

Semana 3

Ingeniería de Software (PRY3211)

Formato de respuesta

|  |  |
| --- | --- |
| **Nombre estudiante:** | |
| **Asignatura:** | **Carrera:** |
| **Profesor:** | **Fecha:** |

# Descripción de la actividad

En esta semana realizarás la actividad sumativa grupal con encargo de la Experiencia 1, llamada "Definiendo la visión del producto inicial bajo una propuesta de planificación ágil", la cual estará dividida en dos partes. En la primera parte, entregaran de manera grupal la propuesta del tipo de software a desarrollar con un documento Project Burndown basado en la lista de Requisitos del Sistema y aportando un Diagrama de Caso de Usos para representar la vista de escenario principal. Además, deberán aportar una Planificación ágil basada en Scrum basado en una Planilla Product Backlog con principales épicas e historias de usuarios y sus sprint, aportando una Planilla de Planificación del tiempo RoadMap distribuidas en semanas y la implementación de la herramienta Trello para la organización de principales tareas y entregables.

En la segunda parte, realizarán una presentación a través de un video de no más de 7 minutos, usando la herramienta de Teams.

En la tercera parte y de manera individual, deberás escribir una reflexión personal sobre los aprendizajes obtenidos en la Experiencia 1; cómo estos impactarán en tu desarrollo personal e influirán en el ámbito profesional y consecución de tus metas.

**Documento Proyect Burndown (versión 2)**

Sistema [*Sistema de Gestión para Biblioteca Comunitaria*]

Descripción de la metodología de trabajo (Scrum)

Versión 1.0

Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autores** |
| 00/00/2023 | 1.0 | Primera versión definición de la Visión del Proyecto con los apartados y contenidos asociados |  |
|  |  |  |  |
|  |  |  |  |

Tabla de Contenidos

Contenido

[1. Introducción 3](#_Toc1700174074)

[1.1.1 Propósito de este documento 4](#_Toc1242438244)

[1.1.2 Problemática que resolver 4](#_Toc2023862465)

[1.1.3 Objetivo del Proyecto 4](#_Toc1858285031)

[1.1.4 Alcances 5](#_Toc1946355572)

[2. Descripción General de la Metodología ágil a adoptar. 5](#_Toc1950648504)

[2.1. Fundamentación 6](#_Toc431988535)

[2.2. Valores de trabajo 6](#_Toc447856862)

[3. Personas y roles del proyecto. 6](#_Toc265027249)

[4. Product Backlog. Lista de Componentes y artefactos para Construir. 7](#_Toc67129394)

[4.1 Épicas e historias de usuarios. 8](#_Toc1571379472)

[5. Definición del Done 9](#_Toc1899544251)

[5. Definición tecnologías de Desarrollo a utilizar 9](#_Toc42934262)

[7. Descripción de herramientas de gestión en modalidad Scrum. 11](#_Toc442726364)

# Introducción

Este documento describe la implementación de la metodología de trabajo Scrum para el desarrollo del proyecto **"ComunidadLibro"** destinada a digitalizar y optimizar la gestión de bibliotecas comunitarias, facilitando el proceso de préstamos, devoluciones y fomento de la lectura en comunidades locales. El sistema busca resolver las deficiencias en la gestión manual tradicional de bibliotecas comunitarias, proporcionando herramientas tecnológicas que mejoren tanto la experiencia de los bibliotecarios como de los lectores de la comunidad.

## **Propósito de este documento**

El propósito de este documento es poder facilitar la información de referencia necesaria a las personas involucradas en el desarrollo del Sistema ComunidadLibro.

Se dará a conocer cómo se ejecutará el desarrollo del producto de software en un ciclo de vida adaptativo e integrativo basado en la metodología Scrum. También se presentarán las Épicas y sus historias de usuarios, los componentes y artefactos a construir durante las 8 semanas de desarrollo distribuidos en 4 sprints de 2 semanas cada uno.

Además, se presentarán los roles del equipo Scrum y sus principales competencias técnicas que deben poseer y sus funciones específicas en el proyecto ComunidadLibro. Este documento servirá como guía de referencia para todos los stakeholders, desde el Product Owner hasta los desarrolladores, estableciendo las bases para una implementación exitosa del sistema.

