

**Departamento de Ciencias de la Computación
(DCCO)**

Carrera de Ingeniería de Software

Prácticas Pre Profesionales (PPP)

Perfil del Proyecto

Presentado por: Pozo Steven

Tutor académico: Ing. Ruíz Jenny

Ciudad: Quito

Fecha: 25/10/2024

ÍNDICE

1.	Introducción	5
2.	Planteamiento del trabajo.....	5
2.1.	Formulación del problema.....	5
2.2.	Justificación.....	6
3.	Sistema de Objetivos.....	6
3.1.	Objetivo General	6
3.2.	Objetivos Específicos (03).....	6
4.	Alcance	7
5.	Marco Teórico.....	7
5.1.	Metodología (Marco de trabajo 5W+2H).....	8
6.	Ideas a Defender	11
7.	Resultados Esperados.....	11
8.	Viabilidad (Ej.)	12
8.1.	Humana.....	12
8.1.1.	Tutor Empresarial.....	12
8.1.2.	Tutor Académico.....	13
8.1.3.	Estudiantes	13
8.2.	Tecnológica	13
8.2.1.	Hardware	13
8.2.2.	Software	14
9.	Conclusiones y recomendaciones	14
9.1.	Conclusiones.....	14
9.2.	Recomendaciones	14
10.	Planificación para el Cronograma:.....	15
11.	Referencias.....	17

1. Introducción

Actualmente los sistemas automatizados para cualquier tipo de gestión dentro de una institución son importantes para agilizar las actividades. Sin embargo, el colegio Saint Patrik School que cuenta con una biblioteca extensa, por mucho tiempo ha venido gestionando los usuarios, libros y préstamos mediante una hoja de cálculo de Excel. Esto ha ocasionado redundancia de datos que imposibilitan realizar préstamos de libros en base a únicos libros o usuarios.

Para resolver estos problemas, se ha planteado el desarrollo de una aplicación backend que simplifique la gestión eficiente de los datos de la biblioteca. Este proyecto se enfoca en la creación de un servidor que actúe como una API para administrar las transacciones de préstamos de libros, manteniendo las reglas de negocios y asegurando la uniformidad de la información. Además, se planea la creación y puesta en marcha de una base de datos que guardará de manera ordenada y segura la información de los usuarios, libros y transacciones.

Finalmente, se incluirán documentos de soporte, tales como el manual técnico y los casos de uso, los cuales actuarán como orientaciones para el mantenimiento y futura ampliación del sistema, garantizando que el equipo técnico pueda entender y expandir la funcionalidad API.

2. Planteamiento del trabajo

2.1. Formulación del problema

El colegio Saint Patrik School necesita un sistema que automatice la gestión de su biblioteca. Para solucionar estas limitaciones, se propone el desarrollo de una aplicación backend con una API que permita administrar de forma precisa y centralizada los procesos de préstamos, creación de usuarios y gestión de libros. Este sistema permitirá mantener la unicidad de cada usuario y libro, minimizando los errores derivados de la manipulación de datos y garantizando la integridad de la información almacenada. La API y base de datos se alojarán de manera local para restringir el acceso a los datos únicamente al bibliotecario y al administrador del

sistema, brindando un mayor control y seguridad en la administración de la biblioteca.

2.2. Justificación

Es común encontrar que muchas instituciones o microempresas aún gestionan sus datos mediante hojas de cálculo para procesos internos, lo que puede llevar a errores, redundancia de datos y problemas de integridad en la información. Frente a esta realidad, la implementación de una aplicación backend basada en una arquitectura cliente-servidor local se presenta como una solución confiable y segura para la gestión de bibliotecas institucionales.

El desarrollo de un sistema de este tipo permite centralizar el almacenamiento y acceso a la información de la biblioteca, minimizando los errores derivados del manejo manual de datos. Además, una arquitectura cliente-servidor local garantiza que los datos permanezcan bajo control exclusivo del bibliotecario y el administrador del sistema, evitando la exposición de información en plataformas externas o en servicios de hospedaje que pudieran poner en riesgo la privacidad de los datos. Esta solución responde a las necesidades de instituciones como el colegio Saint Patrik School, donde la integridad y seguridad de los datos son primordiales.

3. Sistema de Objetivos

3.1. Objetivo General

Desarrollar una aplicación backend en una arquitectura cliente-servidor que automatiza la administración de la biblioteca de la escuela Saint Patrick School a través de una API y base de datos, empleando el framework de Java Springboot. Este sistema posibilitará disminuir las equivocaciones en la gestión de datos y preservar la integridad de los datos, uniendo los datos de libros, usuarios y transacciones de crédito de forma segura y eficaz en un ambiente local.

