



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES ARAGON**

**DESARROLLO DE INTERFAZ WEB PARA LA CIRCULACION  
DE MATERIAL BIBLIOGRAFICO EN BIBLIOTECAS DE LA  
UNAM**

**TESIS**

**QUE PARA OBTENER EL TITULO DE  
INGENIERO EN COMPUTACIÓN**

**PRESENTAN**

**CONTRERAS TAPIA JOSE LUIS  
TORRES RIOS SERGIO ALBERTO**

**DIRECTOR DE TESIS M en C.**

**PEÑALOZA ROMERO ERNESTO**



**CIUDAD NEZAHUALCÓYOTL, ESTADO DE MEXICO, CDMX**

**2021**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

---

---

# Índice

Capítulo I. Antecedentes .....	1
1.1 Dirección General de Bibliotecas .....	1
1.2 Circula .....	2
1.3 ALEPH .....	3
1.3 Koha .....	4
1.4 Definición del problema .....	5
1.5 Objetivo .....	5
1.6 Alcances .....	5
Capítulo II. Marco teórico .....	7
2.1 Ingeniería de software .....	7
2.1.1 Modelos del ciclo de vida .....	8
2.1.2 Análisis de requisitos .....	12
2.1.3 Diseño .....	13
2.1.4 Implementación .....	15
2.1.5 Pruebas .....	15
2.2 Aplicaciones web .....	16
2.2.1 Arquitectura cliente/servidor .....	16
2.2.2 Páginas web .....	17
2.2.3 Frameworks de JavaScript .....	20
2.2.4 Servicios web .....	22
2.3 Bases de datos .....	25
2.3.1 Características de las bases de datos .....	26
2.3.2 Lenguaje SQL .....	27
2.3.3 Sistema gestor de base de datos .....	28

---

---

2.3.4 Modelos de datos .....	30
2.3.6 Normalización de bases de datos.....	32
2.4 Sistema integrado de gestión bibliotecaria .....	32
2.4.1 Módulos de un SIGB.....	33
Capítulo III. Análisis de requerimientos.....	35
3.1 Requisitos del sistema .....	35
3.1.1 Requisitos funcionales.....	35
3.1.2 Requisitos no funcionales.....	36
3.1.3 Requerimientos mínimos del sistema .....	37
3.2 Análisis del módulo de circulación .....	38
Capítulo IV. Diseño y desarrollo del sistema.....	41
4.1 Diseño de la base de datos .....	41
4.1.1 Definición de tablas .....	42
4.1.2 Diagrama entidad relación.....	48
4.2 Diseño de la interfaz gráfica .....	49
4.2.1 Inicio de sesión.....	50
4.2.2 Usuarios .....	51
4.2.3 Inventario.....	58
4.2.4 Préstamo .....	64
4.2.5 Devolución.....	66
4.2.6 Configuración .....	67
4.3 Desarrollo de la interfaz gráfica .....	71
4.3.1 Autenticación .....	71
4.3.2 Usuarios .....	73
4.3.3 Inventario.....	80

---

---

4.3.4 Préstamo .....	85
4.3.5 Devolución.....	87
4.3.6 Configuración .....	88
Capítulo V. Pruebas y mantenimiento.....	91
5.1 Implantación .....	91
Prerrequisitos .....	91
Primeros pasos.....	91
5.2 Pruebas. ....	93
5.2.1 Operaciones de escritura desde el servidor de base de datos .....	93
5.2.2 Operaciones de escritura desde el sistema CIRCULA .....	94
5.2.3 Concurrencia .....	94
5.3 Mantenimiento .....	94
Conclusiones.....	96
Referencias.....	98

## **Capítulo I. Antecedentes**

### **1.1 Dirección General de Bibliotecas**

La Dirección General de Bibliotecas es la entidad universitaria encargada de coordinar el Sistema Bibliotecario y de Información de la Universidad Nacional Autónoma de México (SIBIUNAM) conforme a las políticas que establece el consejo del Sistema Universitario. (Dirección General de Bibliotecas, 2017)

La DGB tiene como propósito principal el asegurar que el SIBIUNAM cumpla con sus funciones para proporcionar a la comunidad universitaria los servicios bibliotecarios y de información que garanticen el acceso a la información para así contribuir al desarrollo académico y cultural.

Entre las distintas funciones que tiene la DGB tenemos las siguientes:

- Colaborar con entidades nacionales y extranjeras para realizar proyectos de acceso a la información.
- Garantizar el desarrollo y mantenimiento del sistema de información sobre el acervo perteneciente a la UNAM.
- Apoyar a las entidades y dependencias universitarias en la elaboración de proyectos para el desarrollo de las bibliotecas.
- Apoyar al programa de capacitación, actualización y desarrollo profesional del personal de las bibliotecas.
- Supervisar que el presupuesto asignado a la dependencia sea utilizado de acuerdo con la normativa vigente, y contribuir al uso racional de este mismo para la adquisición del acervo.
- Establecer los criterios para la prestación de servicios bibliotecarios y de información en las unidades de servicio bibliotecario de la dependencia y asegurarse de que esto se cumpla.
- Organizar, normalizar y actualizar constantemente los recursos bibliográficos de la UNAM

El 22 de marzo de 1954 el presidente Adolfo Ruiz Cortines entregó las instalaciones de Ciudad Universitaria e inauguró los primeros cursos en los edificios del Pedregal de San Ángel. El Departamento de Bibliotecas y el de la Biblioteca Central se instalaron en la Torre de Rectoría y eran dos órganos diferentes hasta que fueron ubicados en el edificio que hoy los alberga. (Villarello Reza, 2011)

El número de bibliotecas universitarias era de 40 y la Biblioteca Central contaba con 32 mil volúmenes. Para 1960 la Biblioteca Central aumentó su acervo a 40 mil volúmenes y asumió las funciones generales para todas las bibliotecas.

La DGB comenzó a operar en 1965, pero no fue hasta 1990 cuando se estableció el Sistema Bibliotecario Universitario como tal, el cual dio cuerpo al Reglamento General del Sistema Bibliotecario SIBIUNAM. Con la aprobación del reglamento por parte del Consejo Universitario, el sistema quedó formalmente conformado.

Se estableció como política que para convertir un acervo en biblioteca debe tener como mínimo las siguientes características: contar con volúmenes de materiales, documentales y publicaciones periódicas, contar con al menos una persona contratada para atender y realizar las actividades propias de una biblioteca, proporcionar los servicios de préstamos externos e internos, tener sus materiales procesados técnicamente, contar con un catálogo, disponer de un local destinado para las colecciones y como último tener un presupuesto específico para la compra de recursos de información.

## **1.2 Circula**

En un principio, las bibliotecas de la UNAM utilizaban un sistema de control manual para realizar los préstamos a domicilio y llevar un registro de estos haciendo uso de la tarjeta de préstamo de la obra y la credencial del lector. Estos registros se almacenaban físicamente en estantes dentro de la biblioteca.

Después de un tiempo surgió el problema de que, al ir aumentando el volumen de usuarios, el número de transacciones aumentaba con ellos por lo que se requería

de demasiado espacio. A demás, este servicio se hacía demasiado lento cuando varios usuarios intentaban acceder a él de manera simultánea. (Voutssás, 1987)

En 1986 el Ing. Juan Voutssás Márquez (Subdirector de Informática de la DGB) desarrolló el sistema CIRCULA con el fin de ayudar a realizar estas tareas de mejor manera haciendo uso de recursos tecnológicos. El sistema CIRCULA se encontraba dentro del sistema central BrittonLee y contaba con varias terminales en distintos pisos de la Biblioteca Central.

Este fue el primer sistema masivo de administración basado en código de barras y lector láser en México. (Voutssás, 2019) Para registrar los datos de un préstamo, el bibliotecario solo tenía que pasar el lápiz lector sobre la etiqueta del libro y de la credencial.

### **1.3 ALEPH**

En 1996 la DGB optó por utilizar ALEPH 300 (Automated Library Expandable Program), una alternativa de uso comercial la cual contaba con distintos módulos incluyendo el módulo de circulación. En el periodo de 1976 a 1996, la colección bibliográfica llegó a 9 millones de piezas documentales por lo que fue clave el uso de ALEPH para lograr una mejor organización de los materiales bibliográficos.

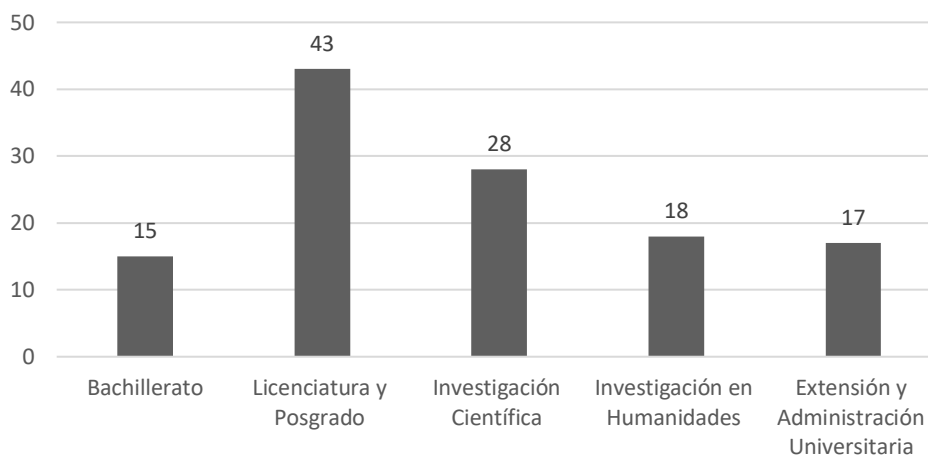
Para 2005 se decidió utilizar la nueva versión ALEPH 500 la cual hizo necesario la actualización de la infraestructura de cómputo y la adquisición de equipo con mayor capacidad, así como la transferencia de datos entre ALEPH 300 y ALEPH 500. Este proyecto implicó un impacto presupuestal del orden de los 4 millones de pesos y estuvo a cargo de la DGB. (Biblioteca Universitaria, 2008) En mayo de 2008, el sistema ALEPH 500.16 era utilizado en 134 bibliotecas.

ALEPH está conformado por los módulos de catalogación, adquisiciones, préstamo interbibliotecario, administración, y circulación. Dentro de los módulos de ALEPH que son utilizados por la DGB el más usado es el módulo de circulación que permite administrar usuarios, ejemplares y los movimientos de préstamo, devolución, resello, multas y sanciones. También se dispone de un servicio vía internet en donde



los alumnos pueden revisar el estatus de sus préstamos de material siempre y cuando se tenga habilitada la función.

Un total de 121 bibliotecas cuentan con el módulo de circulación y están distribuidas de la siguiente forma:



*Figura 1.1 Número de licencias de ALEPH*

Cada servidor de ALEPH 500 debe tener instalada la licencia principal de control del sistema y se divide en licencia WEB y GUI. Desde mayo de 2008 se tienen instaladas 660 licencias GUI y 433 WEB. (Biblioteca Universitaria, 2008)

### **1.3 Koha**

Koha es un sistema de gestión de bibliotecas de código abierto, basado en una interfaz web. Es utilizado por diversas organizaciones públicas para administrar la adquisición, catalogación, circulación, control en serie, gestión de usuarios, informes. Antes de Koha no existían sistemas integrados que incluyeran todos los módulos en una sola interfaz. (Mulay, Paliwal, Iyengar, Pillai, & Rao, 2019)

La DGB se encuentra en el proceso de migración del sistema ALEPH a Koha en las bibliotecas del SIBIUNAM. Desde 2019 se tienen 8 bibliotecas funcionando con Koha y se espera que en un futuro este número aumente. (Leyva Ramírez, 2019)

## **1.4 Definición del problema**

La versión de ALEPH 500 se comenzó a utilizar por la DGB en 2005 por lo que este software lleva más de 10 años trabajando. Debido a las innovaciones tecnológicas de la última década las tecnologías que este sistema emplea se van quedando atrás por lo que hace necesario una actualización del software.

Para poder utilizar ALEPH es necesario tener instalada la licencia en cada servidor y cada cliente. El número de licencias utilizadas actualmente es de más de mil lo cual genera a la UNAM un gasto estimado de 5 millones de pesos anuales. La DGB actualiza su sistema cada 8 años aproximadamente por lo que en este lapso tendría que desembolsar unos 40 millones de pesos.<sup>1</sup>

El costo tan elevado en el uso del sistema y el surgimiento de nuevas tecnologías impulsa el desarrollo de un sistema propio que sea más moderno y económico. Por la complejidad que conlleva el realizar todo un sistema de administración de este tipo, se propone la realización de uno de sus módulos. Al ser el módulo de circulación bibliográfica el más utilizado por las bibliotecas que componen la DGB, es el candidato perfecto para ser implementado. Esto con el fin de reducir los gastos en la universidad y proveer a las bibliotecas de un sistema de última generación.

## **1.5 Objetivo**

Desarrollar una alternativa de software rápido y confiable que permita la administración de la circulación bibliográfica en el sistema bibliotecario de la UNAM, empleando tecnologías de desarrollo web de última generación y herramientas de software libre.

## **1.6 Alcances**

Los sistemas de administración de bibliotecas se encargan de realizar distintas funciones en una biblioteca, para llevar a cabo su labor cuentan con distintos módulos: circulación, OPAC, adquisición, catalogación, etc. Debido a las

---

<sup>1</sup> Documento interno Dirección General de Bibliotecas

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

limitaciones de recursos humanos y al tiempo solamente se desarrollará el módulo de circulación. Este módulo será implementado utilizando software libre con base en un modelo de cliente servidor. En el desarrollo de la interfaz web se usará el framework de JavaScript React, mientras que los datos estarán almacenados en una base de datos PostgreSQL.

## Capítulo II. Marco teórico

### 2.1 Ingeniería de software

La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales. (Pressman, 2010).

La Ingeniería de software es una tecnología con diversas capas (figura 2.1).



*Figura 2.1 Capas de la Ingeniería de Software (Pressman, 2010)*

Cualquier enfoque que se le dé a la ingeniería debería de establecer un compromiso con la calidad. El uso de las filosofías de la calidad como Six Sigma o la gestión de la calidad total, fomentan el desarrollo de la mejora continua lo que genera enfoques más eficaces de la ingeniería de software.

El proceso es la unión entre las capas de tecnología que permite un buen desarrollo de la ingeniería de software. Define una estructura que se debe seguir para obtener nueva tecnología, forma la base para controlar la administración de proyectos de software, y establece distintas reglas.

Los métodos proporcionan los aspectos técnicos para elaborar software. En ellos se incluyen las tareas a realizar tales como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y mantenimiento. Se basan en un conjunto de principios que rigen cada área de la tecnología.

Las herramientas proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos.

### 2.1.1 Modelos del ciclo de vida

Los modelos surgieron de la necesidad de organizar un proyecto de software ya que antes de su existencia se gastaban demasiados recursos en el desarrollo y el planteamiento de requisitos era mal planificado por lo que se tenía que repetir este paso una y otra vez según fuese necesario.

#### 2.1.1.1 Modelo en cascada

También llamado ciclo de vida clásico fue el primer modelo desarrollado y es usado como base para muchos modelos, utiliza un modelo secuencial para el desarrollo que comprende las siguientes etapas:

- Especificación de requisitos: definición de las necesidades, objetivos o problemas que el sistema debe de solucionar.
- Diseño: establece la forma en que se va a construir el sistema y fijando la forma en que se va a representar la información.
- Implementación: pasar de un diseño conceptual a código entendible por una máquina.
- Pruebas: realización de pruebas con el fin de verificar el cumplimiento de los requisitos y encontrar posibles errores.
- Mejoramiento: mejoras sobre un producto funcional en manos del usuario.



Figura 2.2 Modelo en cascada (Granados, 2014)

Este modelo es útil cuando se tienen desarrollos no tan complejos y requisitos bien definidos. El trabajo desde la etapa de análisis hasta el mantenimiento fluye en

forma lineal por lo que es utilizado cuando se deben hacer adaptaciones o mejoras de sistemas ya existentes.

### 2.1.1.2 Modelo iterativo

El modelo iterativo consiste en realizar varias iteraciones sobre el modelo en cascada, teniendo que generar una versión por cada iteración.

Se caracteriza por permitir el desarrollo de versiones cada vez más completas.

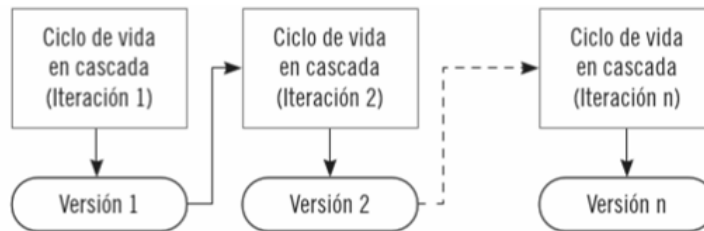


Figura 2.3 Modelo iterativo (Granados, 2014)

Este modelo es útil cuando no se tienen suficientemente claros los requisitos o estos cambian constantemente ya que a veces es necesario realizar versiones limitadas debido a la presión de la competencia.

### 2.1.1.3 Modelo incremental

Este modelo aplica el modelo secuencial de manera escalonada, pero en cada ciclo completo se le añade nuevas funcionalidades en forma de módulos llamados incrementos. Al usar el modelo incremental, el primer incremento aborda los requerimientos básicos sin incluir características suplementarias. El usuario evalúa el producto y con base a la evaluación se plantea el desarrollo del siguiente incremento.

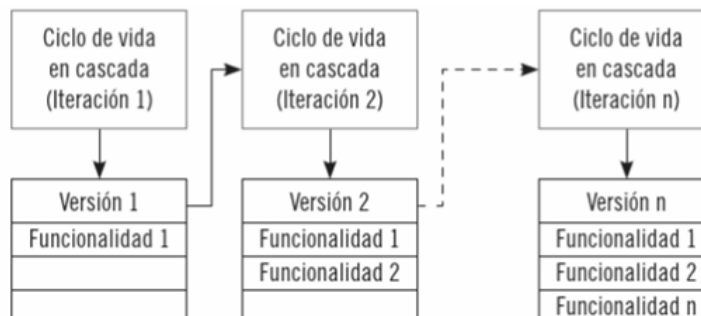


Figura 2.4 Modelo incremental (Granados, 2014)

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

Este modelo es utilizado cuando un sistema no requiera toda su funcionalidad de manera inicial y sean necesarias entregas rápidas, o cuando los requerimientos iniciales están bien definidos pero el objetivo es tan grande que hace imposible el uso de un modelo secuencial.

#### 2.1.1.4 Modelo en V

Este modelo es una variante del modelo de cascada, pero en este caso se añaden las etapas de verificación y validación entre las etapas de especificación y requisitos, y entre diseño y pruebas respectivamente. A medida que el desarrollo avanza desde el lado izquierdo, los requerimientos básicos se vuelven más detallados y una vez que se realiza la implementación, se sube por el lado derecho y se ejecutan ciertas pruebas para validar lo que se realizó cuando se iba hacia abajo.

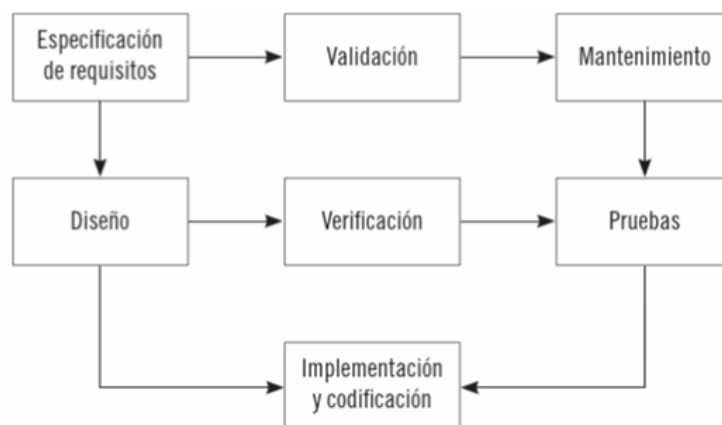


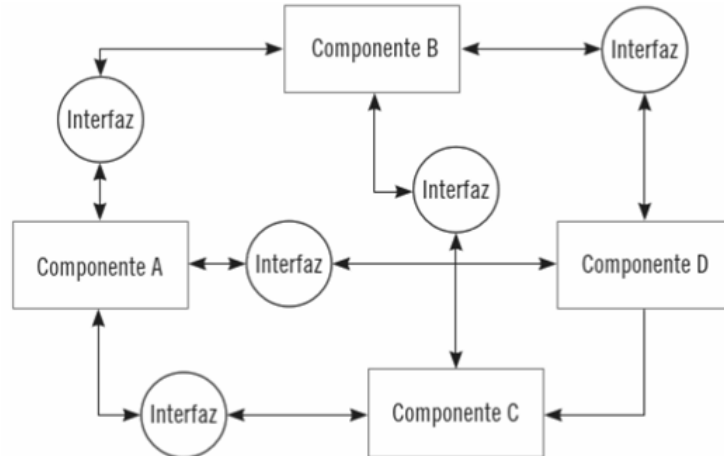
Figura 2.5 Modelo en V (Granados, 2014)

Este modelo es útil para desarrollos que requieran robustez y confiabilidad, como por ejemplo un sistema que realiza operaciones constantes sobre una base de datos.

#### 2.1.1.5 Modelo basado en componentes

Se construye a partir de componentes de software comercial ya existentes, estos añaden nuevas funcionalidades y se integran al sistema por medio de una interfaz bien definida.

Tiene características del modelo iterativo y su desarrollo comienza con elegir los componentes candidatos, en forma de módulos, clases o paquetes. Con este modelo se puede reducir bastante el tiempo de desarrollo.



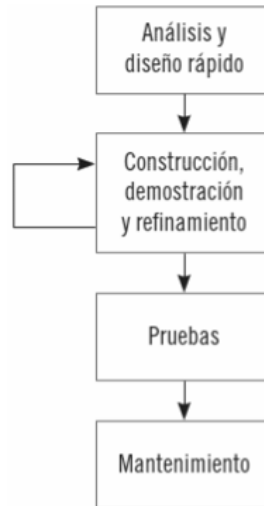
*Figura 2.6 Modelo basado en componentes (Granados, 2014)*

Se aplica en desarrollos que deseen incorporar componentes de tipo comercial siempre y cuando estén dispuestos a hacer el gasto.

#### *2.1.1.6 Desarrollo rápido de aplicaciones*

Con este modelo se tienen que desarrollar constantemente varios prototipos en intervalos cortos de tiempo con el fin de ir mejorando cada uno. Se basa en el modelo en cascada, pero fusiona las etapas de análisis y diseño para reducir el tiempo de desarrollo, esto hace necesario que el cliente se tenga que involucrar lo que puede hacer que los requisitos cambien constantemente.





*Figura 2.7 Modelo RAD (Granados, 2014)*

Es útil cuando se necesitan desarrollos rápidos, siempre y cuando se disponga de un equipo capaz de cumplir los plazos.

### **2.1.2 Análisis de requisitos**

Los requisitos son necesidades que tiene un usuario para resolver problemas. Se pueden clasificar en nivel de especificación según en su nivel de detalle:

- De usuario: son requisitos en lenguaje natural, es decir que pueda entender cualquier ser humano. Por ejemplo, se necesita que se puedan guardar imágenes.
- De sistema: consiste en una descripción completa de los requisitos de usuario. Por ejemplo, el usuario puede elegir que imagen guardar, el usuario puede nombrar las imágenes, el usuario puede escoger el formato de imagen y cualquier otro que se requiera.

A su vez los requisitos de sistema se pueden dividir según la naturaleza de estos:

- Funcionales: Se especifican funciones, comportamiento e interacción con el entorno que debe o no debe tener el sistema. Por ejemplo, el usuario debe darse de alta para usar el sistema, el usuario no puede tomar más de 2 libros a préstamo, el usuario puede ver su historial de préstamo, etc.
- No funcionales: Se refiere a las restricciones del sistema como los estándares que debe seguir, la interfaz, seguridad, etc. Por ejemplo, el

sistema debe funcionar en cualquier navegador web, debe funcionar con una base de datos MySQL, entre otros.

Una vez definidos los requisitos es necesario definir un modelo de los mismos, con el objetivo de delimitar el alcance del sistema y capturar su funcionalidad.

Entre los modelos que pueden ser usados para el análisis de requisitos tenemos el diagrama de flujo y el diagrama de caso de uso. El diagrama de flujo representa la información desde que entra hasta que sale mostrando como se van transformando los datos al moverse dentro del sistema, mientras que el diagrama de caso de uso muestra la relación entre el caso de uso y los usuarios, representando una funcionalidad básica.

### **2.1.3 Diseño**

El diseño del sistema se realiza a partir de los requisitos definidos anteriormente. El objetivo del diseño es obtener un modelo de la entidad que se quiere construir, para ello se tiene que dividir el modelo en tres partes: modelo de arquitectura, de componentes y de interfaz.

#### *2.1.3.1 Modelo de arquitectura*

Esta parte del diseño define la estructura de los componentes del sistema, propiedades y relación entre ellos. Existen varios tipos:

- Centrada en datos: Todos los componentes del sistema acceden a un almacén central de datos.
- De flujo de datos: Los datos se van transformando conforme pasan por los componentes. En caso de que solo tengamos un flujo de datos se le llama lote secuencial.
- De programa principal-subprograma: El programa principal invoca subprogramas que a su vez invocan a otros subprogramas.
- Orientado a objetos: El programa principal se apoya en objetos bien definidos que interactúan por medio de una interfaz bien definida.