## **Problemática que resolver**

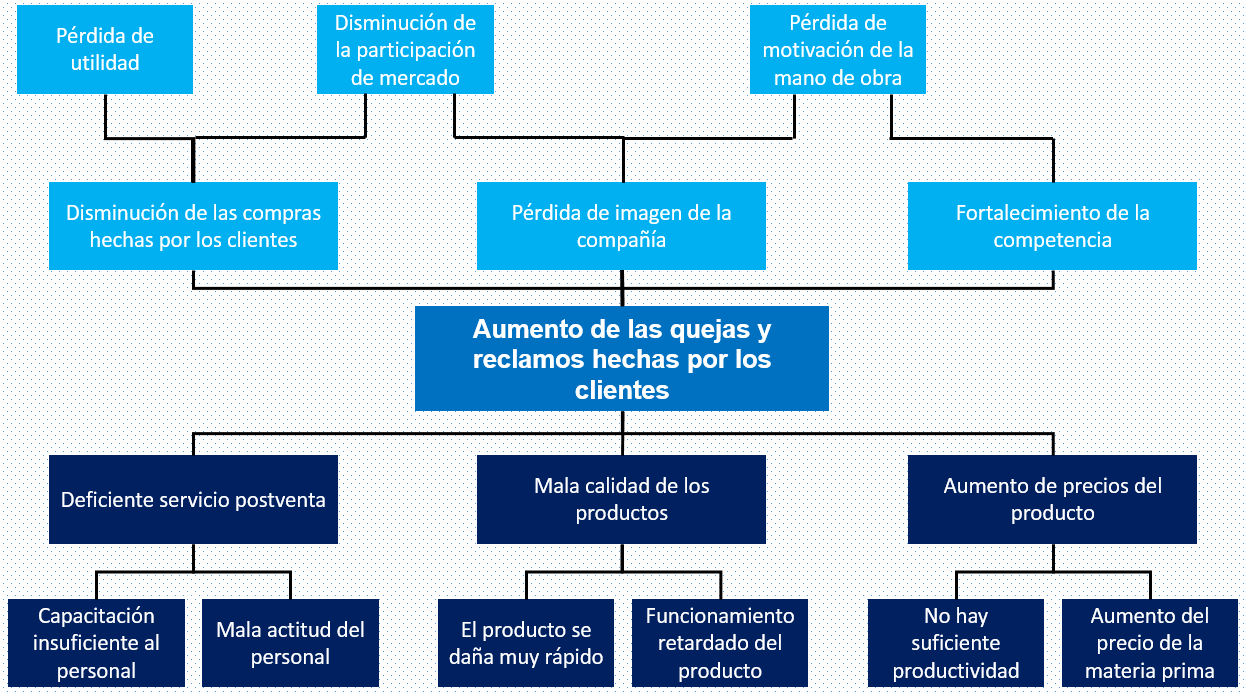
Las bibliotecas comunitarias enfrentan actualmente serios desafíos en la gestión de sus recursos y servicios debido a la dependencia de procesos manuales obsoletos. La mayoría de estas instituciones mantienen registros físicos en cuadernos o fichas de papel para controlar los préstamos de libros, lo que genera múltiples problemas operativos. Los bibliotecarios pierden tiempo considerable en tareas administrativas repetitivas como buscar manualmente la disponibilidad de libros, registrar préstamos a mano y hacer seguimiento de devoluciones pendientes. Esta situación se agrava cuando no existe un sistema centralizado que permita conocer el historial de lecturas de los usuarios o generar informes sobre el uso de la biblioteca, limitando la capacidad de tomar decisiones informadas para mejorar los servicios.

