas UML, con esto se define el Product Backlog y es hora de empezar la construcción de nuestro software que se llevará a cabo en 5 sprints, de una semana cada uno, de la siguiente manera:

Vamos a establecer un calendario de reuniones y asignar responsabilidades específicas a cada miembro del equipo, asegurando que el proyecto se lleve a cabo de manera eficiente y organizada.

#### Reuniones Semanales

##### Scrum Meetings:

- **Daily Stand-up (Lunes a Viernes): 10:00 AM - 10:15 AM**
  - Breve reunión diaria para discutir lo que se hizo ayer, lo que se hará hoy y cualquier impedimento.
- **Sprint Planning (Inicio de Sprint, Lunes): 9:00 AM - 10:00 AM**
  - Planificación de las tareas del Sprint, establecimiento de metas y distribución de trabajo.
- **Sprint Review (Final de Sprint, Viernes): 3:00 PM - 4:00 PM**
  - Demostración de la funcionalidad desarrollada durante el Sprint y recogida de feedback.
- **Sprint Retrospective (Final de Sprint, Viernes): 4:00 PM - 4:30 PM**
  - Reflexión sobre lo que salió bien, lo que se puede mejorar y cómo implementar mejoras en el próximo Sprint.

## Reuniones Adicionales:

- **Backlog Refinement (Miércoles): 2:00 PM - 3:00 PM**
  - Revisión y actualización del Product Backlog, asegurando que las historias de usuario estén claras y priorizadas.

## Asignación de Responsabilidades

### Product Owner: Kevin Ramírez

- Definir y priorizar el Product Backlog.
- Asegurar que el equipo entiende claramente las historias de usuario y los criterios de aceptación.
- Aceptar/rechazar el trabajo completado.

### Scrum Master: Carlos Camacho

- Facilitar las reuniones Scrum.
- Eliminar impedimentos que bloqueen el progreso del equipo.
- Asegurar que Scrum se entienda y se implemente correctamente.

### Desarrolladores: Kevin Marín, Harrison Valencia

- Desarrollar el código según las historias de usuario priorizadas.
- Colaborar en la definición de tareas durante el Sprint Planning.
- Participar en pruebas de integración y asegurar la calidad del código.

Día	Hora	Actividad	Participantes
Lunes	09:00 - 10:00 AM	Sprint Planning	Todos
Lunes	10:00 - 10:15 AM	Daily Stand-up	Todos
Martes	10:00 - 10:15 AM	Daily Stand-up	Todos
Miércoles	10:00 - 10:15 AM	Daily Stand-up	Todos
Miércoles	02:00 - 03:00 PM	Backlog Refinement	Kevin R., Carlos
Jueves	10:00 - 10:15 AM	Daily Stand-up	Todos
Viernes	10:00 - 10:15 AM	Daily Stand-up	Todos
Viernes	03:00 - 04:00 PM	Sprint Review	Todos
Viernes	04:00 - 04:30 PM	Sprint Retrospective	Todos

## **Sprint 1: Configuración Inicial y Registro de Usuarios**

### **Objetivos del Sprint:**

- Configuración del entorno de desarrollo.
- Implementación del módulo de registro de usuarios.

### **Actividades:**

1. **Configuración del Entorno:**
  - Configurar el entorno de desarrollo (NetBeans, Repositorio GitHub.).
  - Configurar el servidor y base de datos (PostgreSQL).
2. **Implementación del Registro de Usuarios:**
  - Crear la estructura de la base de datos para usuarios.
  - Desarrollar la interfaz de registro de usuarios (formulario de registro).
  - Implementar la lógica de negocio para el registro (validación, almacenamiento de datos, etc.).
  - Enviar correos electrónicos de confirmación tras el registro exitoso.

### **Entregables:**

- Código funcional para el registro de usuarios.
- Base de datos configurada y conectada.
- Entorno de desarrollo listo.

### **Metodología RUP:**

- **Requerimientos:** Refinar historias de usuario para el registro de usuarios.
- **Análisis y Diseño:** Diseñar la base de datos y la interfaz de registro.
- **Implementación:** Configurar el entorno de desarrollo y la base de datos, implementar la funcionalidad de registro.
- **Pruebas:** Crear y ejecutar pruebas unitarias para el registro de usuarios.