#### *2.1.3.2 Modelo de componentes*

Un componente es una parte modular que puede quitarse y ponerse dentro de un sistema. Sus características están dadas dependiendo la orientación que se le da al sistema:

- **Visión orientada a objetos:** Aquí los componentes implementan una entidad llamada clase la cual consta de atributos que almacenan datos y métodos que realizan funciones. A los métodos que interactúan con el exterior se les llama interfaces.
- **Visión tradicional:** El componente implementa las funcionalidades y las interfaces de entrada y salida dentro de él. Se diferencia del anterior porque el otro puede incluir todo eso dentro de la clase.

#### *2.1.3.3 Modelo de interfaz*

La interfaz define la forma que nuestro sistema tiene que interactuar con el usuario por lo que una mala interfaz puede producir una mala experiencia en el usuario.

Para producir un buen diseño de interfaz se tienen que seguir tres reglas de importancia:

1. **Dejar el control al usuario:** sin transiciones innecesarias, posibilidad de interrupción, de vuelta atrás, de personalización, etc.
2. **Reducir la carga de memoria del usuario:** atajos intuitivos, organización jerárquica de la interfaz, etc.
3. **Hacer que la interfaz sea consistente:** presentación y obtención de la información según unas reglas de diseño. La entrada de datos se debe dar de forma que el usuario siempre sepa dónde está, cómo puede volver y qué debe hacer para continuar.

El modelado de interfaces de apoya en los casos de uso, para ello se tiene que identificar al usuario, identificar las acciones y elementos que tienen que ver con este y convertir estos objetos en elementos visuales que realicen dicha acción.

#### **2.1.4 Implementación**

En esta etapa se tiene que generar una implementación de acuerdo con la arquitectura, las especificaciones y entidades del sistema mediante el uso de algún lenguaje de programación. Para esto tenemos dos técnicas de desarrollo de software:

- Programación imperativa: se declaran los pasos a seguir mediante un conjunto de instrucciones.
- Programación declarativa: Se describen los métodos y el resultado que se desea, pero no se indica el modo.

La mayoría de los lenguajes de programación son imperativos, a su vez la programación imperativa se puede dividir en varios tipos, por ejemplo:

- Programación estructurada: nos indica que cualquier programa puede ser representado mediante tres estructuras de control básicas que son condición, secuencia e iteración.
- Programación orientada a objetos: se basa en la programación estructurada y nos indica que tenemos que usar objetos para representar conceptos del mundo real.

#### **2.1.5 Pruebas**

Las pruebas de software tienen el objetivo de verificar que se haya obtenido un software de calidad, sin errores y que cumpla con los requerimientos.

Entre las distintas pruebas que se pueden realizar sobre un sistema tenemos las siguientes:

- Unitarias: son pruebas que verifican el funcionamiento de una pieza del sistema. Es decir, módulos, componentes subprogramas, etc.
- De integración: son pruebas que se ejecutan sobre un grupo de módulos para verificar que pueden trabajar unos con otros.
- De sistema: son pruebas que se ejecutan sobre el sistema completo y su misión es verificar aspectos globales tales como la seguridad, velocidad, etc.

## **2.2 Aplicaciones web**

Una aplicación web es una herramienta informática alojada en un servidor a la cual se puede acceder por medio de un navegador web, siempre y cuando tengamos la dirección del servidor.

Entre las ventajas de este tipo de aplicaciones tenemos que es fácil darles mantenimiento y mantenerlas actualizadas sin necesidad de volver a distribuir el sistema a los usuarios que ya tienen acceso a él. Estas aplicaciones pueden ser ejecutadas en múltiples plataformas y no requieren de instalación, esto gracias a la portabilidad que ofrecen los navegadores web. Por último, la potencia de nuestra aplicación no depende de los recursos de nuestro dispositivo sino del servidor que la aloja.

### **2.2.1 Arquitectura cliente/servidor**

La arquitectura web son las tecnologías que se utilizan para levantar un servidor que permita a un usuario visualizar la información que necesita por medio de internet.

La arquitectura cliente/servidor funciona de la siguiente manera:

- El usuario realiza una petición a través del cliente sobre el servidor que contiene la aplicación web.
- Este servidor, por lo general, dispone de un acceso a una base de datos que contiene información necesaria para la ejecución de la aplicación.
- Después de obtener/modificar la información de la base de datos, el servidor envía la respuesta al cliente.
- El cliente renderiza la respuesta suministrada por el servidor y se la presenta al usuario.

#### *2.2.1.1 Arquitectura cliente/servidor de dos capas*

En esta arquitectura solo tenemos las capas de presentación y la lógica de la aplicación, que corresponden al cliente y al servidor respectivamente. Es utilizada cuando requerimos el manejo de pocos datos, cuando el servidor incluye a la base

de datos y cuando queremos que el mantenimiento del sistema sea el mínimo posible.

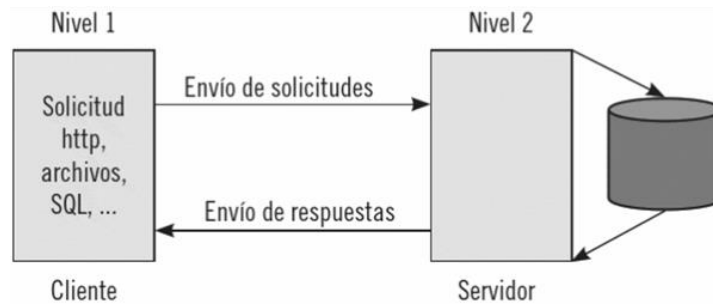


Figura 2.8 Arquitectura de dos capas (Cardador, 2014)

### 2.2.1.2 Arquitectura cliente/servidor de tres capas

En esta arquitectura incluye las capas de presentación, lógica de la aplicación y la capa de base de datos. Se utiliza cuando requerimos el procesamiento de una gran cantidad de datos en la aplicación, cuando la funcionalidad de la aplicación cambia constantemente, cuando se necesite aislar la base de datos para un fácil mantenimiento y cuando los procesos no se relacionan demasiado con los datos.

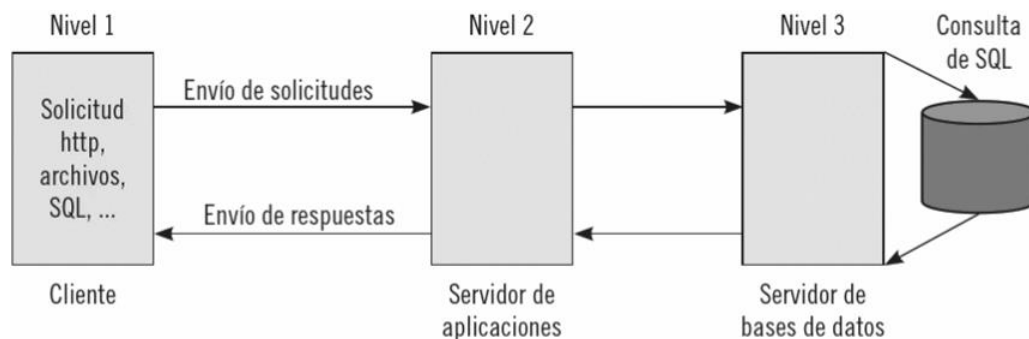


Figura 2.9 Arquitectura de tres capas (Cardador, 2014)

## 2.2.2 Páginas web

Una página web es un documento electrónico que contiene distintos tipos de información, se rige bajo los estándares de la world wide web (www) y se puede acceder a ella mediante un navegador web.

### 2.2.2.1 HTML

HTML, que significa lenguaje de marcado de hipertextos (HyperText Markup Language), es la pieza más básica para la construcción de la web y se usa para

## **Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

definir el sentido y estructura del contenido en una página web. (Mozilla Foundation, 2020)

El HTML no sirve para determinar el funcionamiento en una página web, solo nos indica la manera en cómo se ordenarán los elementos dentro de la página. En términos generales, es un conjunto de etiquetas que sirven para definir elementos de texto, imágenes, listas, videos, etc.

```
<html>
  <head>
    <title>Mi pagina</title>
  </head>
  <body>
    <h1>Hola mundo</h1>
    <p>Pagina web</p>
  </body>
</html>
```

*Figura 2.10 Código HTML de una página web básica*

Los elementos en un documento HTML se separan por medio de etiquetas, que son elementos rodeados por los símbolos "<" y ">". El nombre de los elementos puede ser escrito en mayúsculas, minúsculas o una combinación de ellos. Por ejemplo, la etiqueta <body> puede ser escrita como <body>, <BODY> o de cualquier otra combinación de mayúsculas y minúsculas.

El lenguaje HTML es muy fácil de usar ya que se puede aprender muy rápido y no se requiere tener conocimientos de programación para usarlo. Posee las ventajas de ser admitido por todos los navegadores web, archivos pequeños que no requieren demasiado espacio de almacenamiento y al presentar las etiquetas de forma estructurada, es fácil de interpretar.

### 2.2.2.2 CSS

Hojas de estilo en cascada (Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML. CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios. (Mozilla Foundation, 2020)

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

CSS sirve para definir estilos en una página web como por ejemplo fuente, color de fuente, márgenes, tamaño de imágenes, etc. Permite realizar modificaciones a los estilos sin la necesidad de modificar el código HTML y nos permite reutilizar las hojas de estilo en distintos documentos HTML.

Podemos aplicar los estilos sobre una etiqueta en específico o sobre un grupo de etiquetas, también se pueden sobrescribir las etiquetas que han sido heredadas de la etiqueta padre.

### *2.2.2.3 JavaScript*

JavaScript es un lenguaje de programación interpretado o compilado en ejecución, el cual sirve para dar funcionalidad a una página web. Es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.

Este lenguaje tiene la ventaja de que el código se ejecuta dentro del cliente, lo que se encarga de aligerar la carga de trabajo del servidor. Nos permite implementar en nuestros sitios web cosas que antes solo era posible hacer en aplicaciones de escritorio.

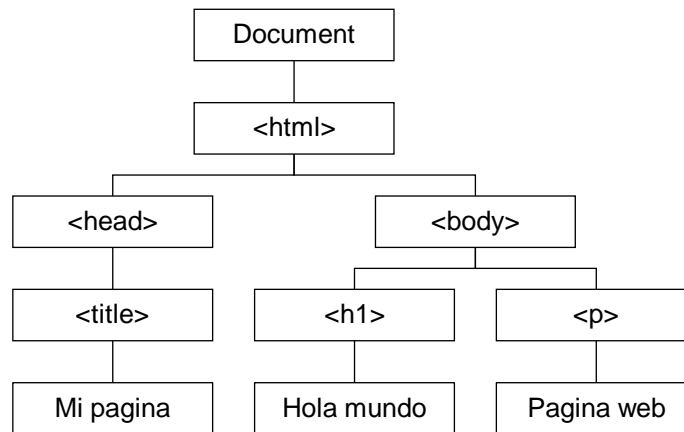
### *2.2.2.4 DOM*

El Modelo de Objetos del Documento (DOM) es un API para documentos HTML y XML. Proporciona una representación estructural del documento, permitiendo la modificación de su contenido o su presentación visual. Esencialmente, comunica las páginas web con los scripts o los lenguajes de programación. (Mozilla Foundation, 2019)

En la Figura 2.11 podemos ver la representación del DOM del documento de la Figura 2.10.

El DOM es importante a la hora de trabajar con JavaScript ya que por medio de él se puede acceder a los elementos de HTML, para así obtener o modificar sus propiedades.





*Figura 2.11 Representación del DOM de un documento HTML*

### 2.2.3 Frameworks de JavaScript

Los frameworks de JavaScript se han vuelto una parte importante en el desarrollo de aplicaciones web modernas, su uso otorga a los desarrolladores herramientas para la creación de aplicaciones web interactivas y escalables. Hoy en día muchas empresas los utilizan y en muchos trabajos de desarrollador se requiere experiencia en ellos.

JavaScript surgió en 1996 para dotar a las páginas web de interactividad ya que antes de su existencia un sitio web se componía solamente de documentos estáticos. Su llegada transformo a la web en un lugar donde ya no solo se podían leer información, sino que también se podía interactuar con ella. A partir de su uso, muchos desarrolladores escribieron bibliotecas que contenían herramientas que podían ser reutilizables lo que ayudo al crecimiento de la web.

Existen muchos frameworks de JavaScript, la mayoría de ellos trabajan de manera similar. Un framework es una biblioteca que ofrece conceptos con los cuales podemos construir aplicaciones interactivas y dinámicas, esto nos permite escalar y mantener el software haciendo que este sea más duradero.

El 95% de los sitios web utiliza JavaScript por lo que se ha vuelto una parte esencial de la web. (Mozilla Foundation, 2020)

### 2.2.3.1 *Angular*

Angular es un framework de código abierto utilizado para crear y mantener aplicaciones web de una sola página. Este framework tiene la capacidad de crear una aplicación de una sola página, por lo que los cambios son instantáneos, y hace que las páginas sean más rápidas. Utiliza un diseño basado en componentes reutilizables, por lo que no hay la necesidad de recargar la página constantemente.

Angular posee las siguientes características:

- Es mantenido por Google.
- Implementa el modelo vista controlador (MVC).
- Utiliza TypeScript que permite escribir código JavaScript más potente.
- Utiliza HTML y directivas para insertar la lógica.
- Utiliza binding bidireccional que permite compartir datos entre un componente y su plantilla.

Entre las empresas que utilizan Angular para el desarrollo de diversas aplicaciones se encuentran Santander, GitHub, NBA, Forbes, Google, Samsung, entre otras.

### 2.2.3.2 *Vue.js*

Vue.js es un framework de código abierto para el desarrollo de aplicaciones de una sola página e interfaces de usuario. Fue lanzado por Evan You en 2014 y utiliza el modelo vista controlador, pero principalmente se encarga del manejo de las vistas y del renderizado en el navegador.

Las características principales de Vue.js son:

- Es mantenido por su creador y la comunidad.
- Añade los componentes a medida que se requieren.
- Solamente utiliza JavaScript.
- Utiliza Templates en HTML y sintaxis JSX.

- Utiliza binding bidireccional que actualiza el modelo si la vista cambia y actualiza la vista si el modelo cambia.

El framework Vue.js ha sido utilizado para el desarrollo de diversas aplicaciones web como Nintendo, Font Awesome, Larvel, GitLab, etc.

### 2.2.3.3 *ReactJS*

ReactJS fue lanzada por Facebook en 2013, técnicamente hablando no es un framework, más bien es una librería para el renderizado de componentes en interfaces de usuario. Puede ser usado tanto para aplicaciones web como para aplicaciones móviles. Anteriormente se usaban clases para definir componentes, pero a partir de la versión 16.8 se introdujeron los Hooks los cuales permiten utilizar las mismas características que otorga el uso de clases.

Las características de ReactJS son:

- Creado y mantenido por Facebook.
- Es solamente la vista en un modelo MVC.
- Aprovecha las características de JavaScript.
- Utiliza componentes con una sintaxis JSX.
- Solamente el estado puede modificar la interfaz, pero no al revés.

Este framework es de los más utilizados para el desarrollo de aplicaciones web, tal es el caso de Airbnb, Facebook, Instagram, WhatsApp, Netflix, Dropbox, BBC, PayPal, Uber, entre muchos otros.

### 2.2.4 **Servicios web**

Los servicios web son tecnologías que se rigen por un conjunto de estándares y protocolos, su función es la de intercambiar datos entre distintas aplicaciones las cuales pueden estar escritas en distintos lenguajes de programación e incluso estar alojadas en distintos sistemas operativos.

#### 2.2.4.1 *SOAP*

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

La arquitectura de los servicios web SOAP está compuesta por tres elementos: el proveedor del servicio, el consumidor del servicio y el registro del servicio. En la Figura 2.12 podemos observar cómo interactúan los tres elementos entre sí.

El Proveedor genera el contrato de integración WSDL que permite definir la interfaz de un servicio web, en donde se describirá la especificación del servicio web. El WSDL se registra en el registro del servicio.

Cuando un consumidor del servicio solicita un servicio web, se localiza en el registro de servicio por medio del UDDI.

El consumidor envía una solicitud para un servicio particular al web service listener, que se encarga de recibir y enviar los mensajes en formato SOAP el cual describe el formato de los documentos para el intercambio de información en servicios web.

El proveedor del servicio recibe el mensaje SOAP del consumidor y ejecuta la operación relacionada con dicha solicitud. El resultado de esta se devuelve como mensaje SOAP al consumidor.

El consumidor recibe el SOAP con la respuesta y/o resultado de la operación y lo procesa.



Figura 2.12 Arquitectura de SOAP (Pascual, 2019)

#### 2.2.4.1 REST

La arquitectura de servicios web REST puede ser utilizada por cualquier cliente HTTP, es más simple que otras arquitecturas, como por ejemplo SOAP. Emplea

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

hipermedios como XML o JSON lo que le permite una gran escalabilidad. En la Figura 2.13 podemos ver cómo funciona REST, se compone de dos partes: un cliente que puede ser una computadora o hasta un smartphone y un servidor que normalmente se conecta a una base de datos.

El cliente envía una petición HTTP al servidor mediante la URI (Identificador Único del Recurso) que es una dirección que nos indica sobre que recurso se va a realizar la operación.

Para poder interactuar con los recursos se requiere de distintos métodos HTTP:

- GET: se utiliza para acceder a los distintos recursos.
- POST: se utiliza para crear nuevos recursos.
- PUT: se utiliza para modificar los recursos existentes.
- DELETE: se utiliza para eliminar recursos.

El servidor recibe la petición y dependiendo la URI y el método HTTP envía una respuesta de hipermedios que generalmente es JSON.

El cliente recibe la respuesta y lo procesa.

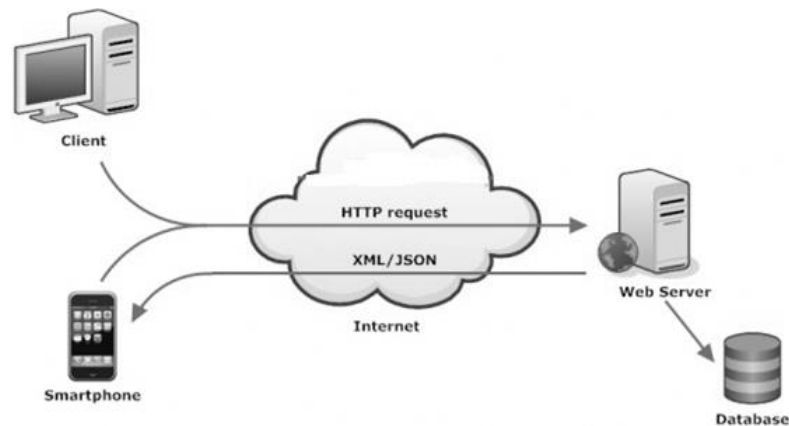


Figura 2.13 Arquitectura de REST (REST, 2015)

#### 2.2.4.2 GraphQL

GraphQL es un lenguaje de manipulación y consulta de datos de código abierto que permite comunicar al cliente con el servidor. Fue desarrollado por Facebook en 2012 y liberado en 2015.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

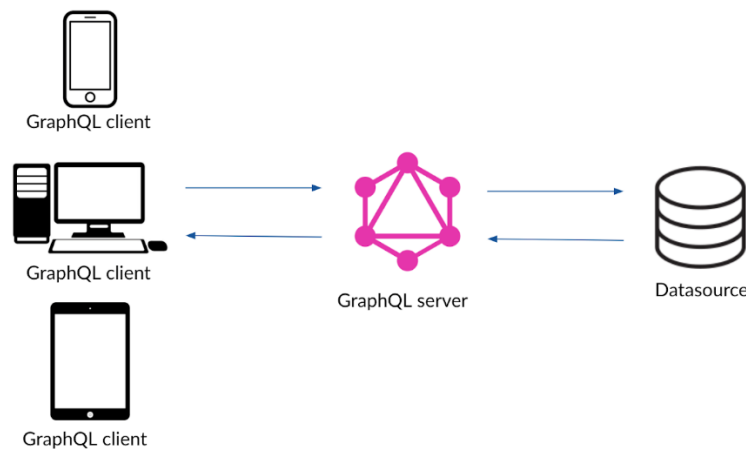
---

Consta de dos elementos, el lenguaje de consulta y manipulación de datos y del entorno de ejecución en donde se responde a las consultas haciendo uso de un esquema que enlista los datos que se pueden entregar.

Consta de dos partes, un servidor y un cliente de GraphQL. El servidor de GraphQL puede comunicarse con una base de datos o hasta con una API REST.

GraphQL resuelve dos problemas del uso REST, al realizar peticiones en REST a veces faltan datos por lo que es necesario realizar otra petición, y otras veces se realizan peticiones que traen muchos datos que son ignorados a la hora de ser procesados. Con GraphQL no se tiene este problema ya que se puede escoger que partes de cada recurso se van a solicitar.

El cliente envía la petición al servidor mediante un query de GraphQL, este lo recibe y dependiendo la consulta regresa una respuesta con el mismo formato. No depende de HTTP por lo que no utiliza métodos ni respuestas de ese tipo.



*Figura 2.14 Arquitectura de GraphQL (Drasner, 2020)*

## 2.3 Bases de datos

Aunque no las veamos, las bases de datos están en prácticamente todos los sistemas. Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. (Marqués, 2011).

Las bases de datos se utilizan principalmente en sistemas de información los cuales

sirven para administrar procesos mediante el tratamiento de los datos. Los datos por sí solos no nos dicen mucho, estos se transforman en información cuando son aplicados a propósitos específicos.

### **2.3.1 Características de las bases de datos**

#### *2.3.1.1 Redundancia Mínima*

La redundancia es cuando la información que se tiene almacenada contiene datos repetidos o duplicados, también se le puede llamar redundancia cuando se tienen datos que pueden ser derivados de otros. La redundancia afecta la integridad y la consistencia de los datos, da como resultado la reducción de capacidad del almacenamiento ya que al almacenar el mismo dato dos veces, la capacidad se va a gastar el doble de rápido.

A veces es necesario la redundancia para almacenar relaciones con otros datos por lo que no siempre se puede eliminar por completo, pero siempre debe mantenerse al mínimo posible.

#### *2.3.1.2 Integridad*

La integridad en una base de datos es la exactitud y fiabilidad de los datos almacenados, es decir, que los datos están completos, que sean validos de acuerdo con el tipo de dato establecido y que sean exactos.

Por ejemplo, un usuario no debería de ser capaz de ingresar un dato de tipo fecha en un campo establecido como numérico, o un usuario debe de ingresar siempre un valor en un campo marcado como obligatorio.

Para mantener la integridad en una base de datos es necesario establecer ciertas reglas: definición de claves primarias únicas y obligatorias, de validación de datos, de integridad referencial, de restricción al borrado, etc.

#### *2.3.1.3 Consistencia*

La consistencia se refiere a la coherencia entre los datos de una base de datos, al perderse la integridad se pierda también la consistencia ya que son conceptos similares. La redundancia también afecta la consistencia, al realizar una actualización sobre un dato duplicado, también se debe de actualizar cada una de las copias para mantener la consistencia.

#### *2.3.1.4 Seguridad*

La seguridad en una base de datos tiene como objetivo proteger los datos de accesos no autorizados, modificación o destrucción en los datos o en la estructura de estos.

Para garantizar la seguridad en una base de datos se deben aplicar medidas de seguridad en distintos niveles.

- Sistema de base de datos
- Sistema operativo
- Red
- Acceso Físico
- Usuarios

#### **2.3.2 Lenguaje SQL**

El lenguaje estructurado de consultas (SQL), fue desarrollado por IBM a principios de los 70's bajo el nombre de SEQUEL con el fin de manejar los datos de su sistema de base de datos. El primer estándar para SQL fue publicado en 1986 por ANSI y comenzó a ser adoptado por diversos desarrolladores de sistemas de base de datos.

Aunque SQL se considera un sublenguaje debido a su naturaleza de no procesamiento, aun así, es un lenguaje completo que le permite crear y mantener objetos en una base de datos, asegurar esos objetos y manipular la información dentro de los objetos. (Oppel & Sheldon, 2010)

Las funciones de SQL se dividen en tres grupos de acuerdo con las funciones que realizan: DDL, DML y DCL.



#### *2.3.2.1 DDL*

El lenguaje de definición de datos (Data Definition Language) se utiliza para realizar operaciones como crear, modificar o borrar objetos de la base de datos los cuales pueden ser tablas, vistas, esquemas, procedimientos entre otros.

Las instrucciones que se encuentran el DDL de manera frecuente son CREATE, ALTER, RENAME y DROP.

#### *2.3.2.2 DML*

El lenguaje de manipulación de datos (Data Manipulation Language) nos permite realizar operaciones sobre los datos almacenados tales como agregar, recuperar, modificar o borrar datos en los objetos de una base de datos.

Las instrucciones que pertenecen al lenguaje DML son SELECT, INSERT, DELETE y UPDATE.

#### *2.3.2.3 DCL*

El lenguaje de control de datos (Data Control Language) tiene como objetivo controlar quien tiene acceso a realizar que operaciones sobre los objetos de la base de datos. Estas restricciones pueden ser de definición o de manipulación de datos.

Las instrucciones de DCL son GRANT y REVOKE, utilizamos GRANT para otorgar un permiso y REVOKE para quitarlo.

### **2.3.3 Sistema gestor de base de datos**

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. (Marqués, 2011).

Un SGBD tiene como objetivo principal el proveer una forma de almacenar y recuperar la información dentro de una base de datos. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la

## Error! Use the Home tab to apply Título 1 to the text that you want to appear here.

provisión de mecanismos para la manipulación de la información. (Silberschatz, Korth, & Sudarshan, 2002)

### 2.3.3.1 RDBMS

Las bases de datos relacionales son un tipo de base de datos que se basan en el modelo relacional. Para poder mantener estas bases de datos es necesario de un sistema de gestión de base de datos relacional (RDBMS), el cual utiliza SQL para realizar operaciones de consulta y mantenimiento sobre los objetos y datos de la base de datos.

Los RDBMS utilizan una arquitectura en tres niveles cuyo propósito es separar el modelo del usuario de la estructura física de la base de datos.