3.2. Objetivos Específicos (03)

- Elaborar una API RESTful en Spring Boot que facilite la administración de las operaciones de préstamo, registro y consulta de libros, proporcionando a los usuarios una interfaz de interacción fácil de entender y facilitando el acceso centralizado a la información. Esta API asegurará la veracidad de los datos en

cada operación y establecer los puntos de acceso, servicios y organización de la base de datos en Spring Boot.

- Implementar un módulo de autenticación y control de acceso basado en roles en Spring Boot, que permita establecer permisos diferenciados para administradores, bibliotecarios y usuarios. Este módulo protegerá la información sensible de accesos no autorizados y garantizará la integridad de los datos.
- Implementar un módulo de bitácora y de ranks, que permitirán ver los libros más leídos y los usuarios que más han leído, así también como la lista de todos los préstamos, con el fin de actualizar stock, premiar alumnos y visualizar todos los prestatarios que deben un libro

4. Alcance

La automatización de la biblioteca permitirá gestionar usuarios, libros y préstamos mediante solicitudes realizadas desde el Frontend hacia el Backend, siendo las APIs en Java Spring Boot las encargadas de responder a dichas solicitudes. En cuanto a los usuarios, se podrá agregar, editar y deshabilitar su estadía dentro de la institución; para los libros, se permitirá agregar, editar o deshabilitar su disponibilidad; y para los préstamos, se gestionará la relación entre usuarios y libros, registrando las fechas de adquisición y devolución para controlar el tiempo límite del préstamo. Adicionalmente, se incluirán dos secciones: una bitácora para mostrar los préstamos no devueltos y los devueltos, y un ranking de estudiantes que han leído más libros. Como complemento, se elaborará documentación detallada, incluyendo casos de uso y un SRS (Software Requirements Specification), además de un respaldo de la arquitectura de la base de datos, describiendo las relaciones entre tablas y entidades principales.

5. Marco Teórico

IntelliJ IDEA Community Edition

IntelliJ es un Editor de Código compatible con Java y Kotlin. Además, permite ver sugerencias relevantes en cualquier contexto que facilita la navegación dentro del código. También ofrece una amplia gama de herramientas que permiten realizar todo

tipo de trabajo, como es el caso de Java Spring Boot, framework utilizado para realizar el backend de la automatización de la biblioteca. (IntelliJ IDEA)

Java Springboot

Java Spring Boot es un entorno de desarrollo fundamentado en el ecosistema Spring, que simplifica la construcción de aplicaciones backend sólidas y de nivel de producción con una configuración simplificada. Facilita a los programadores la creación de aplicaciones independientes que pueden funcionar de manera directa, sin requerir servidores externos, mediante la incorporación de servidores como Tomcat o Jetty. Su mayor beneficio radica en la automática configuración de componentes y dependencias, lo que facilita el proceso de desarrollo al suprimir la necesidad de configuraciones XML complicadas. Spring Boot resulta particularmente beneficioso para la creación de APIs RESTful, proporcionando herramientas como Spring Boot. (Springboot)

5.1. Metodología (Marco de trabajo 5W+2H)

Debe explicar paso a paso el desarrollo de la guía con la herramienta de Excel aplicando el marco de trabajo de las 5W y 2H

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?	¿CUÁNTO?	% DE CUMPLIMIENTO
Introducción del proyecto y documentación de casos de usos V1.0.0	Mediante reuniones presenciales y virtuales	Supervisor Steven Marley	30/09/2024 - 04/10/2024	Porque es necesario conocer el contexto del proyecto y documentar sus funcionalidades.	Sin costo	100%
Creación de la base de datos, introducción a Spring Boot y creación de arquitectura en capas.	Online	Steven	7/10/2024 - 10/10/2024	Para implementar Java SpringBoot como Backend del aplicativo de escritorio.	Sin costo	100%

Construcción de CRUD APIS	Online	Steven	14/10/2024 – 18/10/2024	Para permitir la comunicación del cliente con la base de datos por medio del servidor.	Sin costo	100%
Validación de datos	Online	Steven	21/10/2024 – 25/10/2024	Porque es importante controlar la forma en la que la información entra por los campos de texto.	Sin costo	100%
Cambios y construcción de nuevas APIS en el FrontEnd	Online	Steven	28/10/2024 – 31/10/2024	Porque el sistema puede necesitar nuevas apis o cambios de apis existentes para mejorar su funcionalidad.	Sin costo	100%
Cambios y construcción de nuevas APIS en el FrontEnd - Fase II	Online	Steven	5/11/2024 – 8/11/2024	Porque en la fase II se implementa la creación de nuevas apis así también como un método de backup local para la base de datos.	Sin costo	100%
Backup de la base de datos						
Refactorización de APIS (si es necesario)	Online	Steven	11/11/2024 – 15/11/2024	Es necesario mejorar el código para serlo más legible, en caso de ser necesario	Sin costo	100%