Por otro lado, los lectores de la comunidad experimentan frustración al no poder conocer fácilmente qué libros están disponibles, no tener acceso a recomendaciones personalizadas, y carecer de un historial digital de sus lecturas previas. La falta de herramientas tecnológicas también impide que las bibliotecas comunitarias implementen estrategias efectivas de fomento a la lectura, como rankings de libros más populares, sistemas de reseñas, o la organización de eventos literarios. Esta problemática resulta en una subutilización de los recursos bibliotecarios, menor participación comunitaria en actividades de lectura, y una experiencia deficiente tanto para bibliotecarios como para usuarios. La implementación del sistema ComunidadLibro busca transformar esta realidad mediante la digitalización de procesos y la creación de herramientas que potencian el papel de las bibliotecas como centros de promoción cultural y educativa en sus comunidades.

Ejemplo:

**Figura 1:**

*Mapa del Problema*



Nota: Ejemplo de un mapa de problemas. Betancourt, D. (2016). *Cómo hacer un árbol de problemas: Ejemplo práctico*. Recuperado el 26 de mayo de 2023, de Ingenio Empresa: [www.ingenioempresa.com/arbol-de-problemas](http://www.ingenioempresa.com/arbol-de-problemas)

## **Objetivo del Proyecto**

Desarrollar un sistema web de gestión bibliotecaria denominado "ComunidadLibro" para bibliotecas comunitarias que automatice el proceso de catalogación de libros, optimice la gestión de préstamos y devoluciones, y fomente la participación activa de la comunidad en actividades de lectura mediante herramientas digitales intuitivas que mejoren la experiencia tanto de bibliotecarios como de usuarios lectores.

## **Alcances**

Alcances del Sistema (Lo que SÍ incluye):

Funcionalidades principales:

Sistema completo de gestión de catálogo de libros (registro, edición, búsqueda, eliminación)

Módulo de gestión de usuarios lectores con perfiles personalizados

Sistema automatizado de préstamos y devoluciones con control de fechas.

Panel administrativo para bibliotecarios con informes básicos

Herramientas de fomento a la lectura (recomendaciones, rankings, reseñas)

Sistema de autenticación y autorización por roles

Interfaz web responsiva compatible con dispositivos móviles y de escritorio

Base de datos relacionales para almacenamiento seguro de información.

Alcances Técnicos:

Aplicación web desarrollada con tecnologías estándar (HTML5, CSS3, JavaScript, Python/PHP)

Base de datos MySQL para gestión de información.

Interfaz de usuario intuitivo y accesible

Sistema de respaldo básico de datos

Documentación técnica y de usuario

Alcances de Implementación:

Desarrollo completo en 8 semanas (4 sprints de 2 semanas)

Capacitación básica al personal bibliotecario.

Manual de usuario y administrador

Soporte técnico durante el primer mes post-implementación

Limitaciones y Restricciones del Sistema:

Limitaciones funcionales:

No incluye sistema de pagos en línea para multas o servicios adicionales

No incluye integración con sistemas bibliotecarios externos (MARC, Dublin Core)

No incluye aplicación móvil nativa (solo web responsive)

No incluye sistema de reservas anticipadas de libros.

No incluye módulo de adquisiciones o compras de nuevos libros.

No incluye sistema de códigos de barras o QR para identificación automática

No incluye notificaciones automáticas por correo electrónico o SMS

Limitaciones técnicas:

Sistema monousuario por sesión (no permite múltiples sesiones simultáneas del mismo usuario)

Capacidad máxima recomendada: 1000 libros y 500 usuarios registrados

No incluye integración con redes sociales.

No incluye sistema de chat o mensajería interna.

No incluye funcionalidades offline o sincronización.

Restricciones de implementación:

Requiere conexión a internet estable para funcionar

Necesita servidor web con capacidades mínimas específicas

Bibliotecario personal debe tener conocimientos básicos de computación

No incluye migración automática de datos desde sistemas existentes.

No incluye personalización avanzada de la interfaz de biblioteca

Soporte técnico limitado a horario laboral estándar

Restricciones de seguridad:

No incluye cifrado avanzado de datos personales

No incluye auditoría completa de accesorios y modificaciones.

No incluye sistema de copia de seguridad automática en la nube

No incluye recuperación ante desastres automatizados

Restricciones presupuestarias:

Desarrollo limitado a funcionalidades esenciales definidas en el Product Backlog

No incluye licencias de software premium o herramientas especializadas.