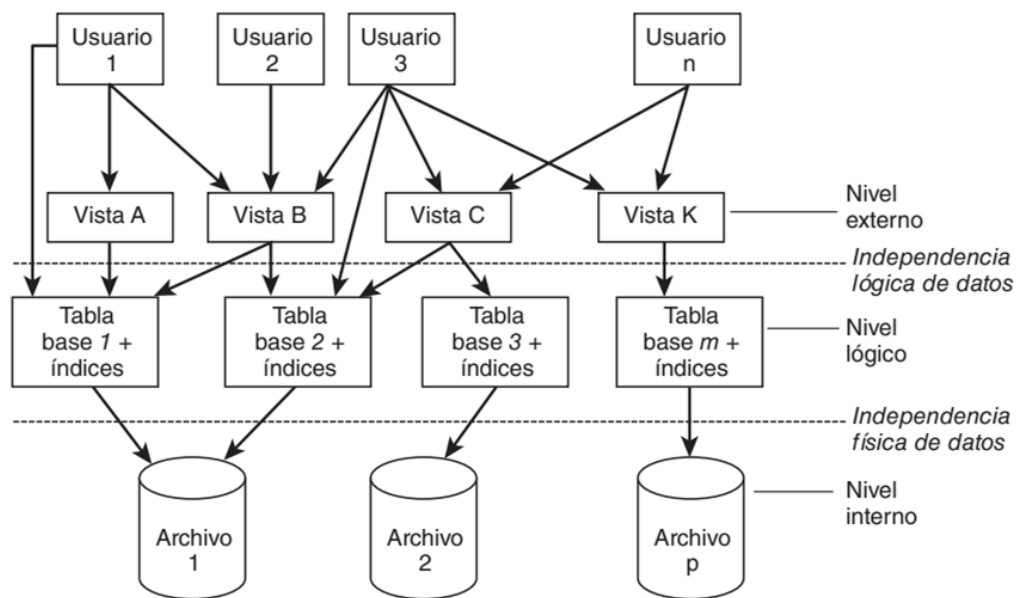


Figura 2.15 Arquitectura en tres niveles (Ricardo, 2009)

- Nivel externo: En este nivel los usuarios manejan la base de datos a través de un conjunto de vistas o modelos externos en donde no es necesario conocer la estructura de la base de datos.
- Nivel lógico: Este nivel incluye la estructura de la información de la base de datos e incluye una descripción de los datos disponibles.

- Nivel interno: Se refiere a la implementación física de la base de datos e incluye las estructuras necesarias para almacenar los datos en forma de archivos dentro de un dispositivo de almacenamiento.

### **2.3.4 Modelos de datos**

El modelo de datos es una colección de herramientas conceptuales que utilizamos para describir los datos, las relaciones, la semántica y las restricciones de los datos.

#### *2.3.4.1 Modelo Entidad Relación*

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es un objeto del mundo real que se puede diferenciar de otros objetos y son descritos dentro de una base de datos por medio de un conjunto de atributos. Por ejemplo, para describir los datos de los clientes de un banco podemos definir una entidad cliente con un conjunto de atributos nombre-cliente, calle-cliente y ciudad-cliente.

Una relación es una asociación entre varias entidades. En el mismo ejemplo del banco tenemos otra entidad cuenta con los atributos número de cuenta y saldo, estos dos están relacionados mediante la relación impositor que es una persona que deposita dinero en una cuenta.

Además de las entidades y las relaciones, el diagrama ER también puede representar el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones a lo cual llamamos cardinalidad.

Una estructura de una base de datos puede ser representada mediante un diagrama entidad relación, el cual consta de lo siguiente:

- Rectángulos: representan a las entidades.
- Óvalos: representan atributos.
- Diamantes: representan relaciones entre conjuntos de entidades.
- Líneas: unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

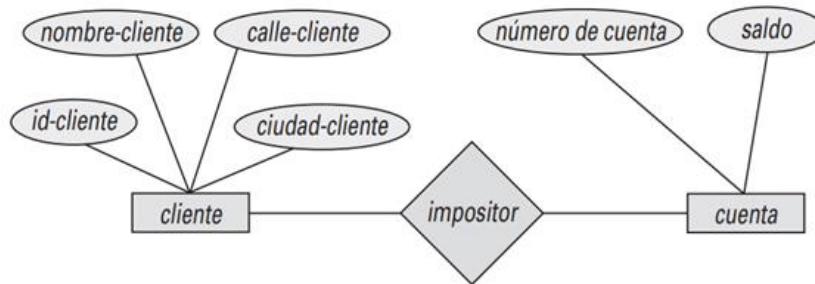


Figura 2.16 Modelo Entidad-Relación (Silberschatz, Korth, & Sudarshan, 2002)

### 2.3.4.2 Modelo relacional

En el modelo relacional se representan los datos y sus relaciones por medio de un grupo de tablas que se componen por columnas identificadas por un nombre único.

Utilizando el ejemplo anterior, la primera tabla representa la entidad cliente, la segunda representa la entidad cuenta, y la última representa la relación que hay entre las dos tablas.

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsaín
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

(a) La tabla *cliente*

<i>número-cuenta</i>	<i>saldo</i>
C-101	500
C-215	700
C-102	400
C-305	350
C-201	900
C-217	750
C-222	700

(b) La tabla *cuenta*

<i>id-cliente</i>	<i>número-cuenta</i>
19.283.746	C-101
19.283.746	C-201
01.928.374	C-215
67.789.901	C-102
18.273.609	C-305
32.112.312	C-217
33.666.999	C-222
01.928.374	C-201

(b) La tabla *impositor*

Figura 2.17 Modelo Relacional (Silberschatz, Korth, & Sudarshan, 2002)

Este modelo necesita de datos para poder representarlas relaciones, es muy útil a la hora de realizar la normalización sobre la base de datos.

El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R. Los diseños de bases de datos a menudo se realizan en el modelo E-R,

y después se traducen al modelo relacional. (Silberschatz, Korth, & Sudarshan, 2002)

### **2.3.6 Normalización de bases de datos**

La normalización es una técnica empleada para producir conjuntos de relaciones con propiedades específicas que permiten reducir la redundancia de los datos y mantener su integridad. La normalización se define por un conjunto de normas, que se conocen como formas normales, que proporcionan una directriz específica de cómo los datos son organizados para evitar anomalías que den lugar a inconsistencias y pérdida de los datos tal como se mantienen almacenados en la base de datos. (Opper & Sheldon, 2010)

Fue desarrollada en 1972 por E. F. Codd y en un inicio incluía tres formas normales.

- Primera forma normal: La primera forma normal (1NF) se encarga de eliminar grupos de datos repetidos e innecesarios.
- Segunda forma normal: La segunda forma normal (2NF) elimina los campos que son independientes de la clave primaria.
- Tercera forma normal: La tercera forma normal (3NF) se encarga de eliminar todos los atributos independientes de una entidad en segunda forma.

## **2.4 Sistema integrado de gestión bibliotecaria**

Un sistema integrado de gestión bibliotecaria (SIGB), es una herramienta de software que permite automatizar las funciones que más comúnmente se realizan en una biblioteca, para lo que emplea una base de datos donde almacena los registros utilizados por cada una de sus funciones.

Los SIGB tiene como objetivo ayudar a los bibliotecarios a realizar tareas como la búsqueda de usuarios, catalogación de material bibliográfico, préstamos, informes entre otras. Al tener los datos en una base de datos se evita la redundancia de información.

### **2.4.1 Módulos de un SIGB**

Un SIGB cuenta con varios módulos que se relacionan entre sí, cada módulo realiza tareas específicas por lo que trabajan sobre una sola base de datos en donde cada módulo pide solo los datos que necesita.

#### *2.4.1.1 Administración*

El módulo de administración se utiliza para configurar el sistema y adaptarlo a las necesidades de la organización. Entre las funciones que realiza este módulo se encuentran la incorporación de bibliotecas o de grupos de bibliotecas, administración de usuarios o grupos de usuarios, control de permisos a los usuarios, políticas de circulación, y carga y descarga de registros.

#### *2.4.1.2 Adquisición*

El módulo de adquisición se utiliza para administrar el presupuesto para la adquisición de material bibliográfico. Las funciones que desempeña son control de pagos y facturas, administración de solicitudes de adquisición, administración de control de órdenes de compra, recepción y carga de materiales adquiridos, control de reclamaciones, administración de suscripciones y administración de proveedores.

#### *2.4.1.3 Circulación*

El módulo de circulación realiza las funciones de administración sobre los usuarios y la información bibliográfica. Sus funciones más importantes son préstamo de material a domicilio, préstamo interbibliotecario, control del préstamo en sala, consultas de materiales en préstamo, reservas, renovaciones y devoluciones, administración de multas y suspensión del préstamo.

#### *2.4.1.4 Catalogación*

El módulo de catalogación tiene como función el análisis y la creación de los registros del material bibliográfico. Sus funciones son recuperación de la

información, control de autoridades y la lista de términos controlados, control de la información de los fondos de la biblioteca.

#### *2.4.1.5 OPAC*

El módulo de catálogo público de acceso en línea (OPAC) muestra el contenido del catálogo y permite a los usuarios acceder e interactuar con la información que contiene. En él se realizan búsquedas sencillas o avanzadas, se realizan renovaciones y reserva de materiales, se muestra el historial de búsquedas, se pueden generar bibliografías a partir de registros marcados, entre otras cosas.

#### *2.4.1.6 Publicaciones seriadas*

El módulo de publicaciones seriadas la información bibliográfica del catálogo. Sus principales funciones son la creación de registros de publicaciones periódicas, control de llegadas, de existencias, reclamación de ejemplares, gestión de proveedores, fondos presupuestarios y control de reclamaciones

## Capítulo III. Análisis de requerimientos

### 3.1 Requisitos del sistema

#### 3.1.1 Requisitos funcionales

Los requisitos funcionales con los que deberá contar el sistema son los siguientes:

- Se deberá acceder al sistema por medio de un usuario y contraseña.
- Si el usuario y/o contraseña no coinciden con los registros, se deberá negar el acceso al sistema y mostrar un mensaje de error.
- Si el usuario y contraseña son correctos se deberá redireccionar a la página principal del sistema.
- Si el usuario intenta acceder a cualquier sección del sistema sin antes haber introducido sus credenciales, se le deberá redireccionar a la página de inicio de sesión.
- Cada usuario tendrá un número de identificación propio (en el caso de la UNAM puede ser el número de cuenta) el cual deberá ser único.
- No se deberá poder crear dos usuarios con el mismo número de identificación o bien modificar el número de algún usuario existente para que coincida con otro, en este caso se deberá mostrar un error.
- El usuario deberá de ser capaz de buscar, crear, editar y eliminar registros de usuarios.
- El usuario deberá ser capaz de observar los detalles de algún usuario.
- Los detalles de usuario deberán contener la información completa del usuario y el historial de multas y préstamos.
- El usuario deberá de poder pagar total o parcialmente las multas.
- El usuario deberá de poder crear multas de manera manual.
- El usuario deberá de ser capaz de buscar, crear, editar y eliminar registros bibliográficos, así como visualizar, crear, editar y eliminar los ítems asociados a ese registro bibliográfico.
- El usuario deberá de ser capaz de realizar, devolver y renovar préstamos.



- Si el usuario al que se le va a prestar o renovar un préstamo tiene algún bloqueo, multa o supera la cantidad de préstamos permitida según las políticas de préstamo se le deberá negar la transacción.
- El usuario deberá de ser capaz de definir las políticas de préstamo según su conveniencia.
- El sistema deberá de generar multas a un usuario de manera automática cuando un ítem es devuelto de manera atrasada, según las políticas de préstamo establecidas.
- El sistema deberá bloquear los préstamos a un usuario de manera automática cuando un ítem es devuelto de manera atrasada, según las políticas de préstamo establecidas.
- El sistema deberá mostrar mensajes de error cuando una transacción sea rechazada.

### **3.1.2 Requisitos no funcionales**

Los principales requisitos no funcionales con los que deberá de contar el sistema son los siguientes:

- Deberá funcionar en cualquier navegador web.
- El sistema deberá de almacenar registros de manera persistente haciendo uso de una base de datos.
- Deberá funcionar con un manejador de base de datos relacionales de software libre como MySQL o PostgreSQL.
- El sistema deberá de poder procesar 5000 transacciones por día.
- El sistema deberá de tener funcionalidad total en menos de 5 segundos.
- El sistema deberá de ser capaz de trabajar con hasta 20 usuarios concurrentes.
- Los permisos de acceso al sistema deberán de ser modificados solo por los administradores.
- El sistema deberá de ser capaz de almacenar registros bibliográficos en formato MARC (XML o JSON).

- El sistema deberá de estar alojado en un servidor web se software libre.

### 3.1.3 Requerimientos mínimos del sistema

#### 3.1.3.1 Cliente

El cliente funciona con el framework de JavaScript ReactJS en su versión 16.8, esta es la versión mínima para utilizar los React Hooks que son parte fundamental del cliente. El navegador deberá de soportar ECMA Script 5. Las versiones mínimas de navegador que se necesitan para el funcionamiento del sistema son:

- Google Chrome 49
- Firefox 50
- Safari 10
- Internet Explorer 9
- Edge 14

#### 3.1.3.2 Servidor

Según la escuela de negocios EAE Business School, en los últimos años se ha catalogado a Microsoft Windows, GNU/Linux y Mac OS como los sistemas operativos más populares y eficaces en el mundo. Debido a lo anterior hemos desarrollado un sistema capaz de funcionar en estos tres sistemas.

Para que el servidor funcione correctamente se necesita como mínimo Node v8.17.0 y PostgreSQL v9.4. La versión requerida de Node es la última versión que salió para SO Linux de 32 bits mientras que la versión requerida de PostgreSQL es la mínima necesaria para implementar las funcionalidades que se requieren.

En cuanto a las dependencias del sistema las versiones necesarias son las siguientes:

- Apollo Server v2.19.0.
- Data Loader v2.0.0.
- Dotenv v8.2.0.
- GraphQL v15.4.0.

- JSON Web Token v8.5.1.
- Pg Promise v10.7.3.

De acuerdo con los requerimientos del servidor se puede concluir que el sistema puede funcionar en sistemas operativos Linux y Windows de 32 y 64 bits mientras que en el sistema operativo MacOS se necesita sea de 64 bits.

Con un procesador Intel Core i3 de 2.0 GHz se consume un 12% del CPU para el servidor de Node y un 5.8% para el servidor de PostgreSQL por lo que tenemos un consumo total de 17.8%. De acuerdo con lo anterior se recomienda utilizar un procesador con una velocidad de 2 GHz como mínimo.

El consumo de memoria RAM del servidor Node y del servidor PostgreSQL es de 271 MB por cada usuario. Si el sistema será utilizado por 20 usuarios de manera concurrente, se recomienda como mínimo contar con una memoria RAM de 8GB.

Según el portal de estadística UNAM del ciclo 2019-2020, se tuvieron 360,883 alumnos en total y 41,332 académicos, lo que da un total de 402,215 de personas que integran la comunidad de la UNAM. Cada miembro ocupa aproximadamente 1MB en memoria ya que se almacena la foto de cada miembro con un tamaño máximo de 1MB y almacenar un registro en la base de datos ocupa casi 1KB. En conclusión, necesitamos como mínimo un disco de 500GB de capacidad.

### **3.2 Análisis del módulo de circulación**

El módulo de circulación en un SIGB debe cumplir con ciertas características como mínimo para brindar un buen servicio a los usuarios de la biblioteca.

La primera característica que se debe incluir es un módulo de acceso mediante usuario y contraseña para que solo los administradores del sistema puedan acceder y así evitar que se ejecuten operaciones de lectura y escritura por usuarios no permitidos. Esto implica que también se tenga una categoría para usuarios comunes y para administradores del sistema.

En cuanto al manejo de la circulación bibliográfica, como mínimo se debería de poder realizar acciones de préstamo, devolución, y renovación del material

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

bibliográfico. Para ello se necesita contar con el registro de los usuarios y del inventario. El inventario forma parte del módulo de catalogación, pero si se tiene solo un módulo de circulación es indispensable que este pueda ser manejado por el módulo de circulación. Se debe de poder realizar operaciones de lectura, creación, edición y borrado (CRUD) sobre los usuarios y el inventario. Se debe de tener una lista reducida de estos datos y una sección en donde se pueda ver a detalle cada elemento. Para los usuarios se debe de ver el historial de préstamos y de multas y para el inventario se deben de ver los ejemplares que hay en cada registro.

Otra característica de un módulo de este tipo es que se pueda detectar cuando un usuario devuelve un ejemplar fuera de tiempo y a causa de esto generar una multa ya sea económica o de restricción de servicio por un periodo de tiempo. En caso de que el usuario regrese un ítem dañado o que lo haya extraviado es necesario poder generar multas de manera manual las cuales pueden variar dependiendo de que ejemplar fue dañado. Estas multas manuales también sirven para registrar algunas otras tarifas como por ejemplo sacar copias dentro de la biblioteca, usar las computadoras o algún otro servicio que ofrezca la biblioteca.

Al llevar un control de las multas o tarifas, se deben incluir características que permitan transacciones monetarias como pagar total o parcialmente las multas, cancelarlas, perdonarlas o transferirlas.

Las sanciones que se aplican cuando un ítem es devuelto fuera de están determinadas por las políticas de circulación, las cuales pueden variar dependiendo la biblioteca, el grupo al que pertenece el usuario, el tipo de ítem o simplemente se puede aplicar la misma política para todos. Estas nos indican la cantidad de préstamos que se pueden hacer a un usuario al mismo tiempo, el periodo de préstamo, las veces que puede renovar, y si se aplica una sanción se puede indicar en cuanto tiempo se hará efectiva (periodo de gracia).

Opcionalmente se pueden agregar las características de préstamo interbibliotecario, reserva de material bibliográfico, transferencia de los préstamos a otras bibliotecas

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

y la posibilidad de que los usuarios comunes puedan revisar el status de sus préstamos.

## Capítulo IV. Diseño y desarrollo del sistema

### 4.1 Diseño de la base de datos

Para poder satisfacer las necesidades que se requieren en las bibliotecas de la UNAM, es necesario tener una base de datos muy grande, ya que el número de personas que pueden acceder al servicio se acerca a los 500,000. Para esto la base de datos necesita almacenar cada uno de los usuarios de manera separada.

Todo el sistema comprende las 134 bibliotecas por lo que es necesario poder administrar cada una de ellas de manera separada, para lo cual se le asigna una biblioteca a cada usuario. Las bibliotecas deben almacenar información básica e información de contacto (teléfono, email, etc.).

En las diferentes bibliotecas tenemos los diferentes tipos de usuarios, los cuales tienen diferentes privilegios. Para esto se necesita tener grupos de usuarios y pueden ser comunes para todas las bibliotecas o para una sola, por ejemplo, si tenemos un grupo que sea alumnos este grupo puede estar repetido en cada una de las bibliotecas. También deben de indicar si el usuario que pertenece a ese grupo podrá acceder al sistema (se dice que el usuario pertenece al staff).

Se debe entender que los usuarios son las personas a las que se les brindará el servicio, no solo las personas que pueden acceder al sistema. Estos deberán de indicar a que biblioteca y a que categoría pertenecen, así como información básica y de contacto como puede ser nombre completo, teléfono, correo, dirección, etc. También deberán de tener un historial de préstamos en donde se pueda almacenar un registro histórico de los estados de sus préstamos y un historial de multas (conocido como cuenta) en donde se pueda ver todos los pagos que tiene por hacer y que ha hecho.

Como parte del inventario se necesitan almacenar los tipos de material bibliográfico para tener un mejor control de estos, para esto se crea la entidad tipos de ítem. Los tipos de ítem deberán de indicar si el elemento estará disponible para préstamo y además indicará si el préstamo tendrá algún costo.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Los registros bibliográficos normalmente son almacenados en formato MARC, pero nosotros seleccionamos un conjunto de campos suficientes para llevar el control de estos ya que la información detallada de estos registros se maneja en los módulos de catalogación. Los campos que consideramos necesarios para los registros son el autor, título, editorial, el tipo de material, lugar de publicación, fecha de publicación y el código ISBN o ISSN según sea el caso.

Los registros bibliográficos son una entidad simbólica que representan obras bibliográficas, pero no corresponden a un elemento físico. Para esto necesitamos almacenar los ejemplares en una tabla llamada ítems. Los ítems deberán indicar a que registro bibliográfico pertenecen, en que biblioteca se encuentran y en qué lugar (piso, estante, etc.), su estatus, la fecha en que fueron adquiridos y el precio en caso de que algún usuario deba reponer algún ejemplar. Adicionalmente deberán de contar con un código único para que cuando se necesite llevar a cabo un préstamo se pueda encontrar el ítem fácilmente en el sistema.

Las políticas de circulación deberán de indicar en donde van a ser aplicadas ya que una política puede aplicarse en diferentes las bibliotecas, a diferentes grupos de usuario y en diferentes tipos de ítems, deberá indicar la cantidad de préstamos máxima, así como el periodo de préstamo, también deberá poder incluir sanciones, renovaciones máximas y periodo de gracia.

#### **4.1.1 Definición de tablas**

Las tablas quedaran definidas de la siguiente forma.

##### *4.1.1.1 Bibliotecas*

Esta es una de las tablas principales ya que tiene relación con caso la mitad de las tablas, en esta se almacenan los registros de las diferentes bibliotecas que comprenden la administración del sistema.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
UNIQUE	codigo	VARCHAR(10)	Código de la biblioteca
	nombre	VARCHAR(50)	Nombre de la biblioteca

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

URL	VARCHAR(100)	Dirección de la página web de la biblioteca
-----	--------------	---

*Tabla 4.1 Estructura bibliotecas*

#### 4.1.1.2 Grupos de usuario

Los grupos de usuario son los distintos tipos de usuarios que podemos tener en una biblioteca, como los administradores, alumnos, profesores, alumnos de posgrado, personal, etc.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
UNIQUE	codigo	VARCHAR(10)	Código del grupo de usuario
	nombre	VARCHAR(50)	Nombre del grupo de usuario
FK	biblioteca_id	UUID	Identificador de la biblioteca asociada
	staff	BOOLEAN	El grupo es de administrador o no

*Tabla 4.2 Estructura grupos de usuario*

#### 4.1.1.3 Usuarios

Esta tabla define a todo tipo de usuarios, tanto a los que manejan el sistema como a los que se les brinda el servicio de préstamo.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
UNIQUE	codigo	VARCHAR(10)	Código del usuario
	a_paterno	VARCHAR(20)	Apellido paterno
	a_materno	VARCHAR(20)	Apellido materno
	nombre	VARCHAR(30)	Nombre del usuario
	f_nacimiento	TIMESTAMP	Fecha de nacimiento
	genero	CHAR(1)	Genero
FK	biblioteca_id	UUID	Identificador de la biblioteca asociada
FK	grupo_usuario_id	UUID	Identificador del grupo asociado
	f_registro	TIMESTAMP	Fecha de registro
	f_vencimiento	TIMESTAMP	Fecha de vencimiento
	email	VARCHAR(100)	Correo electrónico
	telefono	VARCHAR(15)	Teléfono de casa
	celular	VARCHAR(15)	Teléfono celular
	usuario	VARCHAR(20)	Usuario de acceso al sistema
	password	VARCHAR(64)	Contraseña de acceso
	foto	VARCHAR(130)	Dirección URL de la foto del usuario



*Tabla 4.3 Estructura usuarios*

#### 4.1.1.4 Tipos de ítem

Esta tabla contiene los diferentes tipos de material bibliográfico que se pueden tener, como pueden ser libros, revistas, tesis, material audiovisual, periódicos, entre otros.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
UNIQUE	codigo	VARCHAR(10)	Código del tipo de ítem
	nombre	VARCHAR(50)	Nombre del tipo de ítem
	costo_prestamo	NUMERIC(7,2)	Costo de préstamo
	disponible_prestamo	BOOLEAN	El ítem está disponible para préstamo o no

*Tabla 4.4 Estructura tipos de ítem*

#### 4.1.1.5 Registros bibliográficos

Esta tabla es donde se almacenan los datos bibliográficos de las diferentes obras que se tienen en las bibliotecas.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
	autor	VARCHAR(50)	Autor del registro
	titulo	VARCHAR(50)	Título del registro
FK	tipo_item_id	UUID	Identificador del tipo de ítem
	isbn_issn	VARCHAR(25)	Código ISBN o ISSN según sea el caso
	editorial	VARCHAR(50)	Editorial del registro
	l_publicacion	VARCHAR(50)	Lugar de publicación
	f_publicacion	TIMESTAMP	Fecha de publicación

*Tabla 4.5 Estructura registros bibliográficos*

#### 4.1.1.6 Ítems

Los ítems son los ejemplares que se tienen físicamente en cada biblioteca, un registro bibliográfico puede tener diferentes ítems.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
UNIQUE	codigo	VARCHAR(10)	Código del ítem

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

FK	registro_bib_id	UUID	Identificador del registro bibliográfico asociado
	f_adquisicion	TIMESTAMP	Fecha de adquisición del ítem
	estado_item	VARCHAR(20)	Estado en que se encuentra
	ubicacion	VARCHAR(10)	Piso o lugar en donde se encuentra ubicado
	precio	NUMERIC(7,2)	Valor monetario del ítem
FK	biblioteca_id	UUID	Identificador de la biblioteca asociada

*Tabla 4.6 Estructura ítems*

#### 4.1.1.7 Préstamos

Los préstamos almacenan el historial de préstamos de los usuarios, cada registro contiene el detalle de cada préstamo: fecha en que se prestó, se devolvió, en qué fecha se debió entregar y un contador de cuantas veces se renovó.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
FK	usuario_id	UUID	Identificador del usuario asociado
FK	item_id	UUID	Identificador del ítem asociado
	f_prestamo	TIMESTAMP	Fecha de préstamo
	f_vencimiento	TIMESTAMP	Fecha en que termina el préstamo
	f_devolucion	TIMESTAMP	Fecha en que se devolvió
	renovaciones	INTEGER	Número de veces que se ha renovado

*Tabla 4.7 Estructura préstamos*

#### 4.1.1.8 Multas manuales

Las multas por atraso se generan de manera automática, pero se necesita una manera de que se puedan agregar multas manuales, en esta tabla pueden ir este tipo de multas como pueden ser un ítem dañado, extraviado, una reposición de credencial, entre otras.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
UNIQUE	codigo	VARCHAR(10)	Código de la multa manual
	nombre	VARCHAR(50)	Nombre de la multa manual
FK	biblioteca_id	UUID	Identificador de la biblioteca asociada
	cargo	NUMERIC(7,2)	Cantidad por defecto de la multa manual

*Tabla 4.8 Estructura multas*

#### 4.1.1.9 Cuentas

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Esta tabla lleva el historial de multas de cada usuario, tanto multas manuales como multas generadas por el sistema. Si son multas generadas por el sistema llevan una referencia al ítem que no devolvieron a tiempo y si son multas manuales llevan la referencia a la multa manual y un ítem si es que así el usuario lo desea

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
	cargo	NUMERIC(7,2)	Cantidad inicial de la multa
	pendiente	NUMERIC(7,2)	Cantidad restante
FK	multa_id	UUID	Identificador de la multa asociada
FK	item_id	UUID	Identificador del ítem asociado
FK	usuario_id	UUID	Identificador del usuario asociado
	nota	TEXT	Nota de la multa

*Tabla 4.9 Estructura cuentas*

#### 4.1.1.10 Políticas

Las políticas son los criterios que se tienen para realizar el préstamo como pueden ser el número de préstamos máximo, el número de renovaciones, multas y periodos de préstamo.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
	nombre	VARCHAR(20)	Nombre de la política
	descripcion	TEXT	Descripción
FK	biblioteca_id	UUID	Identificador de la biblioteca asociada
FK	grupo_usuario_id	UUID	Identificador del grupo asociado
FK	tipo_item_id	UUID	Identificador del tipo de ítem asociado
	prestamos	INTEGER	Cantidad máxima de préstamos
	p_prestamo	VARCHAR(10)	Periodo de préstamo
	p_sancion	VARCHAR(10)	Periodo de bloqueo de préstamos
	multa	NUMERIC(7,2)	Multa económica
	renovaciones	INTEGER	Renovaciones máximas
	p_gracia	VARCHAR(10)	Periodo de gracia

*Tabla 4.10 Estructura políticas*

#### 4.1.1.11 Direcciones

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Algunas veces es necesario tener múltiples direcciones para localizar a los usuarios y para ello se almacenan las direcciones en una tabla aparte en vez de incluir los campos en la tabla usuarios.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
FK	usuario_id	UUID	Identificador del usuario asociado
	direccion1	TEXT	Dirección línea 1
	direccion2	TEXT	Dirección línea 2
	estado	VARCHAR(30)	Estado
	c_postal	VARCHAR(10)	Código postal
	municipio	VARCHAR(30)	Municipio

*Tabla 4.11 Estructura direcciones*

#### 4.1.1.12 Bloqueos

Llevan el registro de que usuarios no tienen permitido acceder al servicio de préstamo durante algún periodo el cual esta almacenado en esta tabla.