## **Sprint 2: Autenticación y Catálogo de Libros**

### **Objetivos del Sprint:**

- Implementar la funcionalidad de inicio de sesión y autenticación.
- Desarrollar la funcionalidad para consultar el catálogo de libros.

### **Actividades:**

1. **Autenticación:**
  - Crear la funcionalidad de inicio de sesión.
  - Implementar la autenticación de usuarios (verificación de credenciales).
2. **Catálogo de Libros:**
  - Crear la estructura de la base de datos para libros.

- Desarrollar la interfaz de consulta del catálogo.
- Implementar la lógica para consultar el catálogo (búsqueda y filtrado de libros).

#### **Entregables:**

- Funcionalidad completa de inicio de sesión y autenticación.
- Interfaz funcional para consultar el catálogo de libros.
- Base de datos actualizada con la estructura de libros.

#### **Metodología RUP:**

- **Requerimientos:** Refinar historias de usuario para autenticación y catálogo.
- **Análisis y Diseño:** Diseñar la interfaz de inicio de sesión y consulta del catálogo.
- **Implementación:** Desarrollar la autenticación y la funcionalidad de consulta del catálogo.
- **Pruebas:** Realizar pruebas de integración y aceptación para las nuevas funcionalidades.

### **Sprint 3: Gestión de Libros por el Admin**

#### **Objetivos del Sprint:**

- Implementar la funcionalidad para que el Admin gestione el catálogo de libros (agregar, editar, eliminar libros).

#### **Actividades:**

1. **Gestión de Libros:**
  - Crear la interfaz de gestión de libros para el Admin.
  - Implementar la lógica para agregar libros al catálogo.
  - Implementar la lógica para editar información de los libros.
  - Implementar la lógica para eliminar libros del catálogo.

#### **Entregables:**

- Funcionalidad completa para que el Admin gestione el catálogo de libros.
- Interfaz funcional de gestión de libros.

#### **Metodología RUP:**

- **Requerimientos:** Refinar historias de usuario para la gestión de libros.
- **Análisis y Diseño:** Diseñar la interfaz y la lógica de gestión de libros.
- **Implementación:** Implementar la funcionalidad para agregar, editar y eliminar libros.
- **Pruebas:** Realizar pruebas unitarias y de integración para la gestión de libros.

### **Sprint 4: Solicitud y Devolución de Préstamos**

### Objetivos del Sprint:

- Implementar la funcionalidad para que los usuarios soliciten préstamos de libros.
- Implementar la funcionalidad para que los usuarios devuelvan libros prestados.

### Actividades:

1. **Solicitud de Préstamos:**
  - Crear la interfaz para solicitar préstamos.
  - Implementar la lógica para gestionar solicitudes de préstamos.
2. **Devolución de Libros:**
  - Crear la interfaz para devolver libros.
  - Implementar la lógica para gestionar devoluciones de libros.

### Entregables:

- Funcionalidad completa para solicitar préstamos.
- Funcionalidad completa para devolver libros prestados.

### Metodología RUP:

- **Requerimientos:** Refinar historias de usuario para la solicitud y devolución de préstamos.
- **Análisis y Diseño:** Diseñar la interfaz y la lógica de préstamos y devoluciones.
- **Implementación:** Desarrollar la funcionalidad de solicitud y devolución de préstamos.

## Sprint 5: Gestión de Préstamos y Pruebas Finales

### Objetivos del Sprint:

- Implementar la funcionalidad para que el Admin gestione los préstamos de libros.
- Realizar pruebas finales y ajustes basados en feedback.

### Actividades:

1. **Gestión de Préstamos:**
  - Crear la interfaz para que el Admin gestione los préstamos.
  - Implementar la lógica para registrar préstamos y devoluciones.
  - Notificar disponibilidad de libros para préstamos.
2. **Pruebas Finales:**
  - Realizar pruebas de integración y aceptación.
  - Recopilar feedback de usuarios y realizar ajustes necesarios.
  - Preparar la documentación técnica y manual de usuario.

### Entregables:

- Funcionalidad completa de gestión de préstamos por el Admin.

- Pruebas de integración y aceptación realizadas.
- Documentación técnica y manual de usuario completados.