No incluye hosting o infraestructura de producción.

No incluye mantenimiento a largo plazo (más de 6 meses)

Restricciones de escalabilidad:

Diseñado para bibliotecas comunitarias pequeñas a medianas.

No incluye arquitectura distribuida o microservicios.

No incluye balanceadores de carga o alta disponibilidad.

No incluye optimización para grandes volúmenes de datos (más de 10.000 registros)

Nota Importante: Estas limitaciones están definidas para la versión 1.0 del sistema ComunidadLibro. Funcionalidades adicionales podrán ser consideradas en futuras versiones según las necesidades específicas de cada biblioteca comunitaria y la disponibilidad de recursos para desarrollo continuo.

# Descripción General de la Metodología ágil a adoptar.

## **Fundamentación**

Para el desarrollo del proyecto **ComunidadLibro**, se ha seleccionado un **ciclo de vida iterativo e incremental** basado en la metodología **Scrum** , debido a las características específicas del proyecto y las necesidades de las bibliotecas comunitarias.

**Justificación del Ciclo de Vida Seleccionado:**

El proyecto ComunidadLibro requiere un enfoque adaptativo que permita validación temprana y continua con los usuarios finales (bibliotecarios y lectores). A diferencia de un ciclo de vida en cascada, donde los requisitos se definen completamente al inicio, el ciclo iterativo permite:

* **Entrega temprana de valor:** Cada sprint de 2 semanas entregará funcionalidades utilizables
* **Retroalimentación constante:** Los bibliotecarios podrán probar y validar cada incremento
* **Adaptabilidad:** Posibilidad de ajustar requisitos según necesidades emergentes.
* **Reducción de riesgos:** Detección temprana de problemas técnicos o de usabilidad

**Adaptación de Scrum al Proyecto:**

La implementación de Scrum en ComunidadLibro se estructura de la siguiente manera:

* **Sprints de 2 semanas:** Duración óptima para entregar funcionalidades completas
* **Product Backlog dinámico:** Lista priorizada de historias de usuario que puede evolucionar
* **Sprint Planning:** Planificación colaborativa al inicio de cada sprint
* **Daily Scrum:** Reuniones diarias de 15 minutos para sincronización del equipo
* **Sprint Review:** Demostración de funcionalidades al Product Owner y stakeholders
* **Sprint Retrospective:** Mejora continua del proceso de trabajo

## **Valores de trabajo**

Competencias Técnicas Requeridas:

Para el Equipo de Desarrollo:

Desarrollo Frontend: HTML5, CSS3, JavaScript, Bootstrap, diseño responsive

Desarrollo Backend: Python (Flask) o PHP, manejo de APIs REST

Base de Datos: MySQL, diseño de esquemas relacionales, consultas SQL

Control de Versiones: Git, GitHub, manejo de ramas y solicitudes de fusión

Testing: Pruebas unitarias, pruebas de integración, depuración

Seguridad Web: Validación de formularios, prevención de inyecciones SQL

UX/UI Básico: Principios de usabilidad, accesibilidad web

Para el Scrum Master:

Facilitación: Técnicas de moderación de reuniones y resolución de conflictos

Metodologías Ágiles: Conocimiento profundo de Scrum, Kanban, Lean

Herramientas de Gestión: Trello, Jira, GitHub Projects

Coaching: Habilidades para guiar al equipo en prácticas ágiles

Para el propietario del producto:

Análisis de Negocio: Compresión de procesos bibliotecarios

Gestión de Requisitos: Técnicas de elicitación y priorización

Comunicación con Stakeholders: Capacidad de traducir necesidades técnicas

Habilidades Bladas para Trabajo Ágil:

Comunicación efectiva:

Capacidad de expresar ideas técnicas de manera clara

Escucha activa durante ceremonias Scrum

Comentarios constructivos y oportunos

Transparencia en reportar impedimentos y avances

Colaboración y Trabajo en Equipo:

Mentalidad de equipo por encima de logros individuales

Disposición para ayudar a compañeros en dificultades

Participación activa en decisiones técnicas grupales.