CLAVE	CAMPO	TIPO	DESCRIPCION
PK	id	UUID	Identificador del registro
FK	usuario_id	UUID	Identificador del usuario asociado
	f_inicio	TIMESTAMP	Fecha de inicio del bloqueo
	f_termino	TIMESTAMP	Fecha de término del bloqueo

*Tabla 4.12 Estructura bloqueos*

### 4.1.2 Diagrama entidad relación

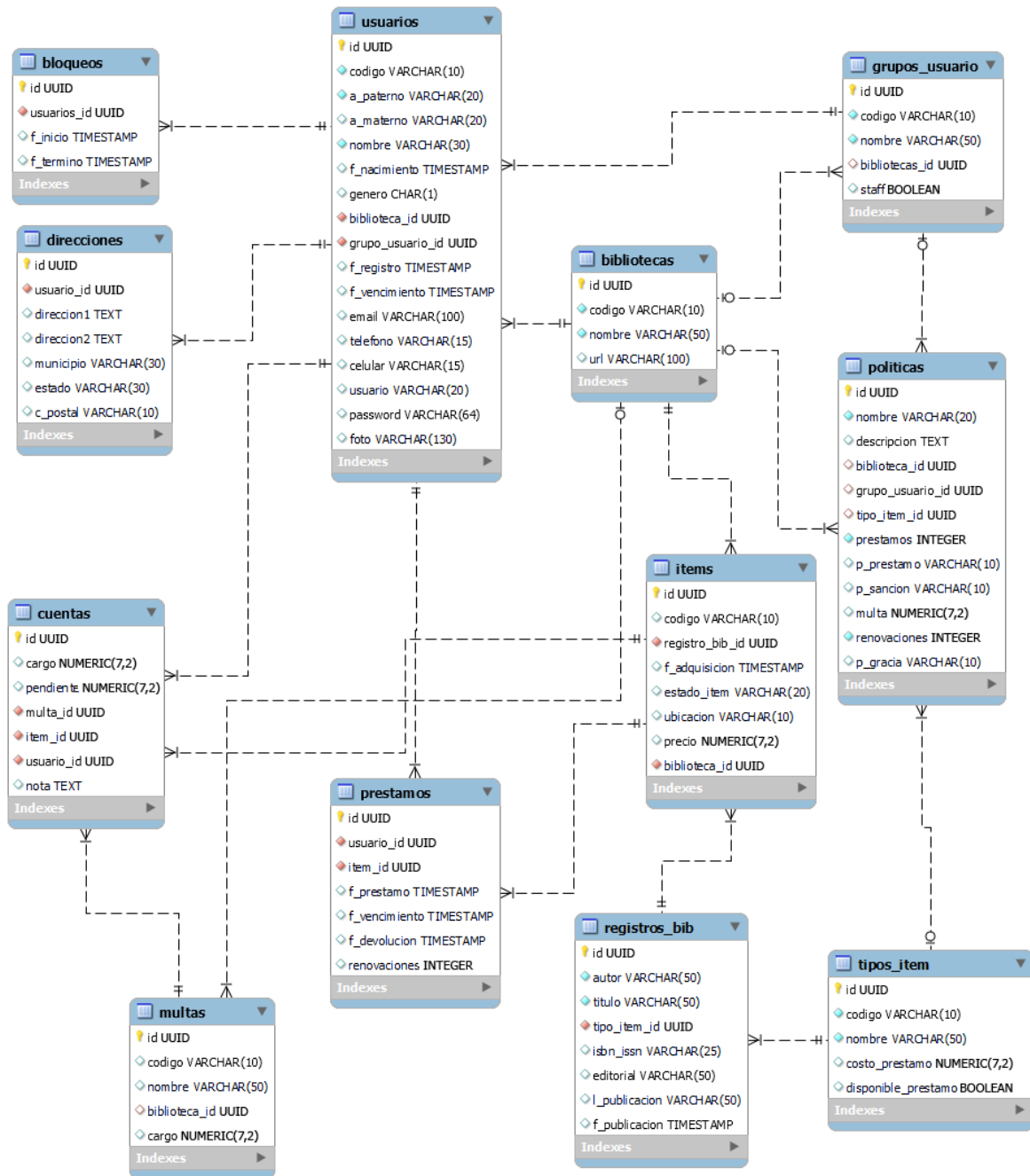


Figura 4.1 Diagrama entidad relación

## 4.2 Diseño de la interfaz gráfica

El sistema estará conformado por 5 módulos los cuales estarán disponibles a través de la barra de navegación de la página principal. Para acceder a la página principal primero será necesario autenticarse en la página de inicio de sesión por medio de un usuario y contraseña válidos.

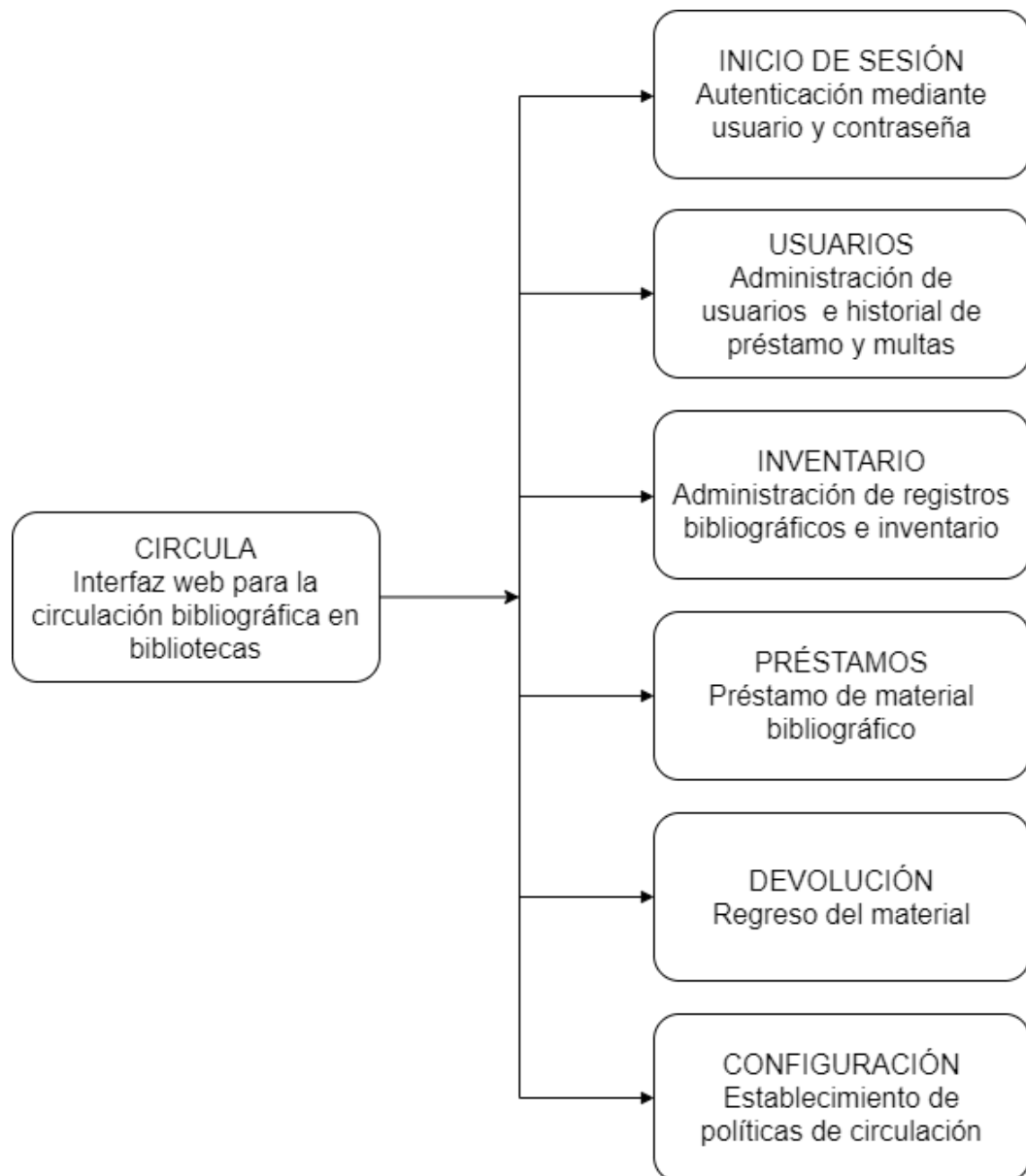


Figura 4.2 Diagrama interfaz gráfica

#### 4.2.1 Inicio de sesión

Este módulo se encargará de proteger al sistema de visitas no deseadas, solo los usuarios administradores del sistema podrán contar con un usuario y contraseña para acceder al sistema. Los usuarios administradores serán aquellos que pertenezcan a algún grupo de usuario con la bandera de staff.

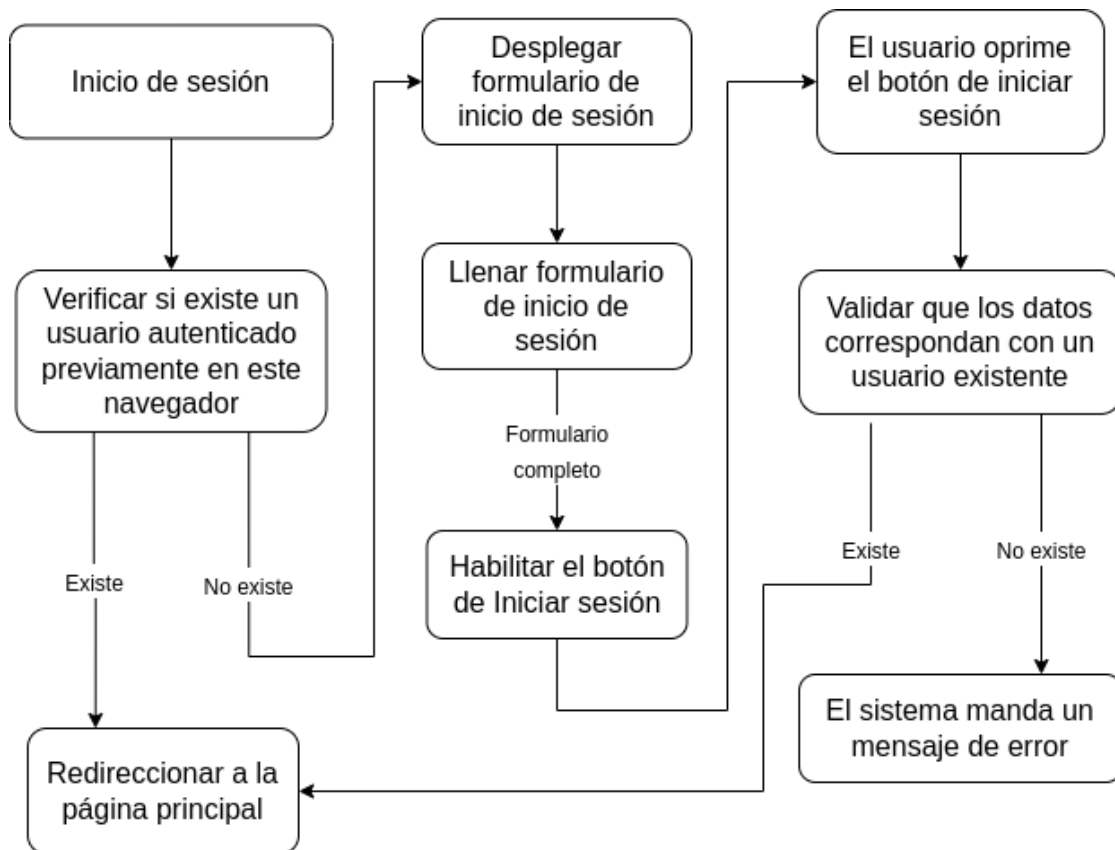


Figura 4.3 Diagrama inicio de sesión

#### 4.2.2 Usuarios

Este módulo se encargará de administrar los usuarios del sistema, permitiendo la lectura y escritura de estos. Se podrá visualizar el detalle de cada usuario, así como su historial de multas y préstamos.

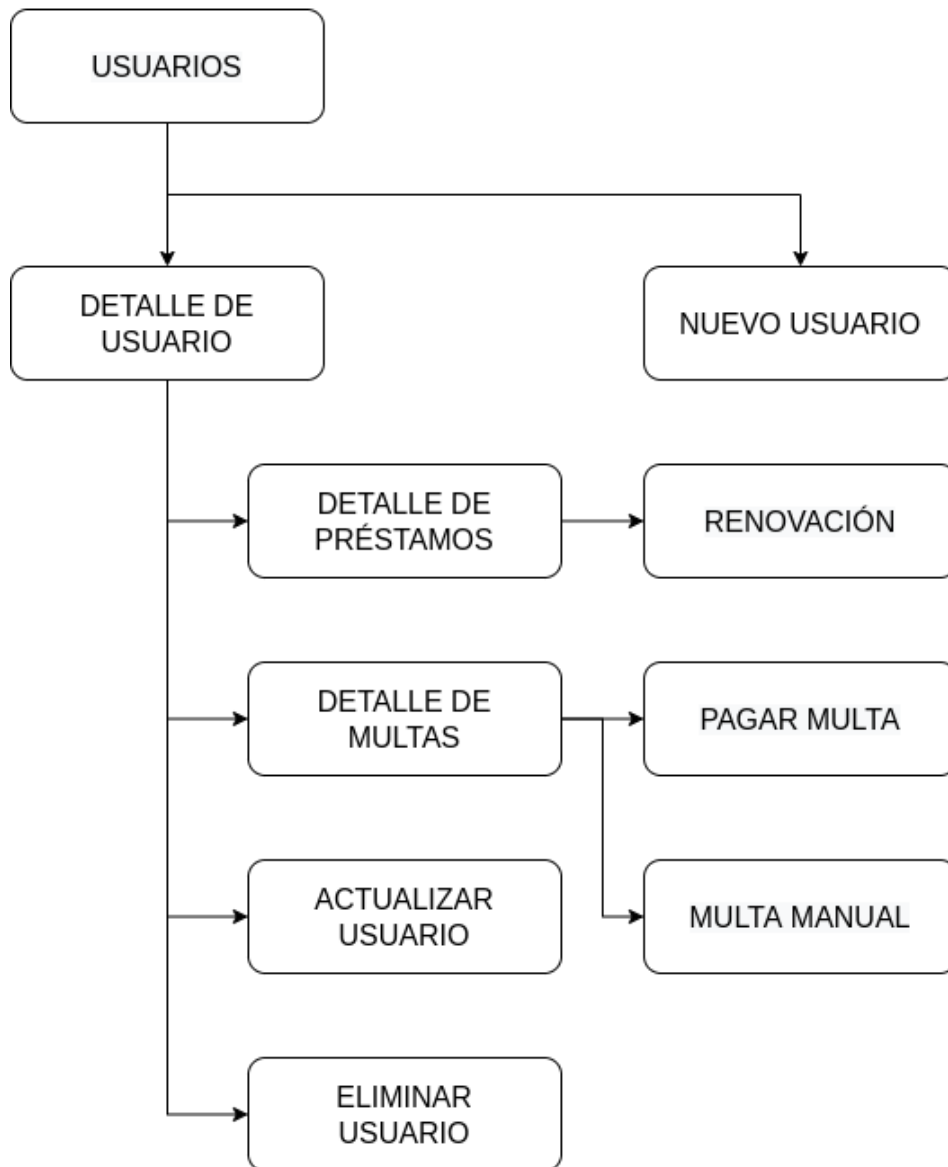
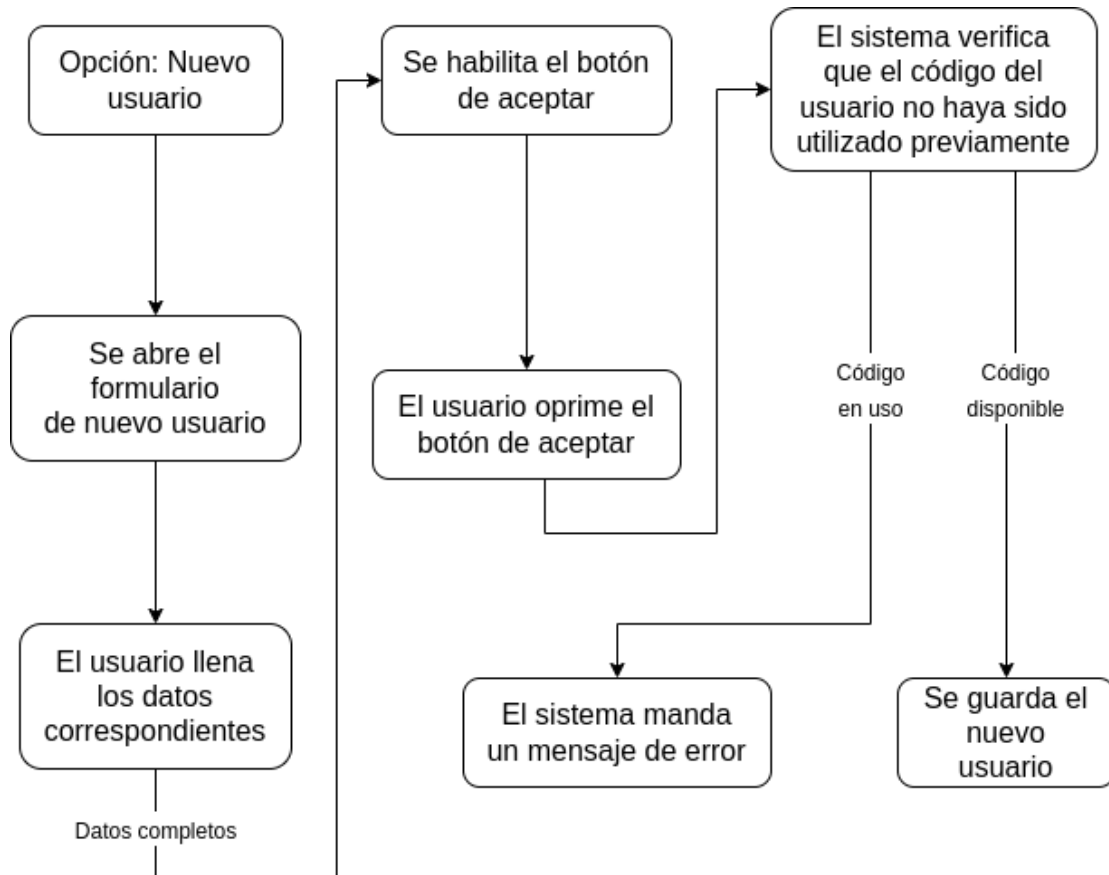


Figura 4.4 Diagrama usuarios



### *Nuevo Usuario*

Esta opción servirá para agregar nuevos usuarios al sistema. Se podrá acceder a ella mediante el botón de opciones del módulo.



*Figura 4.5 Diagrama nuevo usuario*

### *Actualizar Usuario*

Esta opción se utilizará para modificar la información del usuario. Se podrá acceder a ella mediante el botón de opciones del panel del detalle de usuario. Para acceder a este panel se tiene que seleccionar un usuario.

