**PROYECTO PERSONAL**

**Biblioteca Pública Municipal**

Imagen que contiene texto, señal, plato, dibujo

Descripción generada automáticamente

**Fecha de inicio:** 02 / enero / 2024

**Fecha de finalización:**

**Estado del proyecto:** En desarrollo.

**Demostración del proyecto – Video YouTube:**

**Tecnologías utilizadas:** Backend con Java 17 LTS y MySQL 8.2, y Frontend con Java Swing.

**Objetivo:** Plasmar los conocimientos adquiridos en un proyecto de la vida real que me permita seguir practicando y aprendiendo sobre programación; demostrar por medio de un proyecto personal mis habilidades como programadora.

**Características del proyecto:** Sistema que permite gestionar el inventario de una Biblioteca Pública Municipal, realizar préstamos de libros e historietas y gestionar las devoluciones de estos.

**Autora:** María Raquel Martínez Urrutia

Estudiante de Ing. en Desarrollo de Software en UnADM

**LinkedIn:** www.linkedin.com/in/raqmartinez

**Email:** [meztinea.dev@gmail.com](mailto:meztinea.dev@gmail.com)

# Requerimientos del Sistema

La biblioteca pública del municipio de Pabellón de Arteaga necesita un sistema de software que le permita gestionar el inventario de libros e historietas, llevar un registro de los usuarios, préstamos y devoluciones del material. Es necesario que el sistema cuente con una interfaz gráfica intuitiva para que el personal de la biblioteca pueda hacer uso fácilmente y que los datos puedan ser almacenados para consultarlos posteriormente.

Sobre los libros es necesario almacenar el título, el autor, la editorial, la imagen de portada y la cantidad de ejemplares disponibles para préstamos. Sobre las historietas es necesario almacenar los mismos datos de los libros e incluir el número de volumen de la historieta.

Sobre los usuarios que solicitan los prestamos se desea almacenar su nombre completo, correo electrónico, número de teléfono, fotografía y fecha de registro. Es obligatorio que el usuario proporcione un correo electrónico o número de teléfono para poder ser registrado, si no cuenta con uno de los dos, no podrá registrarse.

Se desea que exista un registro de préstamos y devoluciones. Por cada libro o historieta prestado debe existir un registro de préstamo y su respectiva devolución. Sobre los préstamos es necesario registrar la fecha del préstamo, el título del libro o historieta, el autor de la obra, la editorial y el nombre completo del usuario que solicitó el préstamo. También es necesario contar con una funcionalidad que permita identificar aquellos préstamos que están pendientes de devolución y tienen más de 7 días desde la fecha de préstamo, para que el encargado de la biblioteca pueda tomar las medidas pertinentes sobre estos préstamos.

Las reglas para los préstamos son las siguientes:

* Un usuario puede solicitar solo un libro o historieta en cada préstamo.
* Un usuario no puede solicitar un libro o historieta si ya tiene un préstamo pendiente de devolución. Aunque ya haya pasado meses o años.
* Cada préstamo debe ser devuelto por el usuario en un lapso máximo de siete días.

## Requerimientos funcionales y no funcionales

Con base en lo solicitado por el cliente se han identificado los requerimientos funcionales y no funcionales que han de guiar la realización del proyecto.

**Requerimientos funcionales**

1. Es necesario que el sistema cuente con una interfaz gráfica que permita a los usuarios manipularlo de manera eficiente y fácil.
2. Es necesario que el sistema se conecte con una base de datos para almacenar la información de manera persistente y permita consultar los datos en el futuro.
3. El sistema debe contar con una arquitectura que permita realizar mantenimiento de manera fácil y eficaz.

**Requerimientos no funcionales**

1. El sistema debe contar con una función de tipo filtro para identificar los préstamos que están pendientes de devolución.
2. La interfaz gráfica debe ser amigable con el usuario haciendo uso de elementos gráficos para ingresar datos y realizar acciones.
3. El sistema debe contar con un apartado de altas para agregar nuevos libros e historietas.