Respeto por diferentes perspectivas y enfoques

Adaptabilidad y flexibilidad:

Apertura al cambio de requisitos y prioridades

Capacidad de aprender nuevas tecnologías rápidamente

Resiliencia ante obstáculos técnicos

Mentalidad de mejora continua

Autoorganización y Responsabilidad:

Capacidad de gestionar tiempo y tareas de manera autónoma

Compromiso con los objetivos del sprint

Proactividad en identificar y resolver problemas

Responsabilidad compartida por el éxito del proyecto.

# Personas y roles del proyecto.

|  |  |  |
| --- | --- | --- |
| **Persona** | **Rol** | **Función** |
|  | Stakeholder/s |  |
|  | Product Owner |  |
|  | Scrum Master |  |
|  | Developer 1 |  |
|  | Developer 2 |  |
|  | Developer 3 |  |

# Product Backlog. Lista de Componentes y artefactos para Construir.

## **Épicas e historias de usuarios.**

* Tabla resumen con Principales Épicas e Historias de Usuarios v/s Componentes.

[Insertar Tabla Product Backlog ampliada a Sprintbacklog y describir]

**(Insertar link de archivo planilla ubicado en su repositorio)**

**Figura 2:**

*Ejemplo de planilla de Product Backlog ampliada a Sprintbacklog*

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente  
*Nota:* La planilla muestra los diferentes elementos que el usuario debe completar, entre ellos, las Historias de usuario, artefactos, componentes PTS por historia, Esfuerzo, duración, iteración, prioridad y estado.

# Definición tecnologías de Desarrollo a utilizar

Frontend (Lado del Cliente)

HTML5 - Lenguaje actual estándar de marcado para estructura web. Se justifica su uso porque es el estándar universal, semántico, accesible y compatible con todos los navegadores modernos, garantizando que el sistema ComunidadLibro sea accesible desde cualquier dispositivo.

CSS3 - Estándar actual Hojas de estilo para diseño visual. Permite diseño responsive, animaciones nativas y mejor rendimiento que frameworks pesados, asegurando una experiencia visual atractiva y funcional en dispositivos móviles y de escritorio.

Bootstrap 5.3.x Framework CSS responsivo. Acelera el desarrollo, garantiza consistencia visual y responsividad en todos los dispositivos. Para bibliotecas comunitarias es crucial que el sistema sea accesible desde tabletas y teléfonos inteligentes de los usuarios.

JavaScript ES6+ Lenguaje de programación del lado cliente. Proporciona actividad nativa, validaciones en tiempo real y mejor experiencia de usuario, elementos esenciales para formularios de préstamos y búsquedas dinámicas.

jQuery 3.7.x Librería JavaScript que simplifica la manipulación del DOM y llamadas AJAX, reduciendo el tiempo de desarrollo y facilitando la implementación de funcionalidades interactivas como búsquedas en tiempo real.

Backend (Lado del Servidor)

Python 3.9+ Lenguaje de programación principal seleccionado por su sintaxis clara, amplia comunidad, excelente documentación y capacidad para desarrollo rápido y mantenible. Es ideal para equipos con diferentes niveles de experiencia.

Flask 2.3.x Framework web minimalista que es ligero, flexible, fácil de aprender y perfecto para proyectos de mediana escala como ComunidadLibro. No impone una estructura rígida, permitiendo adaptarse a las necesidades específicas de las bibliotecas.

SQLAlchemy 2.0.x ORM (Object-Relational Mapping) que abstrae la base de datos, facilita migraciones y reduce errores SQL. Permite al equipo trabajar con objetos Python en lugar de SQL directo, aumentando la productividad.

Flask-Login 0.6.x Extensión para autenticación que proporciona manejo seguro de sesiones y autenticación de usuarios, crucial para diferenciar entre bibliotecarios y lectores en el sistema.

Werkzeug 2.3.x Librería WSGI incluida en Flask que ofrece manejo robusto de solicitudes HTTP y herramientas de desarrollo, proporcionando la base sólida para el servidor web.