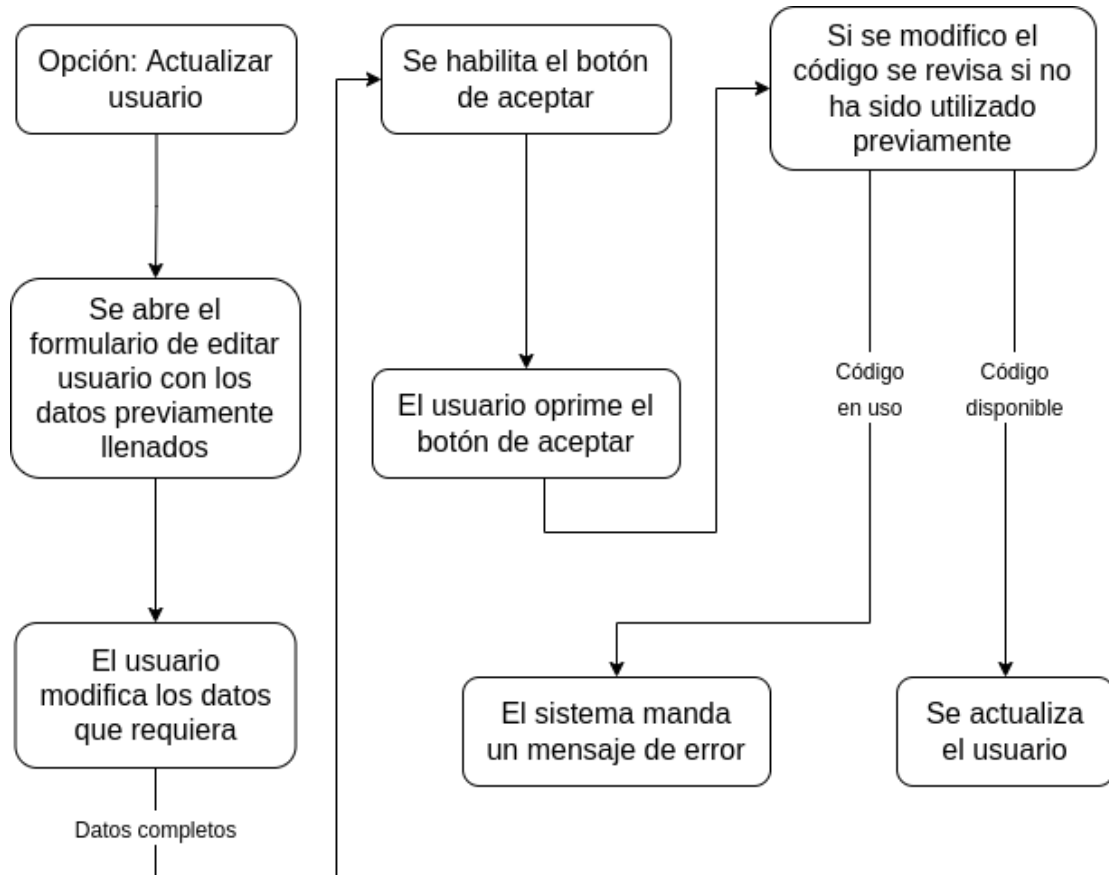


Figura 4.6 Diagrama actualizar usuario

### Eliminar Usuario

Esta opción servirá para eliminar usuarios, sin embargo, esto no será posible si es que este tiene préstamos y/o multas registradas. Esta opción estará disponible desde el botón de opciones del panel del detalle de usuario

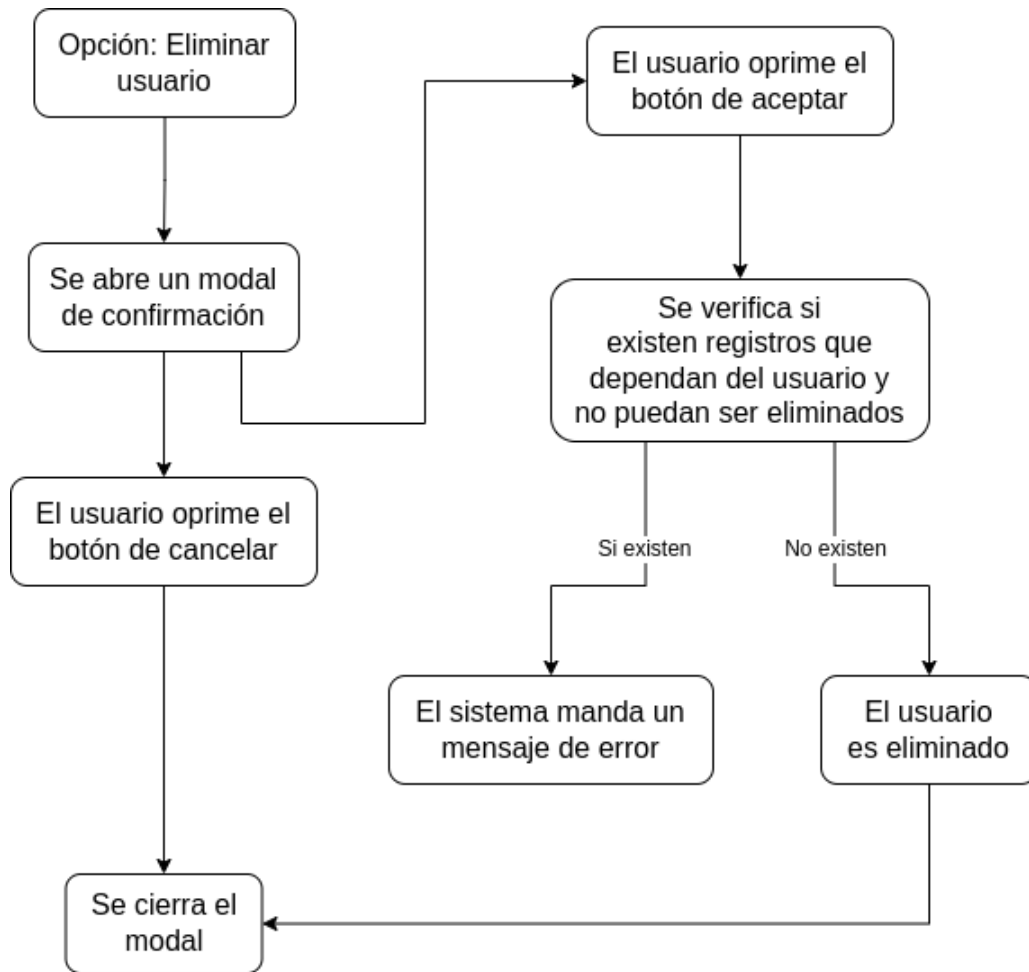


Figura 4.7 Diagrama eliminar usuario

### *Multa manual*

Esta opción servirá para que la biblioteca tenga la posibilidad de generar multas que no se deban a la devolución tardía del material bibliográfico, por ejemplo, cuando un usuario extravía algún libro, cuando lo daña, o también para registrar el pago de una reposición de credencial. Estos conceptos deberán ser definidos en el módulo de configuración. Esta opción estará disponible desde el panel de historial de multas, para lo cual habrá que seleccionar un usuario y posteriormente abrir el historial de multas.

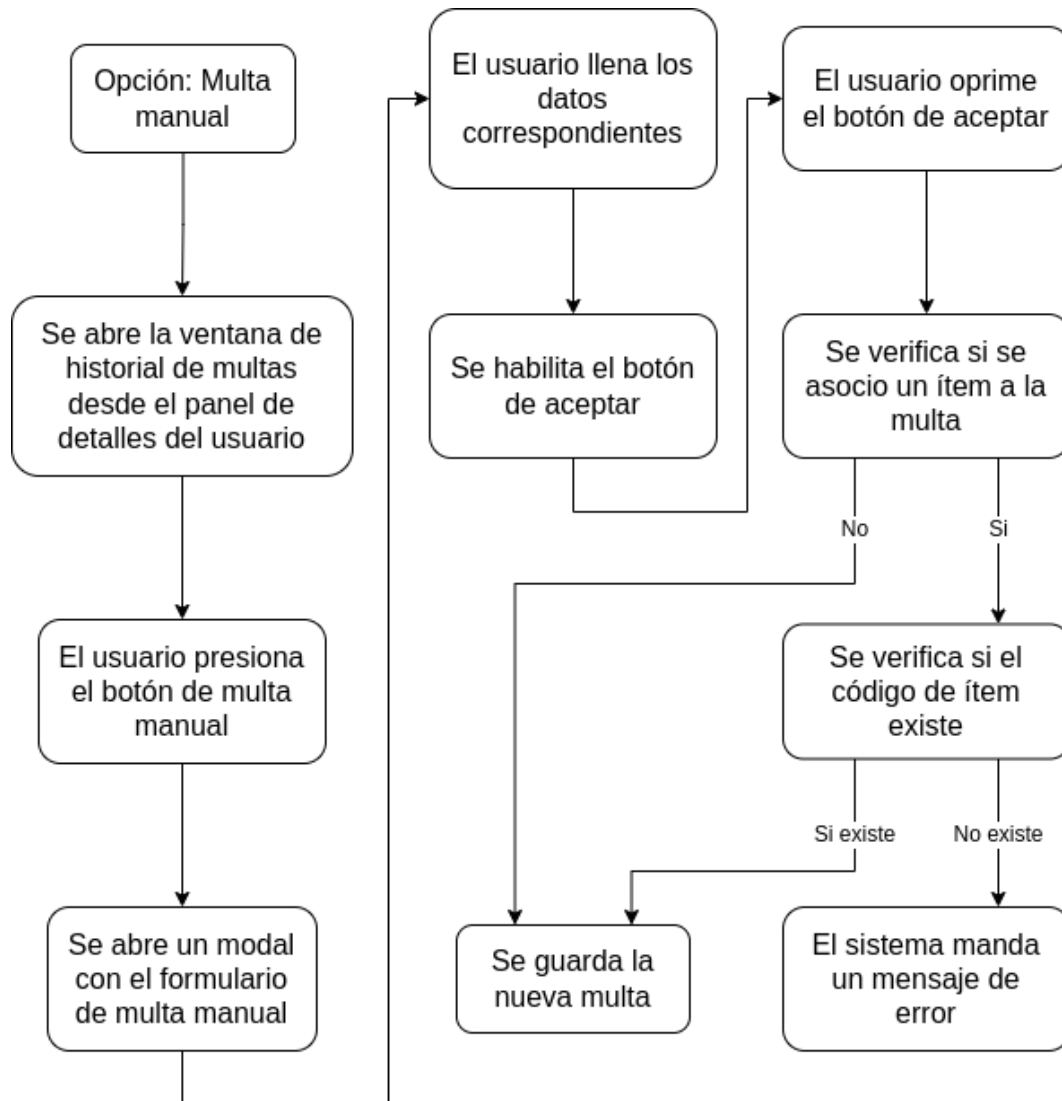


Figura 4.8 Diagrama multa manual

### Pagar

Esta opción servirá para registrar el pago de las multas de manera total o parcial. Estará disponible desde el botón de opciones de la multa que se desea pagar. Para acceder a ella se debe abrir primero el historial de multas.

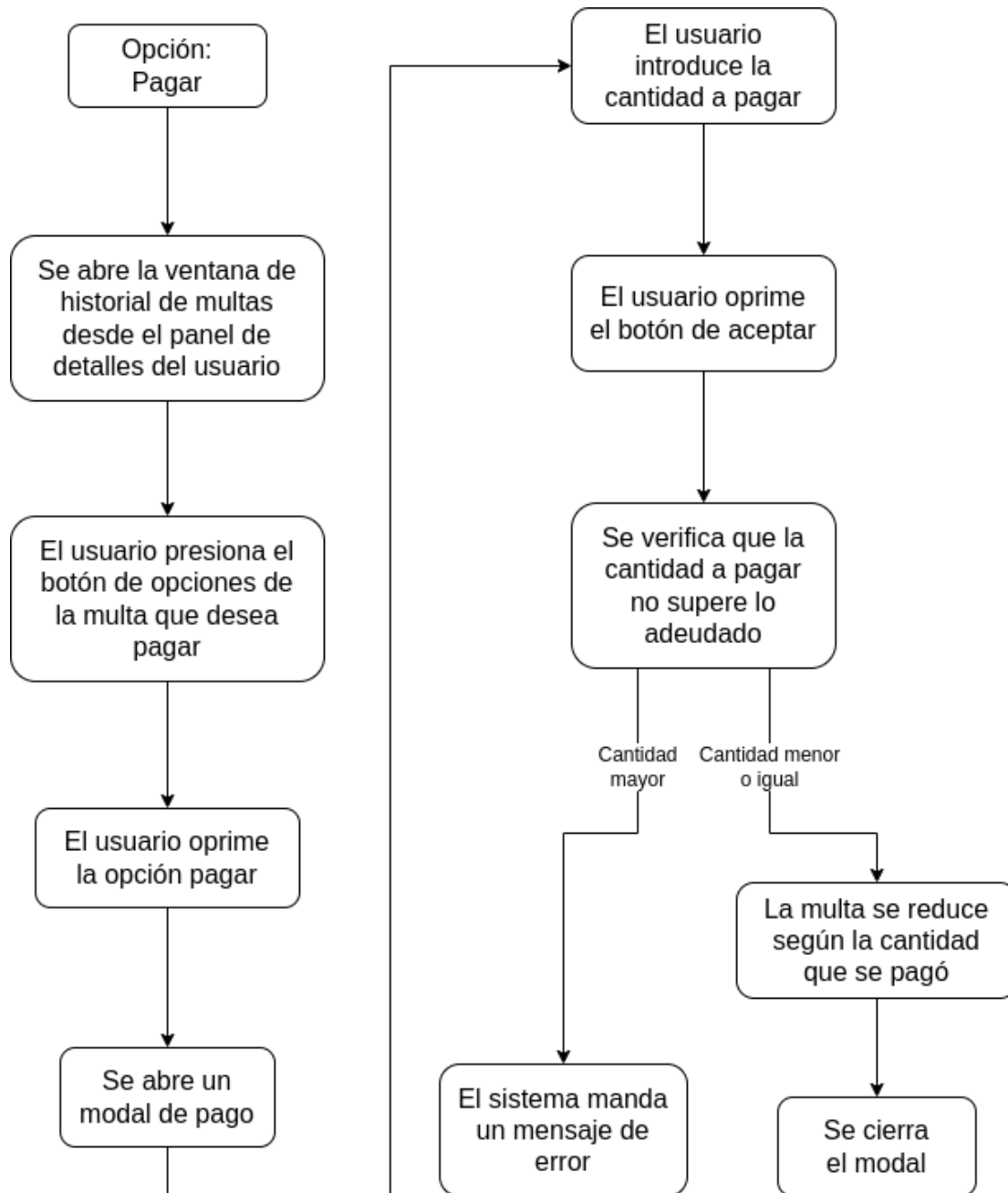


Figura 4.9 Diagrama pagar multa

### Renovación

Esta opción servirá para renovar el préstamo de algún material un periodo de tiempo determinado en las políticas de circulación. Para acceder a esta opción se deberá de abrir el historial de préstamos desde el panel de detalles de usuario y posteriormente ver las opciones para el préstamo que se desea renovar.

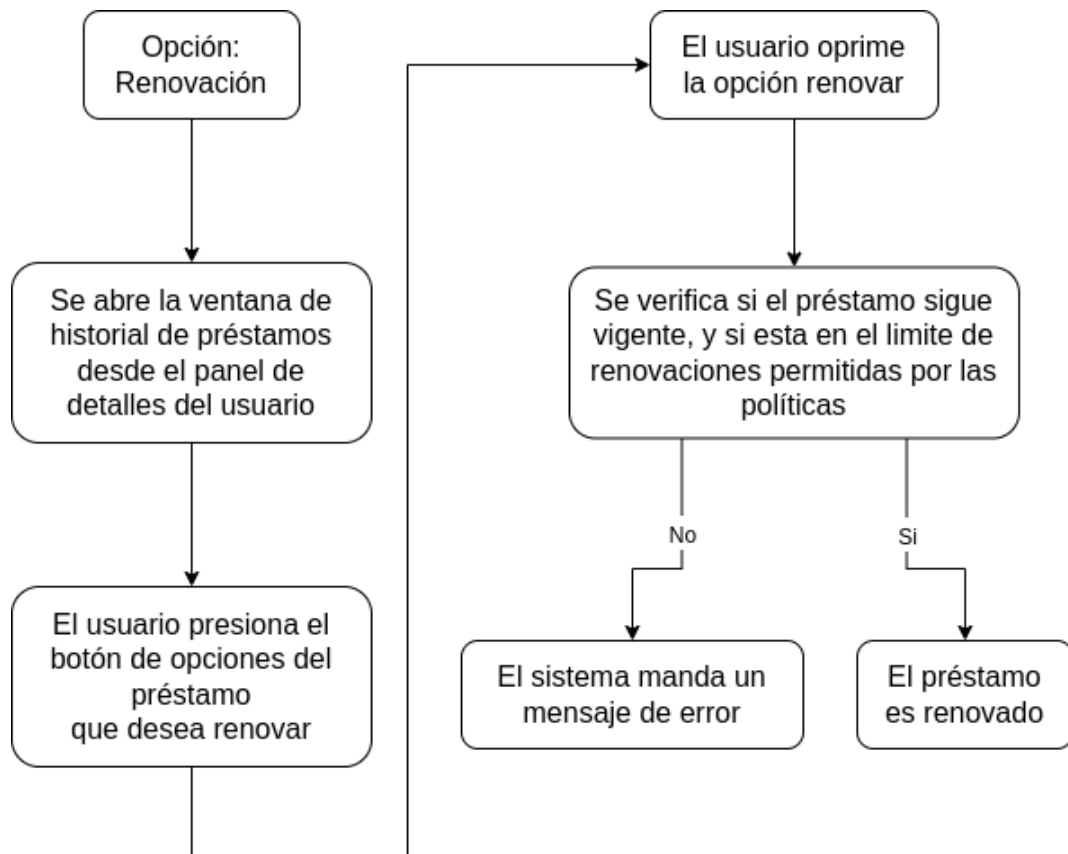


Figura 4.10 Diagrama renovación de préstamo

### 4.2.3 Inventario

Este módulo se encargará de administrar los registros bibliográficos, así como los ítems asociados a ellos, permitiendo operaciones de lectura y escritura.

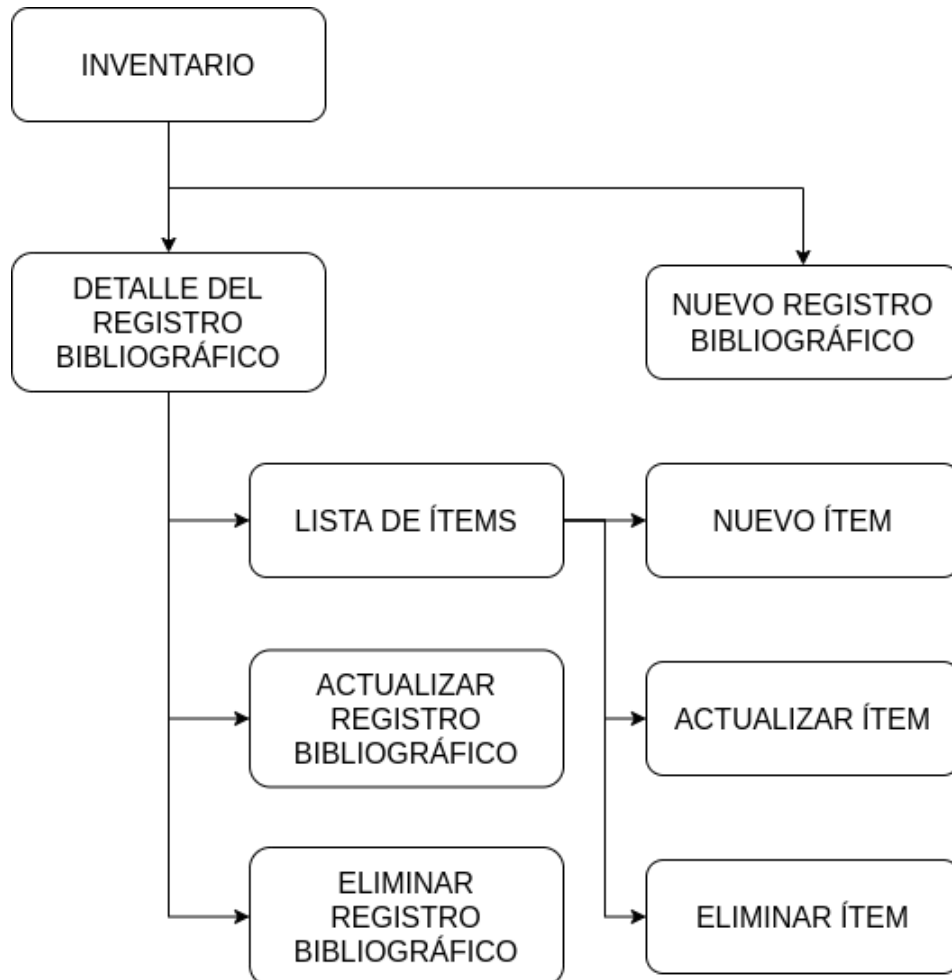


Figura 4.11 Diagrama inventario

#### *Nuevo registro bibliográfico*

Esta opción servirá para agregar nuevos registros bibliográficos al sistema. Se podrá acceder a ella mediante el botón de opciones del módulo.

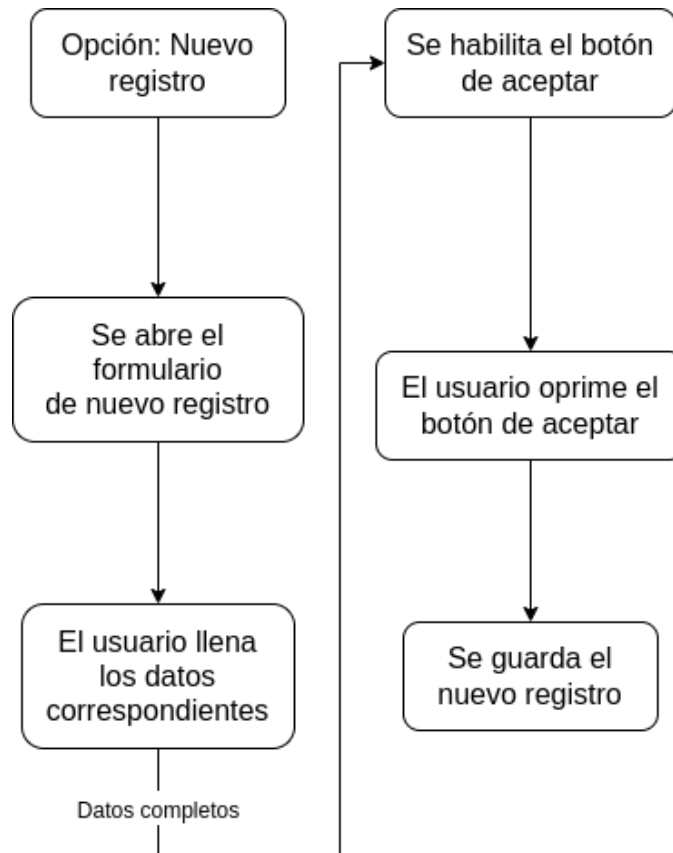


Figura 4.12 Diagrama nuevo registro

### *Actualizar registro bibliográfico*

Esta opción se utilizará para modificar la información del registro bibliográfico. Se podrá acceder a ella mediante el botón de opciones del panel del detalle del registro. Para acceder a este panel se tiene que seleccionar un registro bibliográfico.



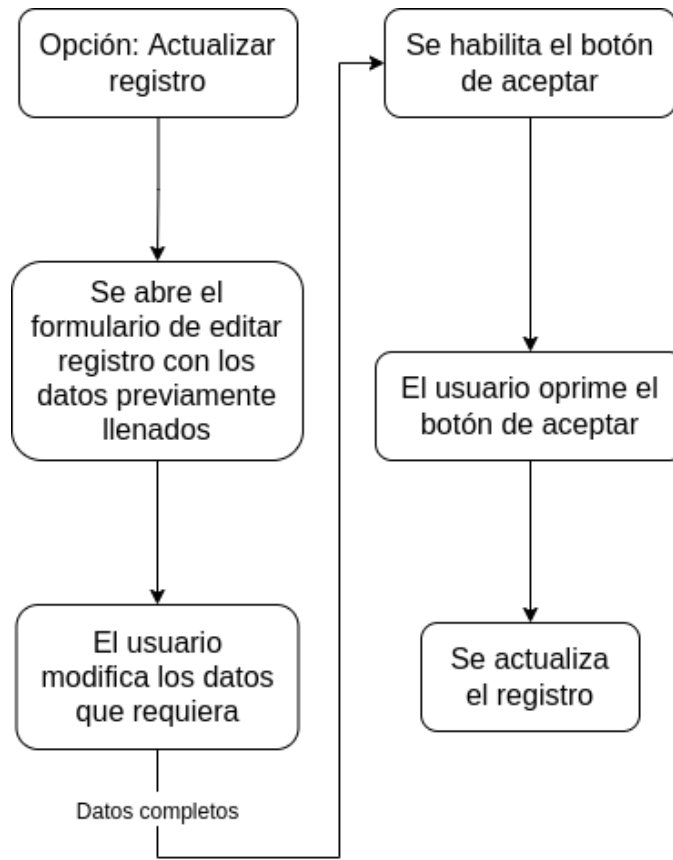


Figura 4.13 Diagrama actualizar registro

### *Eliminar registro bibliográfico*

Esta opción servirá para eliminar un registro bibliográfico, sin embargo, esto no será posible si es que este tiene ítems asociados. Esta opción estará disponible desde el botón de opciones del panel del detalle de registro bibliográfico.

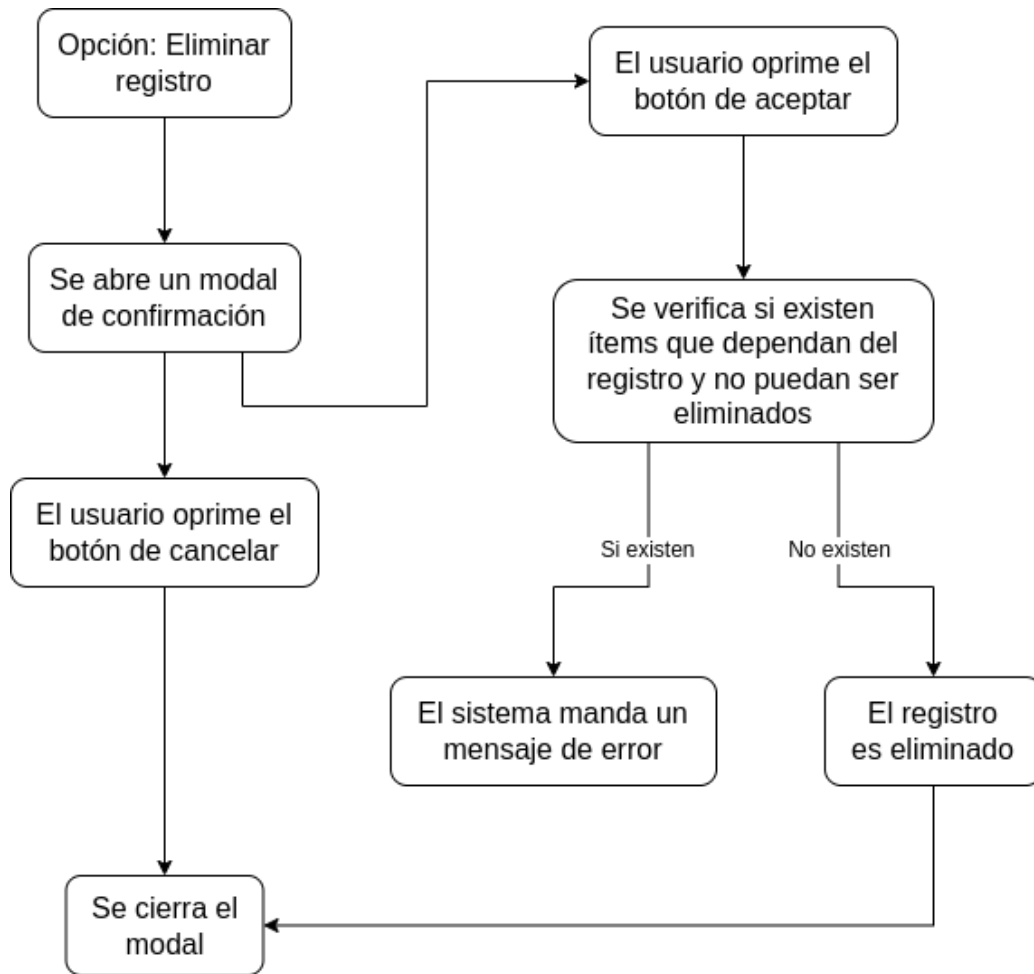


Figura 4.14 Diagrama eliminar registro

### Nuevo ítem

Esta opción servirá para agregar nuevos ejemplares de algún registro bibliográfico al sistema a los cuales denominaremos ítems. Esta opción estará disponible desde el panel de detalles del registro bibliográfico.