Pruebas Unitarias	Online	Steven	18/11/2024 – 22/11/2024	Porque es importante validar su funcionamiento de los métodos usados en las APIS mediante simulaciones del servicio.	Sin costo	100%
Pruebas de vulnerabilidad	Online	Steven	25/11/2024 – 29/11/2024	Identificar vulnerabilidad o riesgos en cuestión de datos y consultas.	Sin costo	100%
Pruebas de vulnerabilidad – Fase II	Online	Steven	02/12/2024 – 06/12/2024	Identificar vulnerabilidad o riesgos en cuestión de datos y consultas.	Sin costo	100%
Carga de información inicial al sistema	Online	Steven	16/12/2024 – 20/12/2024	Para migrar la información de la base de datos de Excel a la base de datos MySQL	Sin costo	100%
Capacitación del uso del sistema en sitio	Presencial	Supervisor Steven Marley	23/12/2024 – 24/12/2024	Para que la bibliotecaria tenga conocimiento sobre el funcionamiento del sistema	Sin costo	100%

Tabla 1 Marco de trabajo 5W+2H

6. Ideas a Defender

En primer lugar, se respalda la arquitectura cliente-servidor con Java Spring Boot, lo que facilita la creación de un sistema modular y adaptable, idóneo para administrar las operaciones de la biblioteca de manera eficaz y escalable. Spring Boot simplifica la puesta en marcha de una API RESTful sólida y segura, que tiene la habilidad de gestionar las interacciones entre el cliente y el servidor de forma rápida y con el menor esfuerzo en la configuración.

Otro concepto esencial es la protección en el acceso a la información a través de la puesta en marcha de un sistema de autenticación y autorización, garantizando que únicamente los usuarios con los permisos pertinentes tengan la capacidad de efectuar acciones como añadir o cambiar libros, usuarios o préstamos. Además, se promueve la implementación de buenas prácticas en el diseño de bases de datos, utilizando un modelo relacional que facilite una organización eficaz de la información relacionada con libros, usuarios y transacciones, y asegure la integridad y uniformidad de los datos. Finalmente, se promueve es la automatización de procesos, como la creación de reportes automáticos sobre préstamos, lo que incrementa la eficacia operacional del sistema. Finalmente, se subraya la relevancia de simplificar el mantenimiento y la escalabilidad del sistema mediante un diseño ordenado y limpio, lo cual facilitará futuras extensiones del sistema sin dificultades.

7. Resultados Esperados

- Primero, se anticipa una automatización eficaz de los procedimientos de administración de la biblioteca, que incluyen la gestión de usuarios, libros y préstamos, lo que disminuirá considerablemente la utilización de técnicas manuales y los fallos vinculados a estas. Con la puesta en marcha de una API sólida en Java Spring Boot, se prevé un backend robusto y escalable capaz de administrar las transacciones de forma eficaz y eficiente.
- Uno de los resultados más destacados se espera es la protección en la gestión de datos delicados, a través de un sistema de autenticación y autorización que asegurará que únicamente los usuarios con los permisos pertinentes tengan acceso a la información o efectuar modificaciones en el sistema. Además, se anticipa una estructura sólida de la base de datos, lo que facilitará una gestión ordenada de los

registros de usuarios, libros y transacciones, garantizando la integridad y coherencia de la información.

- Finalmente, se prevé que la escalabilidad del sistema facilite su adaptación a futuras demandas de la biblioteca, ya sea incorporando más funcionalidades o administrando un volumen de datos más grande conforme la institución se expanda. Para concluir, se anticipa que este proyecto brinde una solución eficaz, segura y de fácil mantenimiento para la administración de la biblioteca.

8. Viabilidad (Ej.)

Cantidad	Descripción	Valor Unitario (USD)	Valor Total (USD)
	Equipo en casa		
1	Laptop dell 3593 Inspiron 15300	1250	1250
1	Monitor Dell 22 Fhd Ips Led	100	100
	Software		
1	Sistema operativo Windows 11 Pro	0	0
1	Visual Studio Code	0	0
1	IntelliJ IDEA Community	0	0
1	Workbench	0	0
1	Github Desktop	0	0
1	Figma	0	0
1	Lucidchart	0	0
		TOTAL	1350

Tabla 2 Presupuesto del proyecto

8.1. Humana

8.1.1. Tutor Empresarial

Ing. Jaime Vela.

Responsabilidades:

- Gerente de la empresa.

- Encargado de la instalación y supervisión del programa HikVision integrado a cámaras de seguridad.
- Manejo de sistema de biométricos.

8.1.2. Tutor Académico

Ing. Jenny Ruíz.

Responsabilidades:

- Supervisa los avances, aplicaciones y aprendizajes obtenidos en las prácticas preprofesionales.