Base de datos

MySQL 8.0+ Sistema de gestión de base de datos relacional seleccionado por ser estable, confiable, con amplio soporte comunitario y excelente rendimiento para aplicaciones web. Es familiar para la mayoría de desarrolladores y administradores de sistemas.

phpMyAdmin 5.2.x Interfaz web para administración MySQL que facilita la administración de base de datos durante el desarrollo y mantenimiento, permitiendo que personal no técnico pueda realizar tareas básicas de administración.

Herramientas de desarrollo

Git 2.40+ Control de versiones que es el estándar de la industria, permite trabajo colaborativo efectivo y seguimiento detallado de cambios, esencial para el desarrollo en equipo durante los 4 sprints.

GitHub Plataforma de repositorio remoto que ofrece integración con herramientas de CI/CD, seguimiento de problemas y documentación. Facilita la colaboración del equipo Scrum y el seguimiento de historias de usuario.

Visual Studio Code 1.80+ Editor de código ligero y extensible con excelente soporte para Python y desarrollo web. Incluye depuración integrada, extensiones para Flask y herramientas de productividad.

Postman 10.x Herramienta para testing de APIs que facilita pruebas de endpoints y documentación de APIs, crucial para validar la funcionalidad backend durante el desarrollo.

Herramientas de Gestión Ágil

Trello Aplicación web para gestión de tareas Scrum con interfaz intuitiva, colaboración en tiempo real y seguimiento visual del progreso. Ideal para que el Product Owner y stakeholders visualicen el avance del proyecto.

GitHub Project Sistema integrado de seguimiento técnico que se conecta nativamente con código, issues y pull request, facilitando la trazabilidad entre historias de usuario y código implementado.

Slack Plataforma de comunicación del equipo que permite comunicación asíncrona e integración con herramientas de desarrollo, esencial para Daily Scrums y comunicación continua del equipo.

5.2 Justificación del Stack Tecnológico

Ventajas del Stack Seleccionado:

Curva de Aprendizaje Moderada Python y Flask son tecnologías accesibles tanto para desarrolladores junior como senior. HTML5, CSS3 y JavaScript son fundamentales y ampliamente conocidos en la industria. Bootstrap acelera el desarrollo frontend sin requerir experiencia avanzada en CSS, permitiendo que el equipo se enfoque en la lógica de negocio específica de bibliotecas.

Desarrollo Rápido y Eficiente Flask permite crear aplicaciones web funcionales con mínimo código repetitivo, ideal para el cronograma de 8 semanas del proyecto. Bootstrap proporciona componentes prediseñados que aceleran la creación de interfaces de usuario. SQLAlchemy simplifica las operaciones de base de datos con un ORM intuitivo. El stack completo permite el desarrollo full-stack con Python como lenguaje principal, reduciendo la complejidad tecnológica.

Escalabilidad y Mantenibilidad La arquitectura modular de Flask facilita el crecimiento futuro del sistema cuando las bibliotecas requieran nuevas funcionalidades. Python tiene un excelente soporte para refactorización y pruebas automatizadas. MySQL puede manejar eficientemente el crecimiento de datos de múltiples bibliotecas comunitarias. La separación clara entre frontend y backend facilita el mantenimiento a largo plazo.

Costo-Efectividad Todas las tecnologías seleccionadas son de código abierto, eliminando costos de licenciamiento que podrían ser prohibitivos para bibliotecas comunitarias. Existe amplia disponibilidad de hosting económico para aplicaciones Python/MySQL. La comunidad activa reduce los costos de soporte y acelera la resolución de problemas. La documentación extensa reduce significativamente el tiempo de desarrollo.

Compatibilidad y Estándares El stack es completamente compatible con los estándares web modernos y funciona en todos los navegadores principales. Bootstrap garantiza un diseño responsivo nativo para dispositivos móviles. Las API REST estándar facilitan futuras integraciones con otros sistemas bibliotecarios o gubernamentales.

Consideraciones Técnicas Específicas:

Para Bibliotecas Comunitarias La simplicidad de implementación permite que el stack se ejecute en servidores básicos o hosting compartido económico. La facilidad de mantenimiento significa que el personal técnico local puede aprender estas tecnologías para soporte básico. MySQL ofrece herramientas robustas de copia de seguridad y recuperación, cruciales para proteger datos de préstamos y usuarios. Flask-Login y Werkzeug proporcionan seguridad web estándar sin complejidad excesiva.