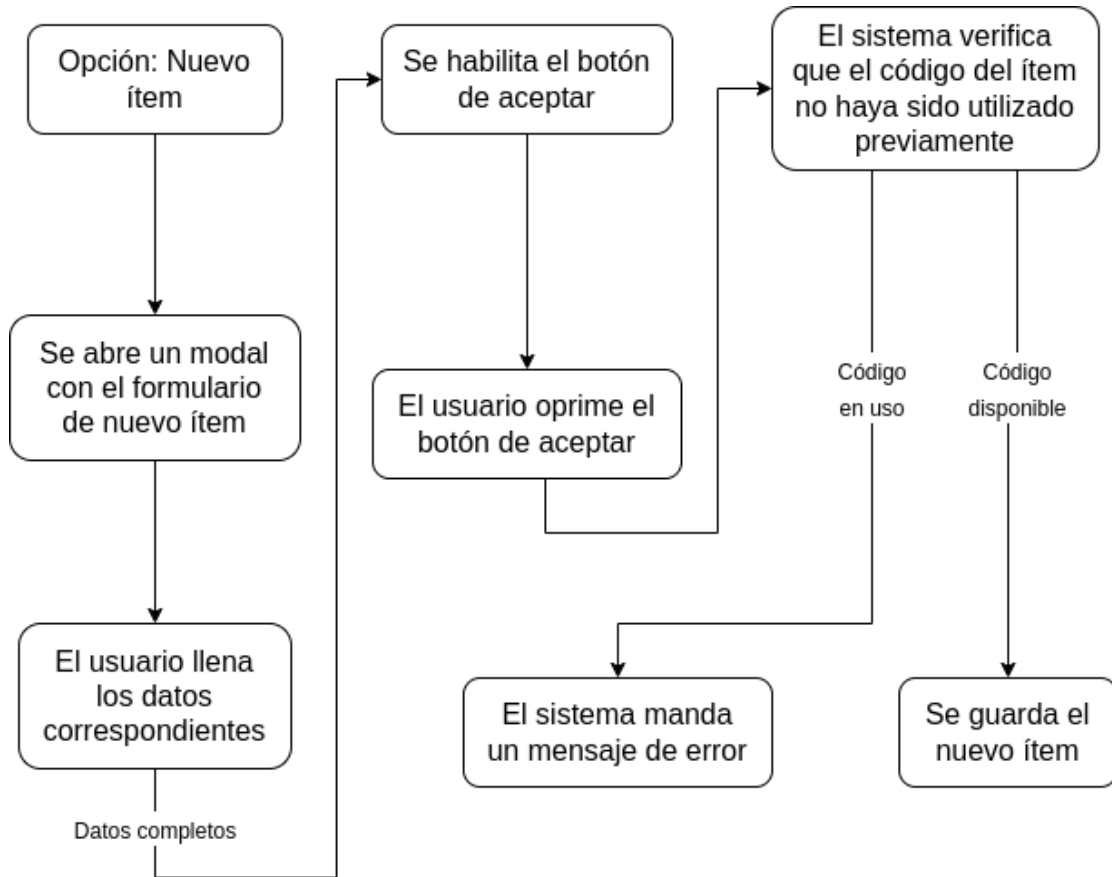


Figura 4.15 Diagrama nuevo ítem

### Actualizar ítem

Esta opción se utilizará para modificar la información de un ítem. En el panel de detalles del registro bibliográfico se mostrarán todos los ítems asociados, y junto a cada uno se tendrá la opción de editar.

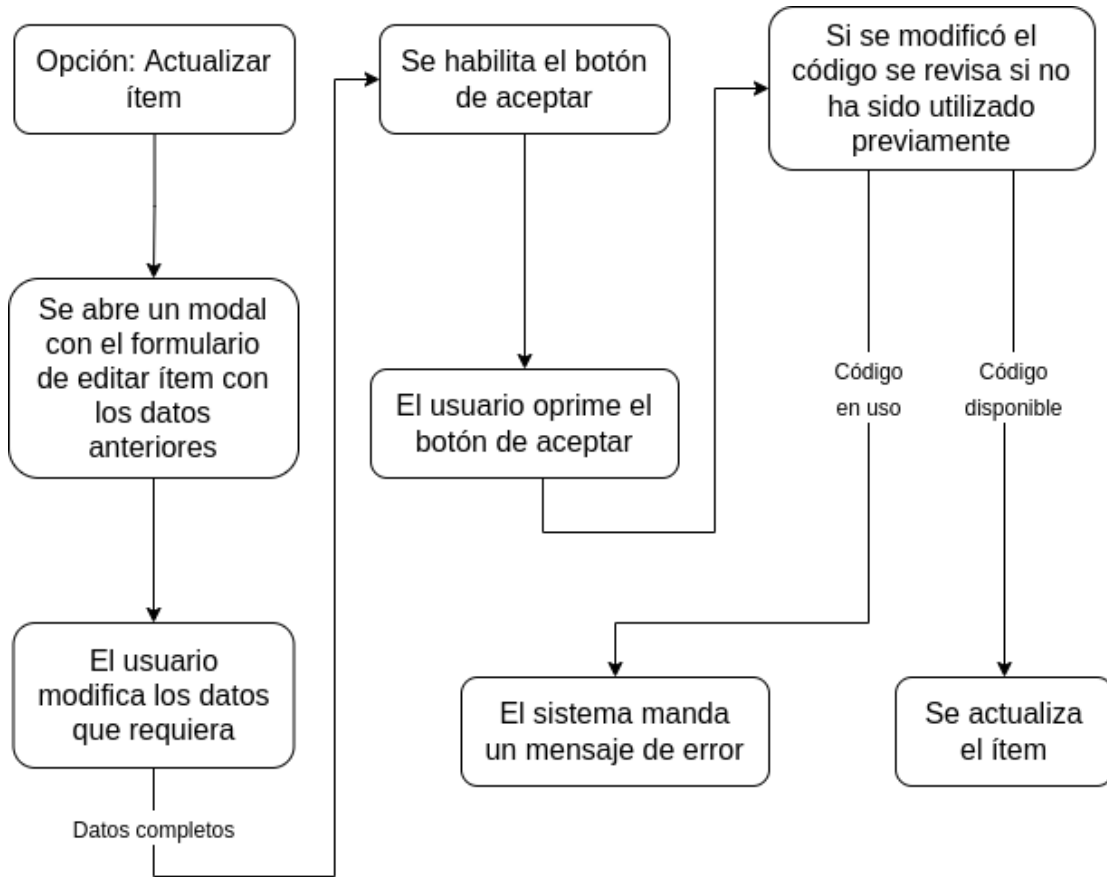


Figura 4.16 Diagrama actualizar ítem

### *Eliminar ítem*

Esta opción servirá para eliminar un ítem, sin embargo, esto no será posible si es que este ítem está asociado a alguna multa o a algún préstamo. Esta opción estará disponible en la lista de ítems del registro bibliográfico.

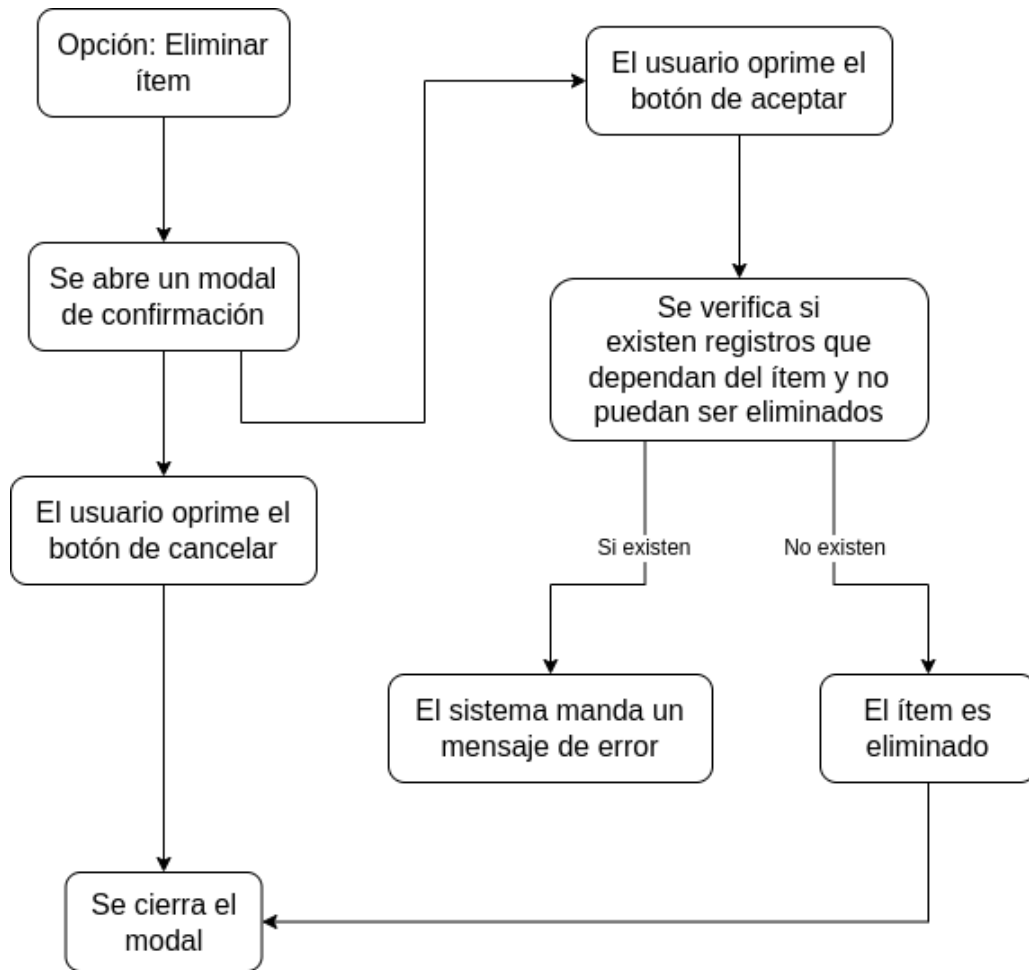


Figura 4.17 Diagrama eliminar ítem

#### 4.2.4 Préstamo

Este módulo se utilizará para realizar préstamo de material bibliográfico a los usuarios. Al tener un módulo que se encargue únicamente de los préstamos se agilizará la tarea de préstamo para los administradores del sistema.

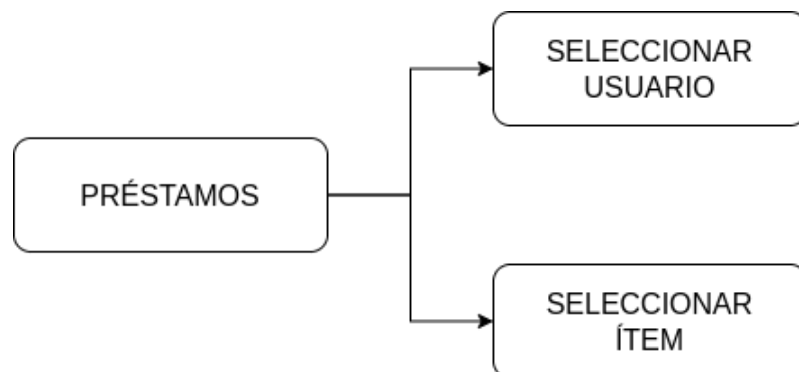
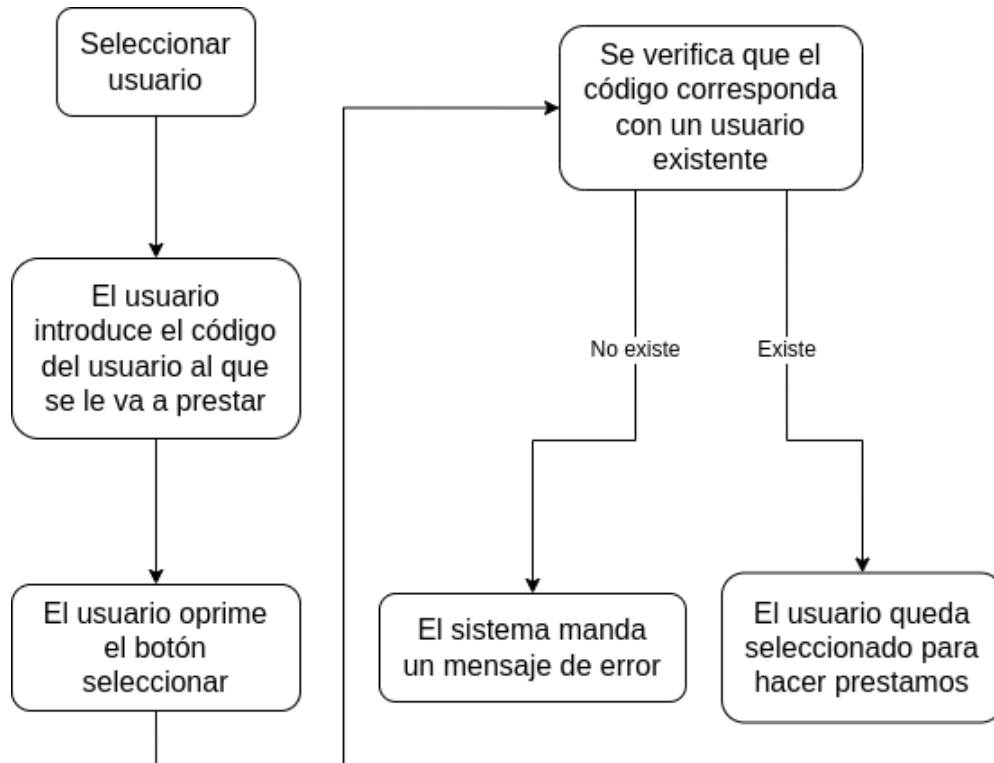


Figura 4.18 Diagrama préstamo

### *Seleccionar usuario*

Para realizar un préstamo primero se deberá seleccionar un usuario de manera obligatoria, si no, no se podrá seleccionar un ítem para prestar.



*Figura 4.19 Diagrama seleccionar usuario a prestar*

### *Seleccionar ítem*

Luego de haber seleccionado un usuario se podrán seleccionar n ítems siempre y cuando estos estén disponibles para préstamo.

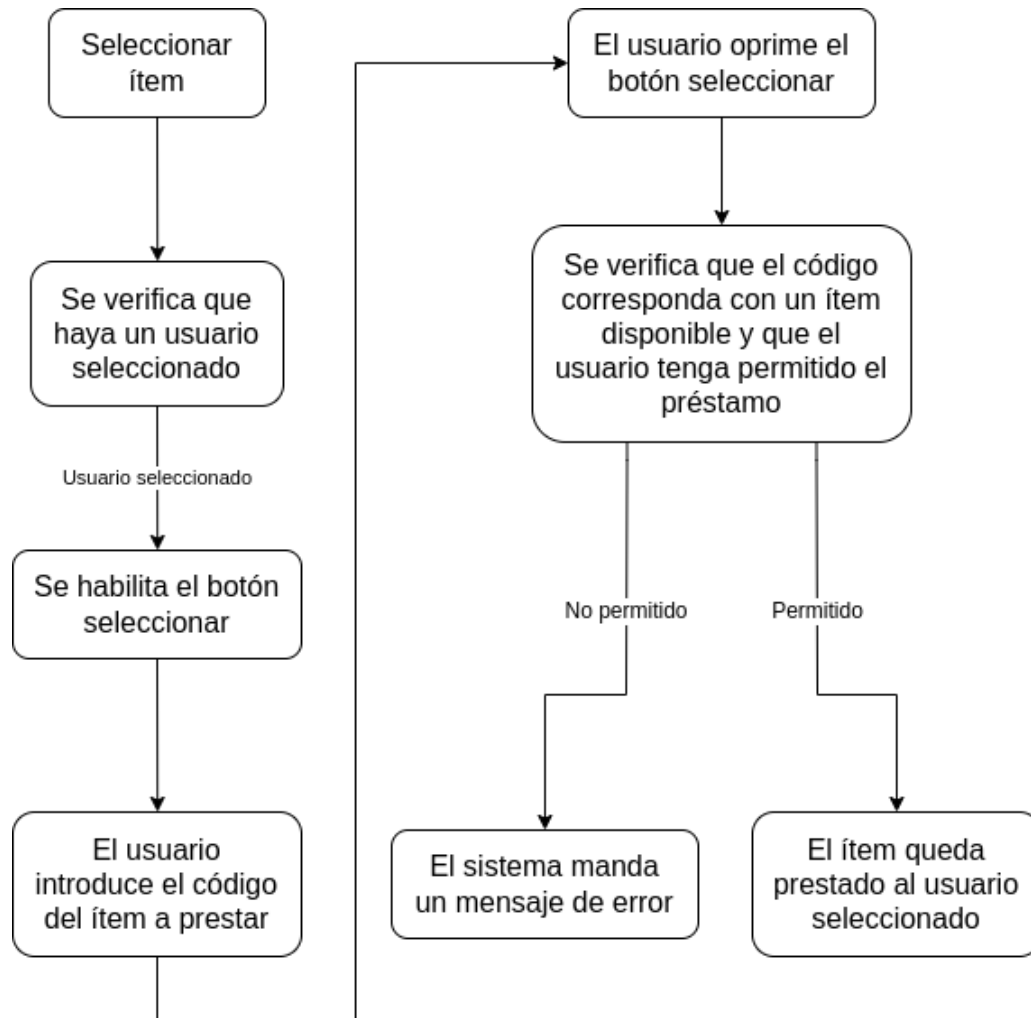


Figura 4.20 Diagrama seleccionar ítem a prestar

#### 4.2.5 Devolución

Este módulo se empleará para devolver el material prestado. Las operaciones de devolución y préstamo están en módulos independientes por que implican el mayor número de transacciones en un módulo de circulación.

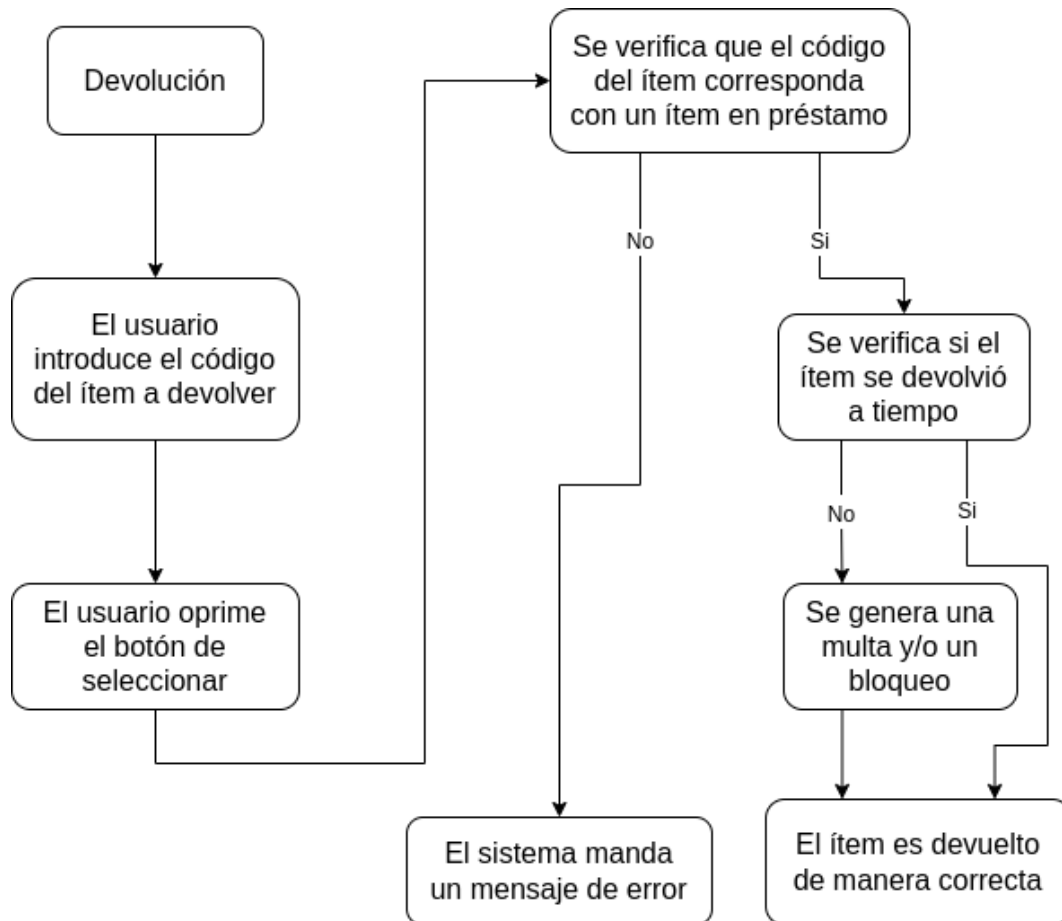


Figura 4.21 Diagrama devolver

#### 4.2.6 Configuración

Aquí se podrán establecer las políticas de circulación, así como las diferentes bibliotecas que administrara el sistema, los grupos de usuarios, los tipos de material bibliográfico y las multas manuales. Todas las anteriores tendrán las operaciones de creación, edición, borrado y lectura (CRUD).



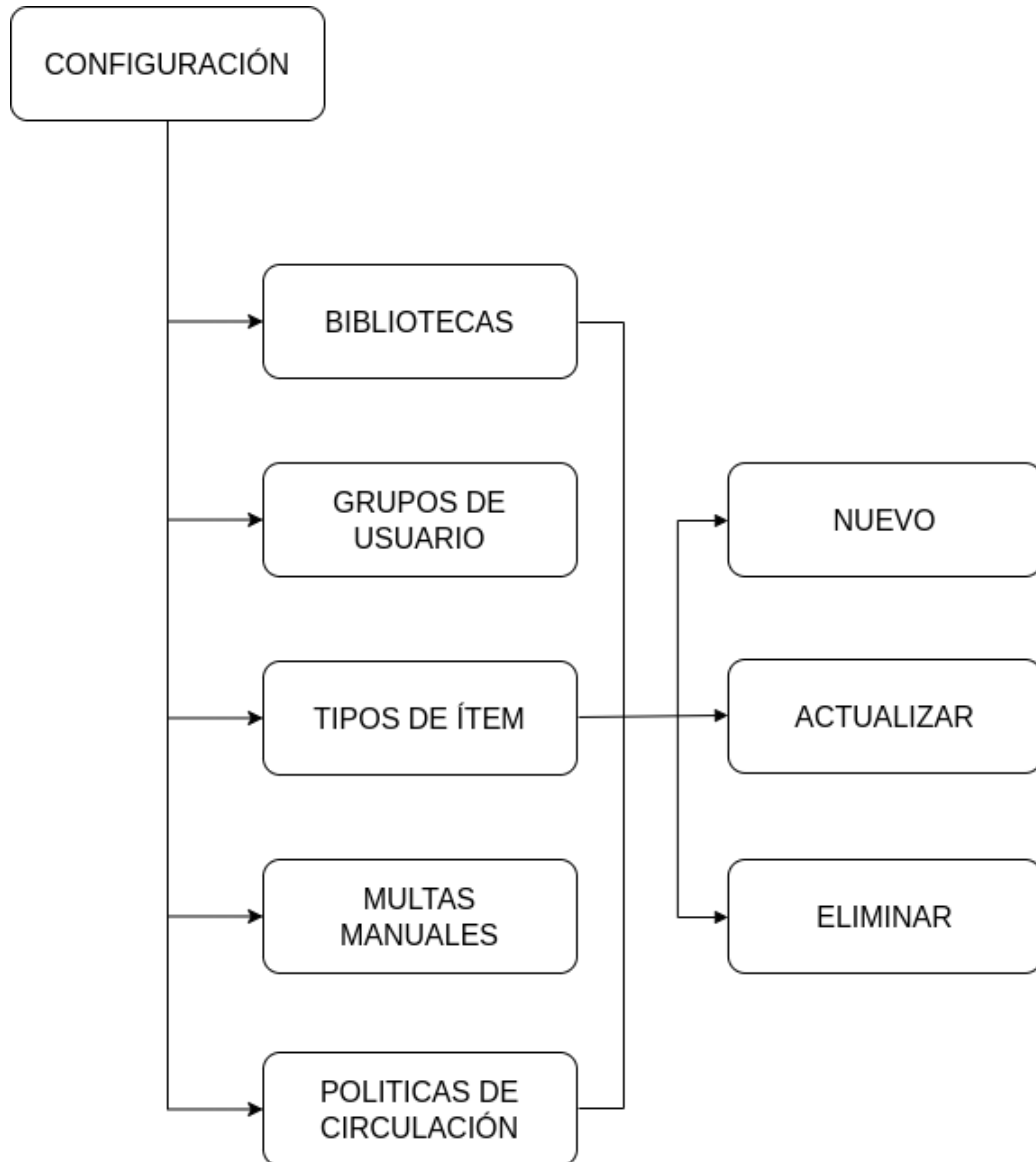


Figura 4.22 Diagrama configuración

### *Nueva configuración*

Esta opción estará disponible en la pestaña de la configuración seleccionada (bibliotecas, grupos de usuario, etc.) y servirá para agregar un nuevo registro.