#### **Metodología RUP:**

- **Requerimientos:** Refinar historias de usuario para la gestión de préstamos.
- **Análisis y Diseño:** Diseñar la interfaz y la lógica de gestión de préstamos.
- **Implementación:** Implementar la funcionalidad de gestión de préstamos y realizar ajustes finales.
- **Pruebas:** Realizar pruebas de integración y aceptación, y recopilar feedback para ajustes finales.

#### **Revisiones:**

- Al final de cada Sprint, se realizó una revisión de Sprint (Sprint Review) para demostrar la funcionalidad completada y recoger feedback.
- Se hicieron retrospectivas de Sprint (Sprint Retrospective) para discutir lo que funcionó bien, lo que no y cómo mejorar en el próximo Sprint.

### **5. Fase de Transición - Metodologías de Desarrollo de Software y RUP:**

#### **Flexibilidad de Scrum y RUP:**

- **Scrum:** Nos permite adaptarnos rápidamente a los cambios gracias a sus iteraciones cortas (sprints) y reuniones regulares (Daily Stand-ups, Sprint Reviews, y Retrospectivas). Esto facilita la incorporación de feedback continuo y la adaptación a los cambios en los requisitos.
- **RUP:** A través del enfoque iterativo e incremental, RUP también permitió la adaptación continua.

#### **Gestión de Cambios:**

- **Backlog Refinement:** En Scrum, el backlog se refina continuamente para incorporar cambios y prioridades nuevas, sobre la marcha surgieron novedades en el software que pudieron ser resueltas en el siguiente sprint. El Product Owner (Kevin Ramirez) juega un papel clave en la gestión y priorización de estos cambios.

#### **Iteraciones y Revisiones:**

- **Sprint Reviews:** Nos permitió ajustar los requisitos y las prioridades a medida que íbamos desarrollando el software, además de ver falencias para tratar de corregirlas en el siguiente sprint.

La transición inicia de la siguiente manera:

### **1. Pruebas:**

Se realizan pruebas unitarias, de integración y de funcionalidad.

Se realizan pruebas de aceptación de usuario, sin informes de defectos.

### **2. Documentación:**

Se realiza la documentación pertinente.

Documentación técnica:

- Se realiza un informe de documentación técnica con los aspectos más importantes del código fuente y se anexa en el repositorio.
- Se realizan comentarios pertinentes en el código.

Documentación de usuario:

- Se realiza un manual de usuario y se anexa en el repositorio.
- Se realiza una guía de instalación en el README.md del repositorio.

Documentación de pruebas:

- Se realiza un informe de pruebas funcionales y de integración y se anexa en el repositorio.

### **3. Despliegue:**

Preparación para el Despliegue del MVP

Actividades de Despliegue:

- Preparación del Entorno de Producción:
  - Configurar Netbeans y PostgreSQL
  - Asegurar la disponibilidad de todos los recursos necesarios.
- Despliegue del Software:
  - Clonar los últimos cambios del GitHub.
  - Ejecutar scripts de despliegue y configuraciones.
- Pruebas Post-Despliegue:
  - Se realizan pruebas para asegurarse de que el sistema funciona correctamente en el entorno de producción.
- Formación del Usuario:
  - Se capacitan a los usuarios finales y al personal de soporte.
  - Se proveen manuales y guías de usuario.
- Monitoreo Post-Despliegue:
  - Se monitorea el sistema para detectar y resolver cualquier problema que surja.
  - Se recolecta el feedback de los usuarios para realizar ajustes finales.



#### **4. Recopilación de Feedback y Ajustes Finales**

Recopilación de Feedback:

- Encuestas y Entrevistas: Se realizan encuestas y entrevistas con los usuarios finales para recoger sus opiniones sobre el sistema.
- Revisión de Comentarios: Se analizan los comentarios y sugerencias recibidas durante las pruebas de aceptación del usuario.
- Reuniones de Feedback: Se agendan reuniones con las partes interesadas para discutir el rendimiento del sistema y posibles mejoras.

#### **5. Ajustes Finales:**

- Corrección de Defectos: Se solucionan defectos o problemas identificados durante el despliegue y las pruebas post-despliegue.
- Mejoras de Usabilidad: Se realizan mejoras basadas en el feedback de los usuarios para hacer el sistema más intuitivo y fácil de usar.
- Actualización de Documentación: Se actualiza la documentación con los cambios realizados.

De esta manera se ve concluido nuestro desarrollo. Se continua con el soporte y mantenimiento.