Para el Equipo de Desarrollo La alta productividad permite desarrollo rápido de prototipos y MVPs, esencial para validación temprana con stakeholders. Python incluye excelentes marcos de pruebas como pytest y unittest para asegurar la calidad. Las herramientas de depuración integradas en Flask aceleran la resolución de problemas. Git y GitHub facilitan el trabajo colaborativo distribuido necesario para la metodología Scrum.

5.3 Arquitectura del Sistema

El sistema ComunidadLibro implementará una arquitectura de tres capas claramente definidas:

Capa de Presentación (Frontend) Compuesta por HTML5, CSS3, Bootstrap y JavaScript, esta capa maneja toda la interacción con usuarios finales. Los bibliotecarios accederán a formularios de gestión de libros y préstamos, mientras que los lectores utilizarán interfaces de búsqueda y consulta de historial. La responsividad garantiza el acceso desde cualquier dispositivo.

Capa de Lógica de Negocio (Backend) Implementada en Python Flask con SQLAlchemy, procesa todas las reglas de negocio específicas de bibliotecas: validación de préstamos, cálculo de fechas de devolución, generación de informes y gestión de usuarios. Los controladores Flask manejan las peticiones HTTP y coordinan con los modelos de datos.

Capa de Datos (Base de datos) MySQL almacena toda la información persistente: catálogo de libros, datos de usuarios, historial de préstamos y configuraciones del sistema. El diseño relacional optimizado garantiza integridad referencial y consultas eficientes para informes y búsquedas.

5.4 Requisitos del Entorno de Desarrollo

Servidor de Desarrollo El entorno requiere sistema operativo Windows 10+, macOS 10.15+, o Linux Ubuntu 20.04+. Se necesita un mínimo de 4 GB de RAM, aunque se recomiendan 8 GB para un desarrollo cómodo. El almacenamiento debe tener al menos 10 GB libres para el entorno de desarrollo, dependencias y base de datos de prueba.

Software Base Requerido Python 3.9+ debe estar instalado con pip para gestión de paquetes. MySQL Server 8.0+ proporcionará la base de datos de desarrollo. Git es esencial para el control de versiones y colaboración del equipo.

Herramientas de Productividad Visual Studio Code se recomienda como editor principal por su excelente soporte para Python y extensiones web. Un navegador web moderno es necesario para probar las interfaces. Postman facilitará las pruebas de APIs durante el desarrollo backend.

Esta tecnológica garantiza un desarrollo eficiente, mantenible y escalable para el sistema de selección ComunidadLibro, adaptándose perfectamente a las necesidades específicas de bibliotecas comunitarias y las capacidades técnicas del equipo de desarrollo asignadas al proyecto.

# Definición de tecnologías de implementación.

6.1 Servidor de aplicaciones

Opción Recomendada: Hosting Compartido

Proveedor sugerido: Hostinger o SiteGround Para el sistema ComunidadLibro se recomienda utilizar un hosting compartido que soporte Python y MySQL. Esta opción es la más económica y práctica para bibliotecas comunitarias que están iniciando con tecnología.

Especificaciones mínimas:

2 GB de espacio en disco

Soporte para Python 3.9+

Base de datos MySQL incluida

Panel de control cPanel o similar

Justificación: Es fácil de administrar, no requiere conocimientos técnicos avanzados, incluye soporte técnico del proveedor y es accesible para organizaciones comunitarias.

Opción Alternativa: VPS Básico

Proveedor: DigitalOcean o Vultr

Servidor virtual con 1 GB de RAM, 1 CPU, 25 GB de SSD

Sistema operativo Ubuntu 20.04

Requiere conocimientos básicos de administración de servidores.

6.2 Servidor de base de datos

MySQL en el mismo servidor

La base de datos MySQL se instalará en el mismo servidor donde está la aplicación web. Esto simplifica la configuración y reduce la complejidad técnica.