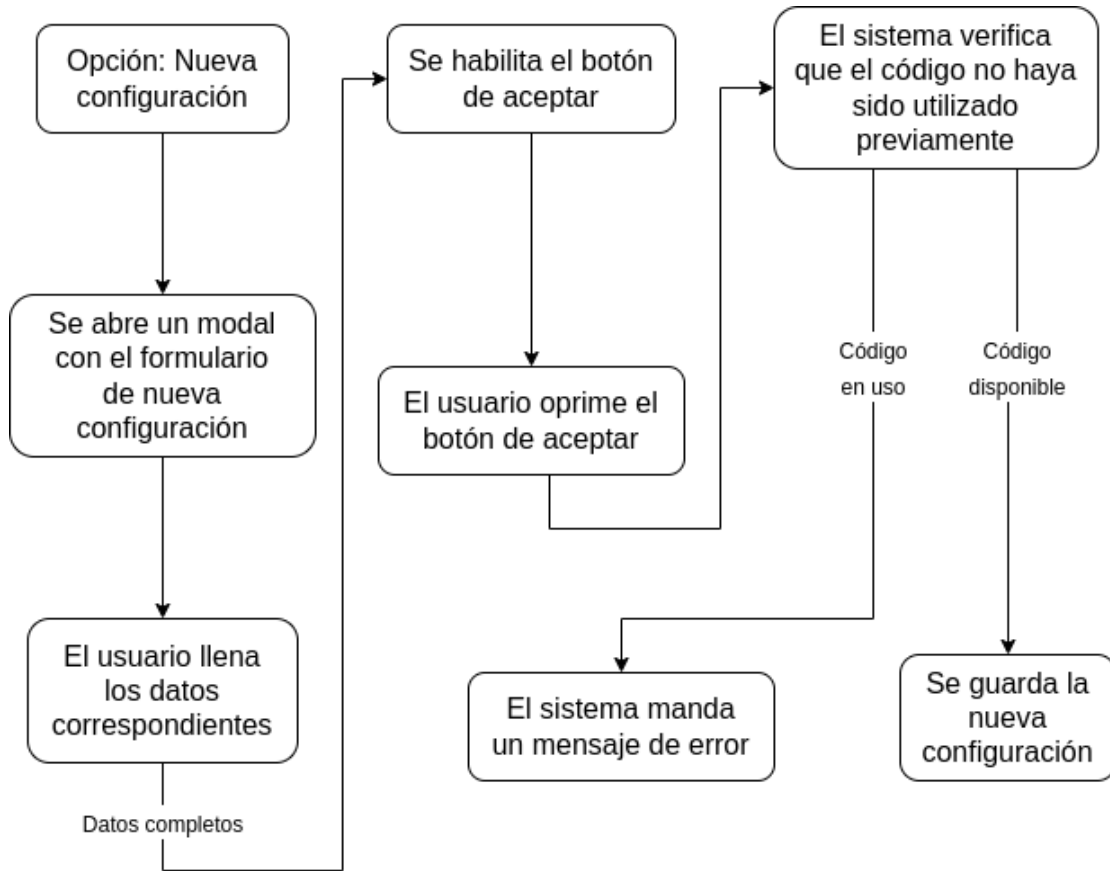


Figura 4.23 Diagrama nueva configuración

### Actualizar configuración

Esta opción estará disponible en la pestaña de la configuración seleccionada junto al elemento que se desea actualizar y servirá para modificar ese registro.

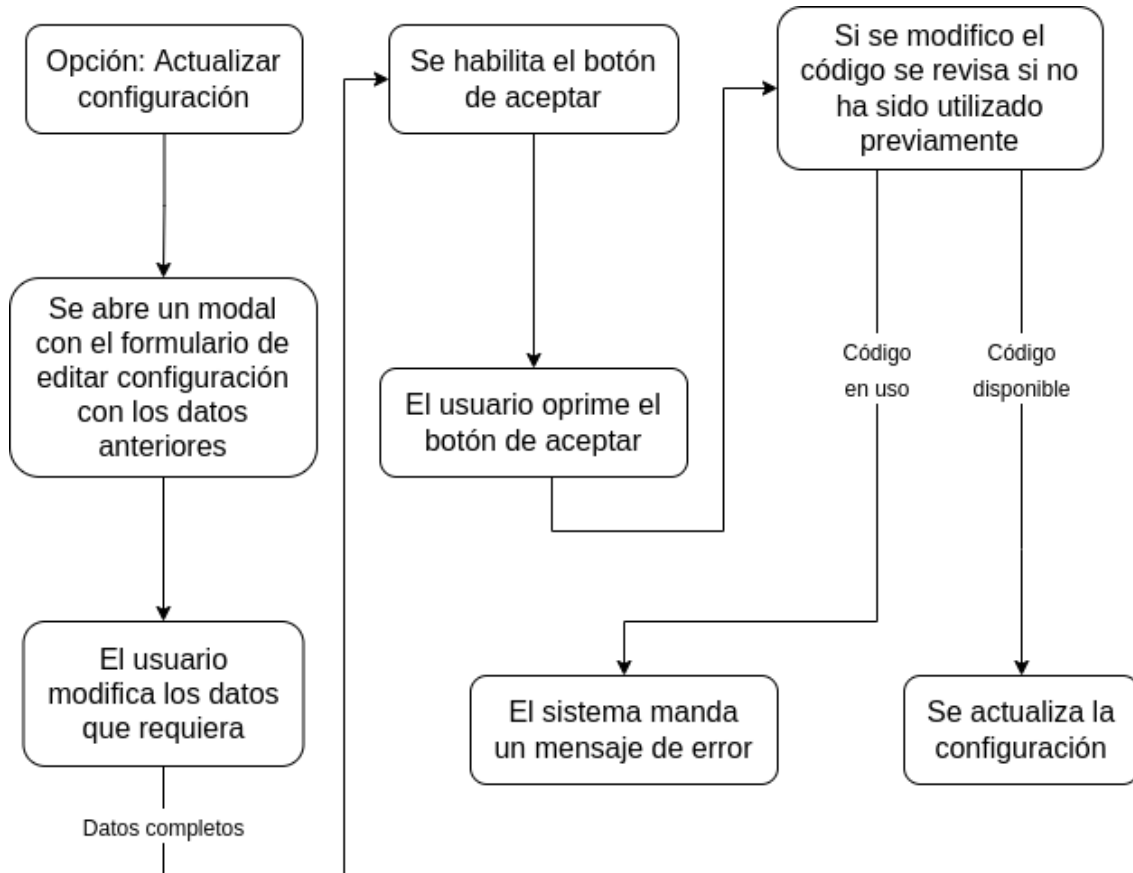


Figura 4.24 Diagrama actualizar configuración

### *Eliminar configuración*

Esta opción estará disponible en la pestaña de la configuración seleccionada junto al elemento que se desea eliminar y servirá para eliminar ese registro siempre y cuando no esté siendo usado por otros registros.

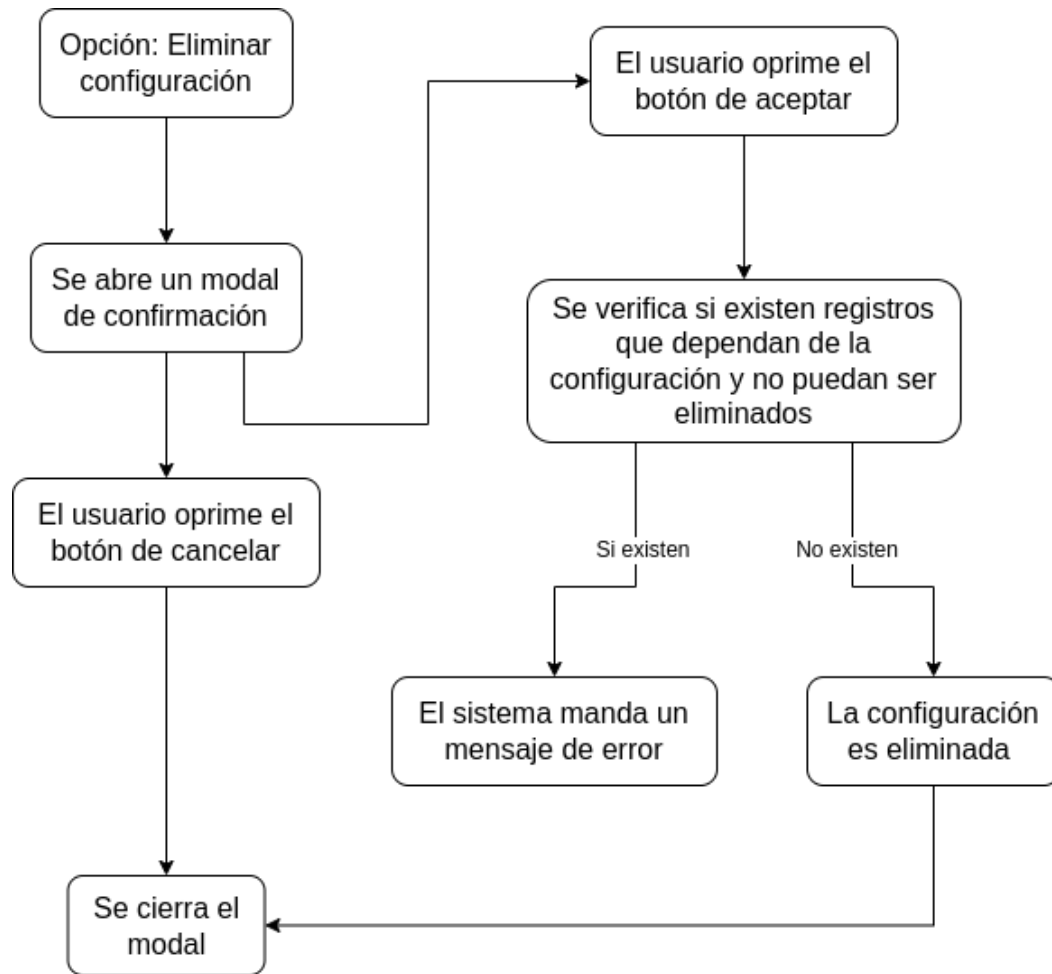


Figura 4.25 Diagrama eliminar configuración

### 4.3 Desarrollo de la interfaz gráfica

El sistema consta de 5 módulos que estarán disponibles cuando el usuario este autenticado. Si no es el caso solo se mostrará la página de autenticación. A continuación, se mostrará el funcionamiento de cada módulo.

#### 4.3.1 Autenticación

El módulo de autenticación de mostrará siempre y cuando no exista un usuario autenticado en el sistema. Se desplegará la vista de la figura 4.25, en donde el usuario deberá colocar sus credenciales (usuario y contraseña) para poder acceder al sistema. Dichas credenciales deberán ser proporcionadas por el administrador del sistema.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

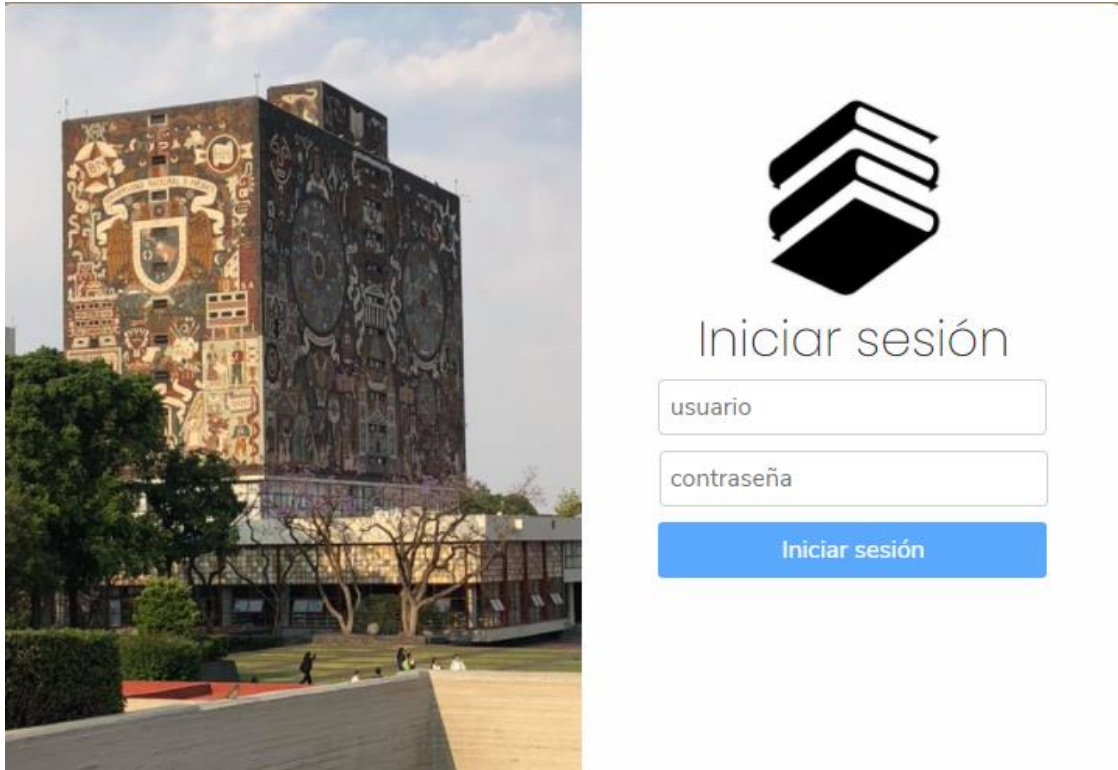


Figura 4.26 Inicio se sesión

Si las credenciales son incorrectas el sistema desplegará un mensaje de error, si no redirigirá al usuario a la página principal del sistema.

Usuario y/o contraseña incorrectos.

Figura 4.27 Error de inicio de sesión

La página principal consta de una barra de navegación y un panel en donde se desplegarán los distintos módulos. A la mitad de la barra de navegación se encuentran 4 módulos del sistema, y al final se encuentra la información del usuario autenticado y un botón de opciones.



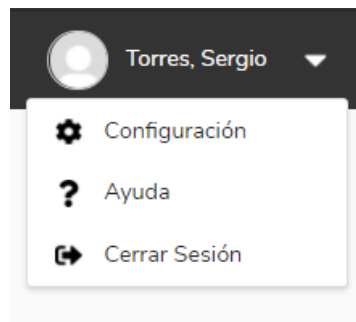
**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

*Figura 4.28 Menú de navegación*

El botón de opciones desplegará un menú en donde se encontrarán las opciones de configuración, ayuda y cerrar sesión.

- La opción Configuración despliega el módulo para modificar las políticas de circulación.
- La opción Ayuda, abrirá una nueva pestaña en donde se podrá revisar la documentación del software.
- La opción de Cerrar sesión hará que la sesión termine y mandará al usuario a la página de autenticación.



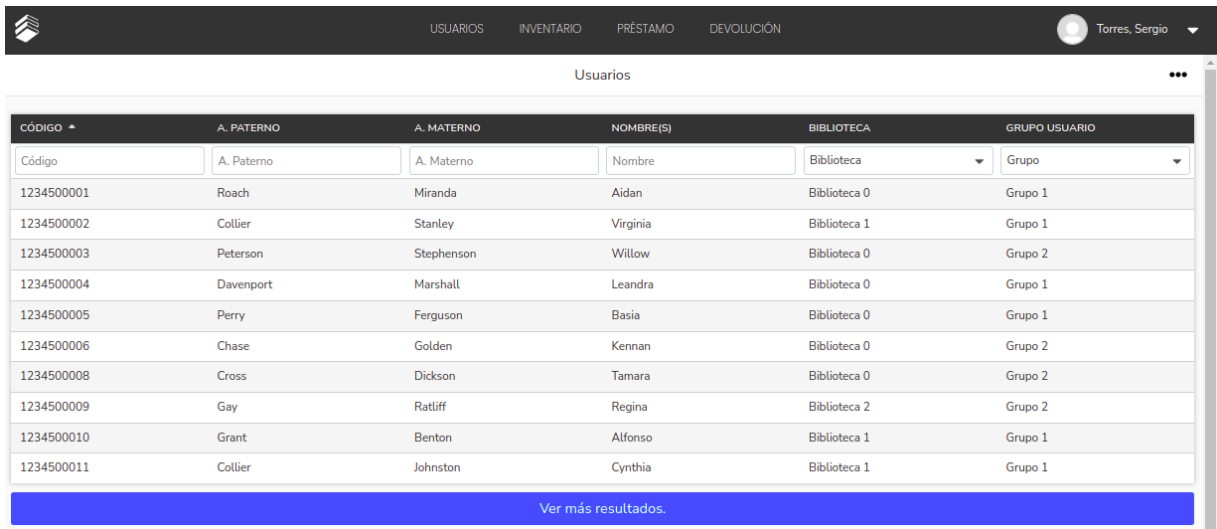
*Figura 4.29 Menú de opciones*

### 4.3.2 Usuarios

Podemos acceder al módulo de Usuarios por medio de la barra de navegación (figura 4.28), el módulo contiene una tabla en donde se visualizan los registros de los usuarios del sistema, en la parte superior de la tabla tenemos la sección de búsqueda, donde el usuario podrá filtrar los resultados según le convenga. Para realizar una búsqueda basta con colocarla en la caja de texto de la categoría correspondiente o seleccionar el filtro si es que hay una caja de opciones. En los encabezados de la tabla podremos ordenar los registros según la columna que queramos.

En la parte inferior tenemos un botón con el cual podemos observar más registros si es que hay algunos, si no, este botón no tendrá efecto alguno.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**



The screenshot shows a web application interface for 'Usuarios'. At the top, there is a navigation bar with tabs for 'USUARIOS', 'INVENTARIO', 'PRÉSTAMO', and 'DEVOLUCIÓN'. The user 'Torres, Sergio' is logged in. Below the navigation bar, the title 'Usuarios' is displayed. A table lists 11 users with columns for 'CÓDIGO', 'A. PATERNO', 'A. MATERNO', 'NOMBRE(S)', 'BIBLIOTECA', and 'GRUPO USUARIO'. A blue button at the bottom of the table says 'Ver más resultados.'.

CÓDIGO	A. PATERNO	A. MATERNO	NOMBRE(S)	BIBLIOTECA	GRUPO USUARIO
1234500001	Roach	Miranda	Aidan	Biblioteca 0	Grupo 1
1234500002	Collier	Stanley	Virginia	Biblioteca 1	Grupo 1
1234500003	Peterson	Stephenson	Willow	Biblioteca 0	Grupo 2
1234500004	Davenport	Marshall	Leandra	Biblioteca 0	Grupo 1
1234500005	Perry	Ferguson	Basia	Biblioteca 0	Grupo 1
1234500006	Chase	Golden	Kennan	Biblioteca 0	Grupo 2
1234500008	Cross	Dickson	Tamara	Biblioteca 0	Grupo 2
1234500009	Gay	Ratliff	Regina	Biblioteca 2	Grupo 2
1234500010	Grant	Benton	Alfonso	Biblioteca 1	Grupo 1
1234500011	Collier	Johnston	Cynthia	Biblioteca 1	Grupo 1

Figura 4.30 Tabla de usuarios

Para agregar un nuevo usuario se tendrá que abrir el menú de opciones que se encuentra en la parte superior derecha.

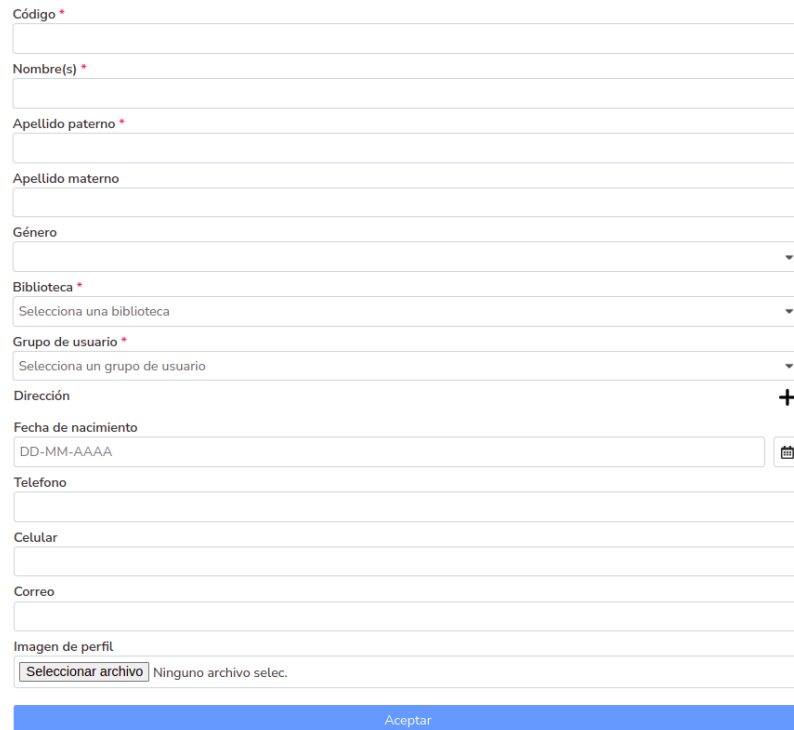


Figura 4.31 Opciones del módulo

Al presionar la opción Nuevo se abrirá el formulario de nuevo usuario en donde deberemos introducir los siguientes datos:

- Código
- Nombre completo
- Genero
- Biblioteca
- Grupo de usuario
- Dirección(es)
- Fecha de nacimiento
- Teléfono
- Celular
- Correo
- Imagen (formato JPG y PNG)

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**



Formulario de registro de un nuevo usuario. Incluye campos para: Código, Nombre(s), Apellido paterno, Apellido materno, Género (lista desplegable), Biblioteca (lista desplegable con 'Selecciona una biblioteca'), Grupo de usuario (lista desplegable con 'Selecciona un grupo de usuario'), Dirección (con botón '+'), Fecha de nacimiento (formato DD-MM-AAAA con calendario), Telefono, Celular, Correo, and Imagen de perfil (botón 'Seleccionar archivo' y 'Ninguno archivo selec.'). Un botón azul 'Aceptar' está al final.

Figura 4.32 Formulario nuevo usuario

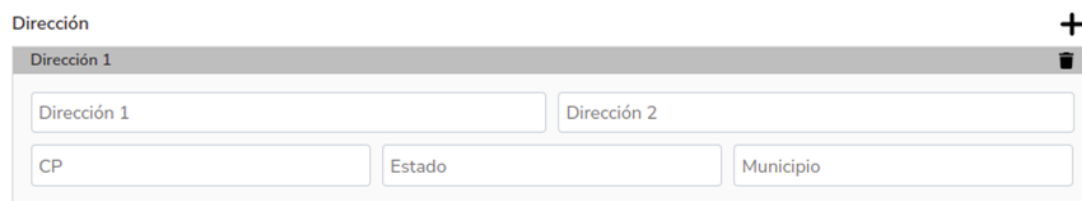
En caso de seleccionar un grupo de usuario marcado como staff, se desplegarán adicionalmente los campos de usuario y contraseña.



Sección de autenticación de usuario. Incluye un campo desplegable 'Grupo de usuario' con 'Grupo 0' seleccionado, y campos de texto para 'Usuario' y 'Contraseña'.

Figura 4.33 Sección de autenticación de usuario

La dirección es un campo especial en donde podremos agregar n direcciones según sea el caso y contara con los campos dirección 1, dirección 2, código postal, estado y municipio.



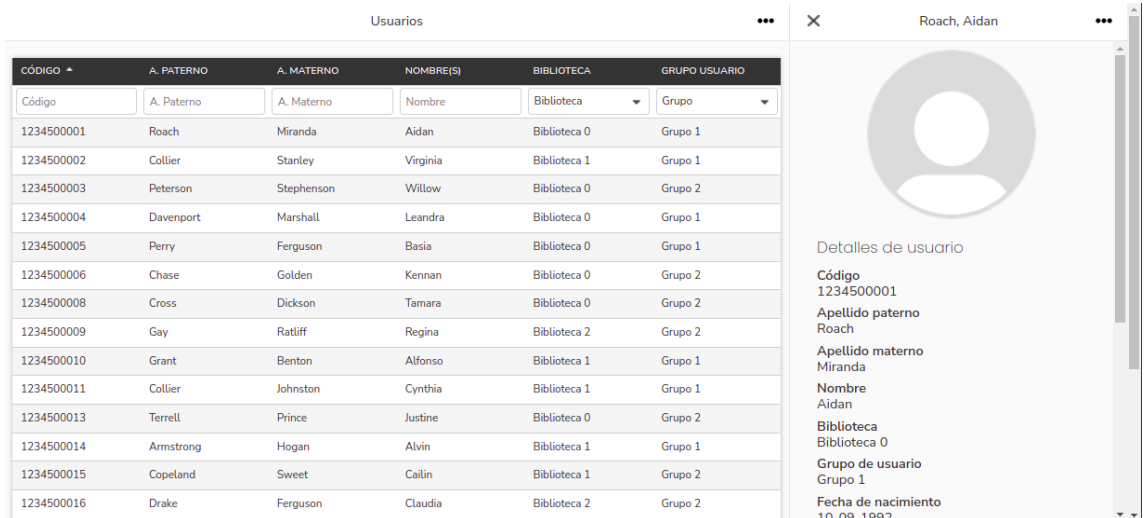
Formulario para agregar una nueva dirección. Incluye un botón '+', un encabezado 'Dirección' con un ícono de basura, y campos para 'Dirección 1', 'Dirección 2', 'CP', 'Estado' y 'Municipio'.

Figura 4.34 Campo de nueva dirección



**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

En la tabla que contiene los registros de puede seleccionar un usuario para ver el detalle el cual contiene la información completa del mismo, así como el historial de préstamos y multas.