8.1.3. Estudiantes

Steven Pozo

Responsabilidades:

- Responsable del backend del sistema.
- Desarrollo de APIS restful con Java SpringBoot.
- Encargado de la manipulación de la base de datos, backups y configuración del sistema operativo.
- Desarrollo de casos de uso.
- Desarrollo de documento de SRS (Software Requirement Specification).
- Pruebas de software y funcionalidad del sistema.

8.2. Tecnológica

Para llevar a cabo este proyecto, se ha tomado un framework de Java que permita establecer comunicación con el cliente de manera eficiente, ya que, Java Springboot es un framework sólido que permite crear APIS y manejar su seguridad dentro de su estructura de 4 capas conocida como arquitectura en capas. A parte, las especificaciones como la memoria RAM y el almacenamiento del computador se muestran en la sección 8.2.1 y el software usado en la sección 8.2.2

8.2.1. Hardware

	Requisitos mínimos	Disponibilidad
--	--------------------	----------------

Memoria RAM	32GB GB de RAM	Alta
Almacenamiento	476 GB disco C 1 terabyte disco D	Alta

Tabla 3 Requisitos de Hardware

8.2.2. Software

	Requisitos mínimos	Disponibilidad
Sistema Operativo	Se recomienda Windows 10 Pro	Alta
IDE	Es recomendable Visual Studio Code para realizar las pruebas con Thunder Client o Postman con las APIs. Usar IntelliJ IDEA Community, para desarrollar las apis y usarlo gratuitamente	Alta

Tabla 4 Requisitos de Software

9. Conclusiones y recomendaciones

9.1. Conclusiones

9.2. Recomendaciones

10. Planificación para el Cronograma:

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?	¿CUÁNTO?	% DE CUMPLIMIENTO
Introducción del proyecto y documentación de casos de usos V1.0.0	Mediante reuniones presenciales y virtuales	Supervisor Steven Marley	30/09/2024 04/10/2024	- Porque es necesario conocer el contexto del proyecto y documentar sus funcionalidades.	Sin costo	100%
Creación de la base de datos, introducción a Spring Boot y creación de arquitectura en capas.	Online	Steven	7/10/2024 10/10/2024	- Para implementar Java SpringBoot como Backend del aplicativo de escritorio.	Sin costo	100%
Construcción de CRUD APIS	Online	Steven	14/10/2024 18/10/2024	- Para permitir la comunicación del cliente con la base de datos por medio del servidor.	Sin costo	100%
Validación de datos	Online	Steven	21/10/2024 25/10/2024	- Porque es importante controlar la forma en la que la información entra por los campos de texto.	Sin costo	100%
Cambios y construcción de nuevas APIS necesitas en el FrontEnd	Online	Steven	28/10/2024 31/10/2024	- Porque el sistema puede necesitar nuevas apis o cambios de apis existentes para mejorar su funcionalidad.	Sin costo	100%
Cambios y construcción de nuevas APIS necesitas en el FrontEnd - Fase II	Online	Steven	5/11/2024 – 8/11/2024	Porque en la fase II se implementa la creación de nuevas apis así también como un método	Sin costo	100%

Backup de la base de datos				de backup local para la base de datos.		
Refactorización de APIS (si es necesario)	Online	Steven	11/11/2024 15/11/2024	– Es necesario mejorar el código para serlo más legible, en caso de ser necesario	Sin costo	100%
Pruebas Unitarias	Online	Steven	18/11/2024 22/11/2024	– Porque es importante validar su funcionamiento mediante pruebas. Y pruebas de entorno para asegurar su funcionamiento sin arriesgar el hardware físico del sistema.	Sin costo	100%
Pruebas de vulnerabilidad	Online	Steven	25/11/2024 29/11/2024	– Identificar vulnerabilidad o riesgos en cuestión de datos y consultas.	Sin costo	100%
Pruebas de vulnerabilidad - Fase II	Online	Steven	02/12/2024 06/12/2024	– Comprobación para la configuración del sistema operativo.	Sin costo	100%
Carga de información inicial al sistema	Online	Steven	16/12/2024 20/12/2024	- Para migrar la información de la base de datos de Excel a la base de datos MySQL.	Sin costo	100%
Capacitación del uso del sistema en sitio	Presencial	Supervisor Steven Marley	23/12/2024 24/12/2024	– Para que la bibliotecaria tenga conocimiento sobre el funcionamiento del sistema	Sin costo	100%

11. Referencias

Página oficial de IntelliJ IDEA: *IntelliJ IDEA: el IDE líder para Java y Kotlin*. (s/f). JetBrains. Recuperado el 6 de noviembre de 2024, de <https://www.jetbrains.com/es-es/idea/>

Página oficial de Java Springboot: *Spring boot*. (s/f). Spring Boot. Recuperado el 6 de noviembre de 2024, de <https://spring.io/projects/spring-boot>