Configuración:

MySQL 8.0 o superior

Una base de datos dedicada a ComunidadLibro

Usuario específico con permisos limitados

Respaldos semanales automáticos

Espacio estimado:

Base de datos: 500MB - 2GB (dependiendo del número de libros y usuarios)

Respaldos: 1GB adicional

6.3 Servicios Básicos Necesarios

Dominio Web

Registro de dominio: bibliotecaXXX.com o .org

Proveedor: Namecheap, GoDaddy o el mismo hosting

Certificado SSL

Certificado SSL básico (incluido en la mayoría de hostings)

Garantizar conexiones seguras HTTPS

Proteger datos de usuarios y bibliotecarios

Correo electrónico básico

Cuenta de correo electrónico: admin@bibliotecaXXX.com

Para notificaciones del sistema

Generalmente incluido en el hosting

6.4 Herramientas de Administración

Panel de control

cPanel o Plesk Interfaz web para administrar el hosting sin conocimientos técnicos avanzados. Permite:

Gestionar archivos del sitio web

Administrador de base de datos MySQL

Configurar cuentas de correo electrónico

Ver estadísticas de uso

Acceso FTP

FileZilla (gratuito) Para subir y actualizar archivos del sistema ComunidadLibro al servidor.

Administración de Base de Datos

phpMyAdmin Herramienta web incluida en el hosting para administrar la base de datos MySQL de forma visual.

6.5 Configuración de Seguridad Básica

Respaldos

Respaldo automático semanal de la base de datos

Descarga mensual de respaldos a computadora local

Respaldo de archivos del sistema antes de actualizaciones

Contraseñas Seguras

Contraseña robusta para acceso al hosting

Contraseña diferente para la base de datos

Cambio de contraseñas cada 6 meses

Actualizaciones

Mantener actualizado el sistema ComunidadLibro

Aplique parches de seguridad cuando estén disponibles

6.6 Requisitos Técnicos del Servidor

Especificaciones mínimas:

Procesador: 1 núcleo de CPU

Memoria RAM: 1GB mínimo, 2GB recomendado

Almacenamiento: 5GB disponibles

Ancho de banda: 100 GB/mes

Sistema operativo: Linux (Ubuntu, CentOS) o Windows Server

Software requerido:

Servidor web: Apache 2.4+ o Nginx 1.18+

Python: Versión 3.9 o superior

MySQL: Versión 8.0 o superior

SSL: Certificado válido instalado

6.7 Plan de Implementación Simple

Semana 1: Contratación de Servicios

Contratar hosting compartido

Registrador de dominio

Configurar cuentas de acceso

Semana 7: Instalación del Sistema

Subir archivos de ComunidadLibro al servidor

Crear base de datos MySQL

Configurar conexión entre aplicación y base de datos

Probar funcionamiento básico

Semana 8: Configuración Final

Configurar SSL y dominio

Crear usuarios iniciales del sistema

Realizar pruebas finales

Capacitar al bibliotecario personal

6.8 Soporte y mantenimiento

Soporte Técnico:

Soporte del proveedor de hosting (24/7 básico)

Manual de usuario para bibliotecarios

Contacto con desarrollador para problemas específicos

Mantenimiento básico:

Verificar respaldos semanalmente

Revisar funcionamiento mensualmente

Actualizar sistema cuando sea necesario

Capacitación requerida:

2 horas de capacitación para bibliotecario principal

Manual básico de uso del sistema.

Video tutorial de 30 minutos para funciones principales

Esta configuración básica garantiza que el sistema ComunidadLibro funcione de manera confiable, siendo fácil de administrar para personal de bibliotecas comunitarias sin experiencia técnica avanzada.

**Roadmap**

1. Adjunta la imagen de tu Roadmap:



1. Adjunta el link de acceso a archivo original guardado en el repositorio del proyecto:
2. Adjunta el link de acceso a tu tablero Trello:

https://trello.com/b/3mQrSrHJ

1. Adjunta el link de acceso al drive de tu proyecto:

**Parte II: Presentación**

Pega el link que se generó al culminar la grabación de tu presentación en Teams





**Duoc UC**