## Arquitectura de 3 Capas

Después de realizar el análisis de los requerimientos se identifica como una buena opción de modelo arquitectónico para el desarrollo del sistema el modelo de 3 Capas. Las capas en la que se dividirá la arquitectura son: capa de presentación, capa de lógica y capa de persistencia.

***Capa de presentación***: La capa de presentación será la encargada de tener contacto directo con el usuario por medio de elementos gráficos (etiquetas, botones, cajas de texto, etc). Esta capa se encargará de facilitar al usuario (bibliotecaria/o) el uso del sistema presentando una interfaz gráfica amigable y de fácil uso. La capa de lógica se comunicará directamente con la capa de presentación.

***Capa de lógica***: La capa de lógica será la encargada de gestionar las reglas de negocio para el proyecto, implementará medidas de seguridad y gestionará las transacciones entre las capas. Actuará como un intermediario entre la capa de presentación y la capa de persistencia, comunicándose con ambas.

***Capa de persistencia***: Esta capa se encargará de realizar y gestionar la conexión con la base de datos. Gestionará la conexión y transacciones en la base de datos, realizando las sentencias SQL necesarias para crear, leer, actualizar y eliminar datos de la base de datos.

Además de las capas mencionadas también participa la base de datos, que no se considera una capa en esta arquitectura, pero tiene una de las funciones más importantes del sistema, almacenar los datos de manera que persistan en el tiempo.

Diagrama

Descripción generada automáticamente

Ilustración . Arquitectura del Proyecto. Modelo Arquitectónico de 3 Capas.

## Base de datos

Dado que el proyecto se ha de realizar con el lenguaje Java, el Sistema Gestor de Base de Datos (SGBD) MySQL se considera como la mejor opción para la creación y gestión de la base de datos.

Se han identificado las entidades que conforman la base de datos por medio de los requerimientos que el cliente ha mencionado anteriormente. Las tablas que definirán la estructura de la base de datos son: usuarios, historietas, libros, prestamos de libros, prestamos de historietas y devoluciones.

El Diagrama Relacional muestra todas las entidades involucradas en el almacenamiento de los datos importantes.

Diagrama

Descripción generada automáticamente

Ilustración . Diagrama Relacional de la Base de Datos.

## Gestión del Proyecto (SCRUM)

**Sprint 1 - Definición del proyecto Pt 1 (1 Semana) 1-7 enero 2024**

* Documentar las especificaciones del proyecto.
* Establecer los requisitos funcionales y no funcionales del proyecto.

**Sprint 2 - Definición del proyecto Pt 2 (1 Semana) 8 -14 enero 2024**

* Definir la arquitectura que seguirá el proyecto.
* Definir la estructura de la base de datos.

**Sprint 3 - Creación de la arquitectura del proyecto, la BD y la ventana principal de la IGU (1 Semana) 15 – 21 enero 2024**

* Generar la división de capas (arquitectura) en el proyecto de Netbeans.
* Crear la estructura de la base de datos en el SGBD MySQL usando lenguaje SQL.
* Crear la ventana principal de la Interfaz Gráfica.

**Sprint 4 – Crear las funcionalidades para guardar libros (IGU, lógica y persistencia) (1 Semana) 22 – 28 enero 2024**

* Crear la interfaz gráfica para guardar libros (capa presentación).
* Crear la lógica para crear libros (capa lógica).
* Crear la sentencia SQL para guardar libros en la base de datos (capa persistencia).

**Sprint 5 – Traer los datos de los libros guardados (SELECT) y mostrarlos en una tabla (1 Semana) 29 enero – 04 febrero 2024**

* Crear una tabla para mostrar los datos de los libros guardados.
* Crear la lógica que permita solicitar los datos a la persistencia y convertir al tipo de dato adecuado.
* Crear la sentencia SQL para solicitar los libros almacenados en la tabla libros.

Historias de usuario

## Desafíos del proyecto

En esta sección se incorporan los desafíos con los que he tenido que lidiar al momento de realizar este proyecto.

**Conversión de una imagen a tipo de dato byte[] (arreglo de bytes) para almacenar en la base de datos.**