The screenshot shows a user management interface. On the left is a table titled 'Usuarios' with columns: CÓDIGO, A. PATERNO, A. MATERNO, NOMBRE(S), BIBLIOTECA, and GRUPO USUARIO. On the right is a detailed view for 'Roach, Aidan' with a profile picture and the following details:

CÓDIGO	A. PATERNO	A. MATERNO	NOMBRE(S)	BIBLIOTECA	GRUPO USUARIO
1234500001	Roach	Miranda	Aidan	Biblioteca 0	Grupo 1
1234500002	Collier	Stanley	Virginia	Biblioteca 1	Grupo 1
1234500003	Peterson	Stephenson	Willow	Biblioteca 0	Grupo 2
1234500004	Davenport	Marshall	Leandra	Biblioteca 0	Grupo 1
1234500005	Perry	Ferguson	Basia	Biblioteca 0	Grupo 1
1234500006	Chase	Golden	Kennan	Biblioteca 0	Grupo 2
1234500008	Cross	Dickson	Tamara	Biblioteca 0	Grupo 2
1234500009	Gay	Ratliff	Regina	Biblioteca 2	Grupo 2
1234500010	Grant	Benton	Alfonso	Biblioteca 1	Grupo 1
1234500011	Collier	Johnston	Cynthia	Biblioteca 1	Grupo 1
1234500013	Terrell	Prince	Justine	Biblioteca 0	Grupo 2
1234500014	Armstrong	Hogan	Alvin	Biblioteca 1	Grupo 1
1234500015	Copeland	Sweet	Cailin	Biblioteca 1	Grupo 2
1234500016	Drake	Ferguson	Claudia	Biblioteca 2	Grupo 2

Detalles de usuario

Código  
1234500001

Apellido paterno  
Roach

Apellido materno  
Miranda

Nombre  
Aidan

Biblioteca  
Biblioteca 0

Grupo de usuario  
Grupo 1

Fecha de nacimiento  
10-09-1992

Figura 4.35 Detalles de usuario

También dispone de un botón para cerrar el detalle y un botón de opciones en donde podremos actualizar o eliminar el registro.

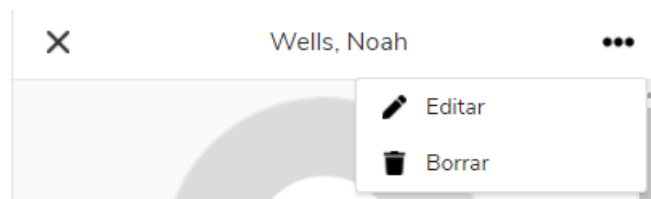


Figura 4.36 Opciones del usuario

Para actualizar un registro tendremos que dar clic en la opción de editar y esto nos desplegará el formulario de nuevo usuario, pero con los datos anteriores.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Código \*  
1234500001

Nombre(s) \*  
Aidan

Apellido paterno \*  
Roach

Apellido materno  
Miranda

Género  
Femenino

Biblioteca \*  
Biblioteca 0

Grupo de usuario \*  
Grupo 1

Dirección +

Dirección 1

Ap #558-7978 Amet St.      Nezahualcóyotl

8932      Estado De México      Municipio

Fecha de nacimiento  
09-09-1992

Telefono  
55-3704 5603

Celular  
55-4715 8129

Correo  
sem.ut@consequat.ca

Imagen de perfil  
Seleccionar archivo | Ninguno archivo selec.

Aceptar

Figura 4.37 Formulario editar usuario

Para eliminar el usuario habrá que dar clic en la opción de eliminar y esto nos desplegará un modal en donde tendremos que confirmar el borrado ya que es una acción delicada.

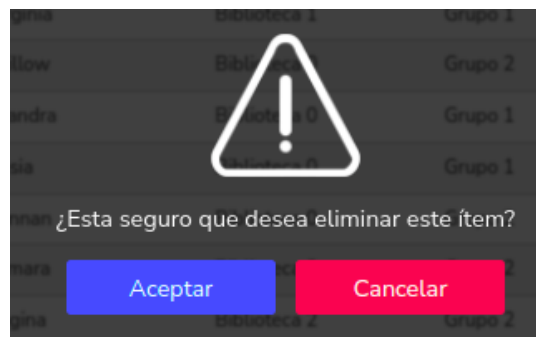


Figura 4.38 Alerta de confirmación

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

El acceso al historial de multas y préstamos se encuentra hasta abajo por lo que tendremos que hacer scroll al panel.



Figura 4.39 Menú acciones del usuario

El botón de préstamos desplegará el panel de historial de préstamos en donde podremos ver el detalle de cada préstamo, así como poder renovar los préstamos activos.

FECHA DE PRÉSTAMO	FECHA DE VENCIMIENTO	FECHA DE DEVOLUCIÓN	RENOVACIONES	CÓDIGO	TÍTULO	
30-11-2020	30-11-2022	-	0	1234500013	Suspendisse sed dolor.	...
30-11-2020	31-01-2021	21-05-2021	0	1234500187	commodo hendrerit. Donec porttitor	...
30-11-2020	31-12-2020	31-01-2021	0	1234500013	Suspendisse sed dolor.	...
30-11-2020	31-01-2021	31-12-2020	1	1234500187	commodo hendrerit. Donec porttitor	...

Figura 4.40 Tabla de préstamos

Para renovar un préstamo tendremos que abrir el menú de opciones del préstamo deseado y dar clic en renovar. El sistema nos mandará un mensaje dependiendo del resultado de nuestra operación, si es que la renovación se realizó correctamente o si es que hubo algún problema. Si el préstamo ya ha sido devuelto el botón de opciones estará deshabilitado.

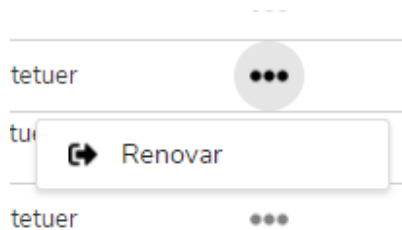


Figura 4.41 Opciones del préstamo

El botón de cuenta en el panel de detalles de usuario despliega el historial de multas del usuario seleccionado. Habrá dos tipos de multas, las generadas por el sistema cuando un ítem es devuelto fuera de tiempo y las multas manuales las cuales podrán ser generadas por el usuario del sistema desde este panel.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**



The screenshot shows a window titled 'Cuenta' with a close button (X) in the top left. In the top right corner, there is a blue button labeled 'Multa manual'. Below this is a table with the following data:

CARGO	PENDIENTE	NOTA	RAZÓN	CÓDIGO	TÍTULO	
10.00	10.00	-	Multa 1	1234500063	rutrum, justo.	...
10.00	10.00	Item Late	-	1234500013	Suspendisse sed dolor.	...
10.00	10.00	Item atrasado	-	1234500187	commodo hendrerit. Donec porttitor	...

Figura 4.42 Tabla de cuenta

Para generar una multa se tiene que seleccionar el botón de multa manual el cual nos desplegara un modal con el formulario. Los datos que deberá llevar es el concepto de la multa (definido en las políticas), el cargo por la multa y opcionalmente se podrá ingresar un código de un ítem si es que la multa se relaciona con el (por ejemplo, un ítem dañado o extraviado) y una nota explicando más del caso.



The screenshot shows a modal window titled 'Multa manual' with a close button (X) in the top right. The form contains the following fields:

- Multa \***: A dropdown menu.
- Cargo \***: A text input field.
- Código**: A text input field.
- Nota**: A text area for notes.

At the bottom of the form is a blue button labeled 'Aceptar'.

Figura 4.43 Formulario multa manual

Para pagar una multa ya sea generada por el sistema o por el usuario, se tiene que abrir el menú de opciones de la multa y presionar pagar. Si es que la multa ya ha sido totalmente pagada el botón de opciones estará deshabilitado.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

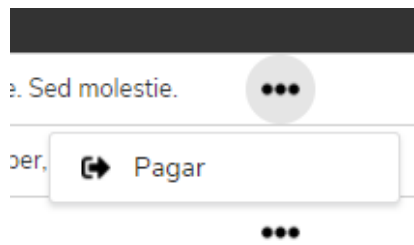


Figura 4.44 Opciones de la cuenta

La opción de pagar nos desplegara el formulario de pago en donde podremos poner una cantidad igual o menor a la cantidad restante por lo que podremos hacer pagos de manera total o parcial.

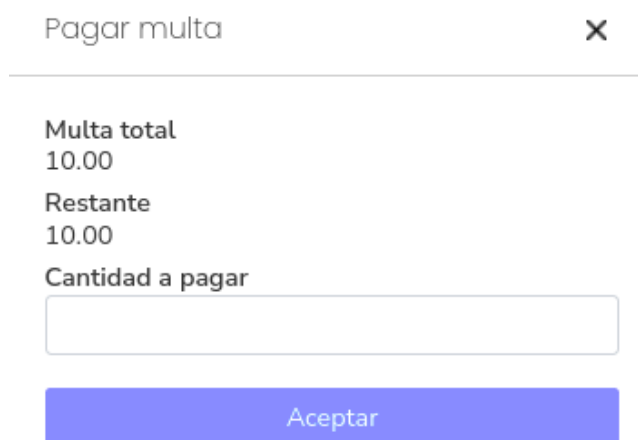
A screenshot of a payment form titled 'Pagar multa' with a close button (X) in the top right corner. The form displays the following information: 'Multa total' with a value of '10.00', and 'Restante' with a value of '10.00'. Below this, there is a label 'Cantidad a pagar' followed by an empty text input field. At the bottom of the form is a blue button labeled 'Aceptar'.

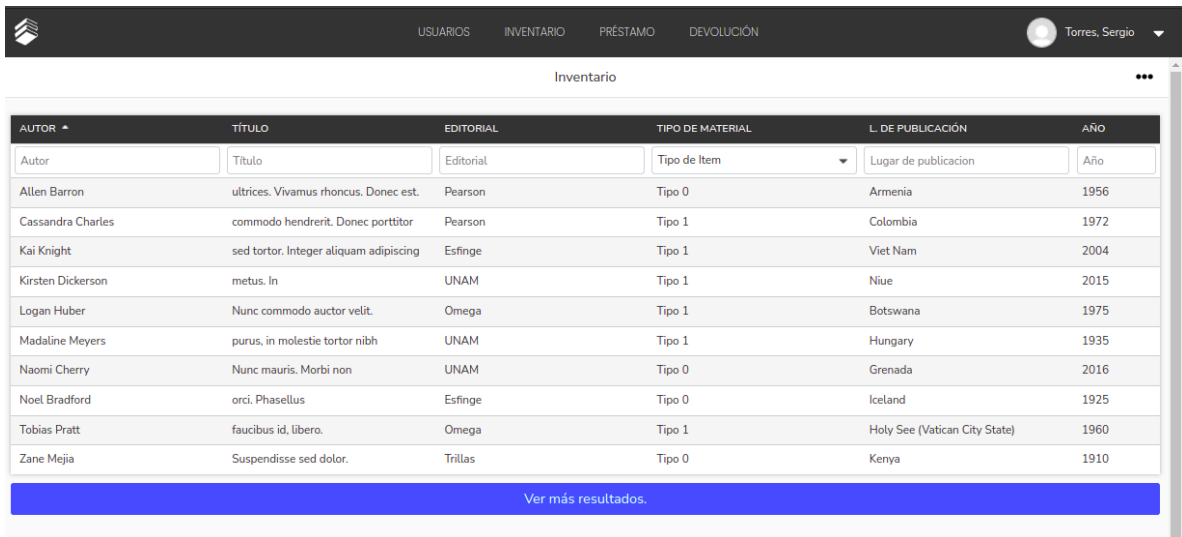
Figura 4.45 Formulario pagar cuenta

### 4.3.3 Inventario

El módulo de inventario está disponible desde la barra de navegación. En esencia es igual al módulo de usuarios, se compone de una tabla en donde se visualizan los registros bibliográficos. Al principio de la tabla tenemos la sección de filtros (que funciona igual que en usuarios), y los encabezados en donde podemos ordenar los datos.

Al final del módulo también disponemos de un botón que nos permite observar más resultados si es que existen.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

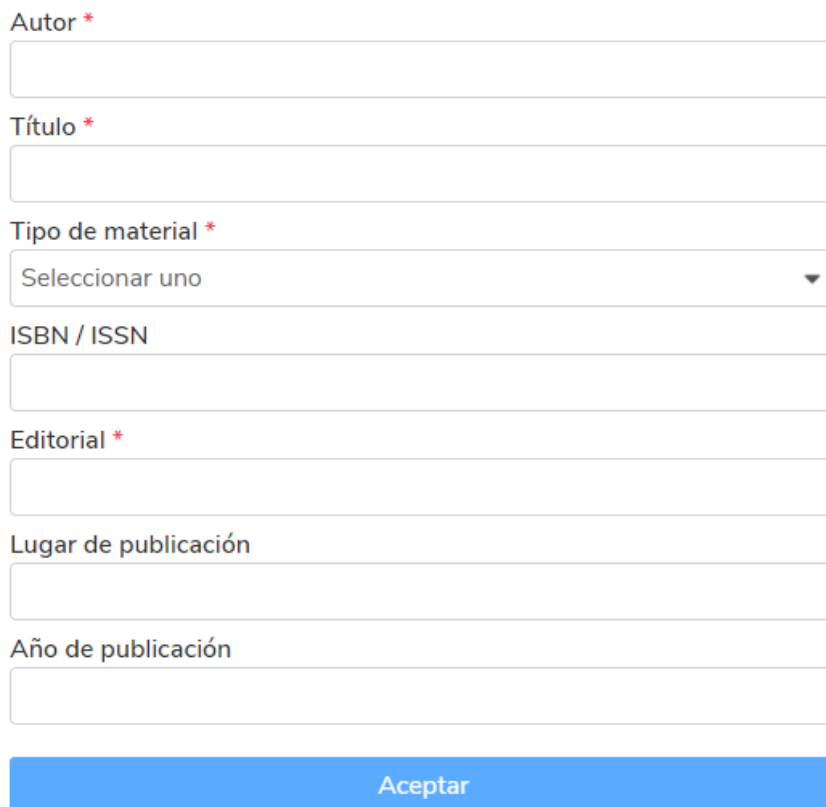


The screenshot shows a web application interface for a library inventory. At the top, there is a navigation bar with the following menu items: USUARIOS, INVENTARIO, PRÉSTAMO, and DEVOLUCIÓN. On the right side of the navigation bar, there is a user profile icon and the name 'Torres, Sergio'. Below the navigation bar, the main heading is 'Inventario'. The main content area displays a table with the following columns: AUTOR, TÍTULO, EDITORIAL, TIPO DE MATERIAL, L. DE PUBLICACIÓN, and AÑO. The table contains 10 rows of data. Below the table, there is a blue button labeled 'Ver más resultados.'.

AUTOR	TÍTULO	EDITORIAL	TIPO DE MATERIAL	L. DE PUBLICACIÓN	AÑO
Allen Barron	ultrices. Vivamus rhoncus. Donec est.	Pearson	Tipo 0	Armenia	1956
Cassandra Charles	commodo hendrerit. Donec porttitor	Pearson	Tipo 1	Colombia	1972
Kai Knight	sed tortor. Integer aliquam adipiscing	Esfinge	Tipo 1	Viet Nam	2004
Kirsten Dickerson	metus. In	UNAM	Tipo 1	Niue	2015
Logan Huber	Nunc commodo auctor velit.	Omega	Tipo 1	Botswana	1975
Madaline Meyers	purus. in molestie tortor nibh	UNAM	Tipo 1	Hungary	1935
Naomi Cherry	Nunc mauris. Morbi non	UNAM	Tipo 0	Grenada	2016
Noel Bradford	orci. Phasellus	Esfinge	Tipo 0	Iceland	1925
Tobias Pratt	faucibus id. libero.	Omega	Tipo 1	Holy See (Vatican City State)	1960
Zane Mejia	Suspendisse sed dolor.	Trillas	Tipo 0	Kenya	1910

Figura 4.46 Tabla de registros bibliográficos

Para agregar un nuevo registro bibliográfico se tendrá que abrir el menú de opciones que se encuentra en la parte superior derecha (figura 4.31).



The screenshot shows a form for adding a new bibliographic record. The form consists of the following fields and controls:

- Autor \***: A text input field.
- Título \***: A text input field.
- Tipo de material \***: A dropdown menu with the text 'Seleccionar uno' and a downward arrow.
- ISBN / ISSN**: A text input field.
- Editorial \***: A text input field.
- Lugar de publicación**: A text input field.
- Año de publicación**: A text input field.
- Aceptar**: A blue button at the bottom of the form.

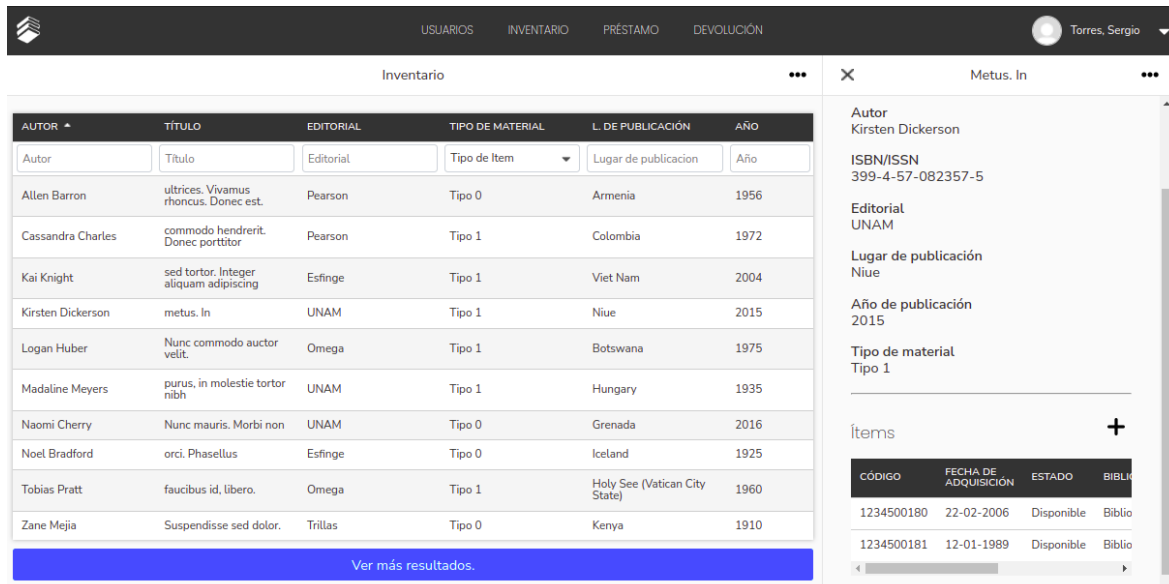
Figura 4.47 Formulario nuevo registro

Al presionar el botón de nuevo nos abrirá el formulario en donde tendremos que agregar los siguientes datos:

## Error! Use the Home tab to apply **Título 1** to the text that you want to appear here.

- Nombre del autor de la obra
- Título de la obra
- Tipo de material bibliográfico (libro, revista, etc.)
- Código ISBN o ISSN según sea el caso
- Editorial de la obra
- Lugar de publicación
- Año de publicación

En la tabla que contiene los registros de puede seleccionar un registro para ver el detalle el cual contiene la información completa del mismo, así como la lista de ejemplares (ítems) asociados.



The screenshot displays a web application interface for a library system. At the top, there are navigation tabs: USUARIOS, INVENTARIO, PRÉSTAMO, and DEVOLUCIÓN. The user is logged in as 'Torres, Sergio'. The main content area is divided into two panels. The left panel, titled 'Inventario', shows a table of bibliographic records with columns for Autor, Título, Editorial, Tipo de Material, L. de Publicación, and Año. The right panel, titled 'Metus. In', provides a detailed view of the selected record for 'Kirsten Dickerson', including fields for Autor, ISBN/ISSN, Editorial, Lugar de publicación, Año de publicación, and Tipo de material. Below these details is a section for 'Ítems' (copies) with a table listing individual items with columns for Código, Fecha de Adquisición, Estado, and Bibliotecario.

AUTOR	TÍTULO	EDITORIAL	TIPO DE MATERIAL	L. DE PUBLICACIÓN	AÑO
Allen Barron	ultrices. Vivamus rhoncus. Donec est.	Pearson	Tipo 0	Armenia	1956
Cassandra Charles	commodo hendrerit. Donec porttitor	Pearson	Tipo 1	Colombia	1972
Kai Knight	sed tortor. Integer aliquam adipiscing	Esfinge	Tipo 1	Viet Nam	2004
Kirsten Dickerson	metus. In	UNAM	Tipo 1	Niue	2015
Logan Huber	Nunc commodo auctor velit.	Omega	Tipo 1	Botswana	1975
Madaline Meyers	purus. in molestie tortor nibh	UNAM	Tipo 1	Hungary	1935
Naomi Cherry	Nunc mauris. Morbi non	UNAM	Tipo 0	Grenada	2016
Noel Bradford	orci. Phasellus	Esfinge	Tipo 0	Iceland	1925
Tobias Pratt	faucibus id. libero.	Omega	Tipo 1	Holy See (Vatican City State)	1960
Zane Mejia	Suspendisse sed dolor.	Trillas	Tipo 0	Kenya	1910

CÓDIGO	FECHA DE ADQUISICIÓN	ESTADO	BIBLI
1234500180	22-02-2006	Disponible	Biblio
1234500181	12-01-1989	Disponible	Biblio

Figura 4.48 Detalles del registro bibliográfico

Al igual que el detalle en el módulo de usuarios, dispone de un botón para cerrar el detalle y un botón de opciones en donde podremos actualizar o eliminar el registro (figura 4.36).

Al presionar el botón de actualizar nos desplegará el formulario de nuevo registro, pero con los datos del registro a modificar.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Autor \*

Título \*

Tipo de material \*

ISBN / ISSN

Editorial \*

Lugar de publicación

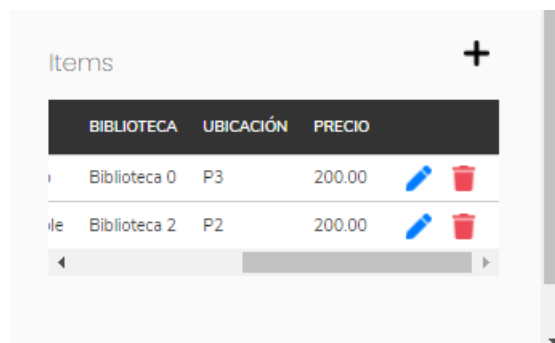
Año de publicación

Aceptar

Figura 4.49 Formulario editar registro

Para eliminar el registro habrá que dar clic en la opción de eliminar y posteriormente confirmar la acción (figura 4.38).

El panel de detalles nos muestra una sección en donde podremos administrar los ejemplares asociados a el registro seleccionado.







BIBLIOTECA	UBICACIÓN	PRECIO		
Biblioteca 0	P3	200.00		
Biblioteca 2	P2	200.00		

Figura 4.50 Tabla ítems



**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Para agregar un nuevo ítem del registro bibliográfico se tendrá que presionar el botón con el símbolo de más. Esto nos abrirá un formulario en donde tendremos que insertar los siguientes datos:

- Código del ítem (puede ser un código de barras).
- Fecha de adquisición del ítem.
- Biblioteca en donde se encuentra físicamente.
- Ubicación (Piso o estantería).
- Precio.




Nuevo ítem ×

Código \*

Fecha de adquisición

Biblioteca \*

Ubicación

Precio

Aceptar

*Figura 4.51 Formulario nuevo ítem*

La tabla en donde se encuentran los ítems tiene una columna de acciones en donde podremos actualizar o eliminar el ítem deseado.

Para actualizar el ítem tendremos que seleccionar el botón con el ícono de un lápiz lo cual nos abrirá el formulario de nuevo ítem, pero con los datos del ítem seleccionado.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Editar ítem ✕

Código \*  
1234500180

Fecha de adquisición  
22-02-2006 📅

Biblioteca \*  
Biblioteca 1 ▾

Ubicación  
PB

Precio  
200

Aceptar

Figura 4.52 Formulario editar ítem

Para eliminar el ítem se tendrá que seleccionar el botón con el ícono de basura y después tendremos que confirmar la acción (figura 4.38).

#### 4.3.4 Préstamo

Se puede acceder al módulo de préstamo mediante la barra de navegación, el módulo se compone de 2 secciones. La del lado izquierdo que es donde tenemos que seleccionar al usuario al que deseamos prestar y en el lado derecho el código de los n ítems que deseamos prestar al usuario. Si queremos cambiar de usuario debemos limpiar el panel con el botón de check en la parte superior derecha.

Usuario Ítems prestados ✓

Código del usuario Seleccionar

Código de barras del ítem Seleccionar

No hay ítems

Figura 4.53 Panel de préstamo

## Error! Use the Home tab to apply Título 1 to the text that you want to appear here.

Si el código de usuario que introducimos en el panel de usuario corresponde con un usuario existente, se mostrara un pequeño detalle donde podremos ver el número de préstamos activos del usuario y la cantidad que debe en multas.

The screenshot shows a user details panel with two main sections: 'Usuario' and 'Ítems prestados'. The 'Usuario' section contains a search box with the value '1234500199' and a 'Seleccionar' button. Below this, the user's name 'Hansen Carver Nathaniel' is displayed, along with their 'Código' '1234500199'. A table shows 'Grupo' as 'Grupo 1' and 'Biblioteca' as 'Biblioteca 0'. Another table shows 'Préstamos' as '-' and 'Multas' as '0.00'. The 'Ítems prestados' section has a search box for 'Código de barras del ítem' and a 'Seleccionar' button. Below this, it says 'No hay ítems'.

Figura 4.54 Detalles del usuario seleccionado

Luego de seleccionar el usuario se nos habilitará el botón de seleccionar ítem. En este panel podremos seleccionar n ítems siempre y cuando se cumpla la condición de que el usuario no sobrepase el número de préstamos permitidos por las políticas de circulación, el usuario no tenga bloqueos activos por devoluciones fuera de tiempo y si el ítem se encuentra disponible para préstamo.

Si el préstamo no es posible se mostrará un mensaje de error al usuario.

The screenshot shows the 'Ítems prestados' panel. The 'Usuario' section is identical to the previous figure. The 'Ítems prestados' section has a search box with the value '1234500015' and a 'Seleccionar' button. Below this, a table lists the following items:

CÓDIGO	FECHA DE VENCIMIENTO	TÍTULO
1234500010	31-05-2022	rhoncus. Nullam
1234500011	31-05-2022	rhoncus. Nullam
1234500012	31-05-2022	Suspendisse sed dolor.
1234500013	24-05-2022	Suspendisse sed dolor.
1234500014	31-05-2022	pulvinar arcu et pede.

At the bottom right, there is a notification area with three messages:

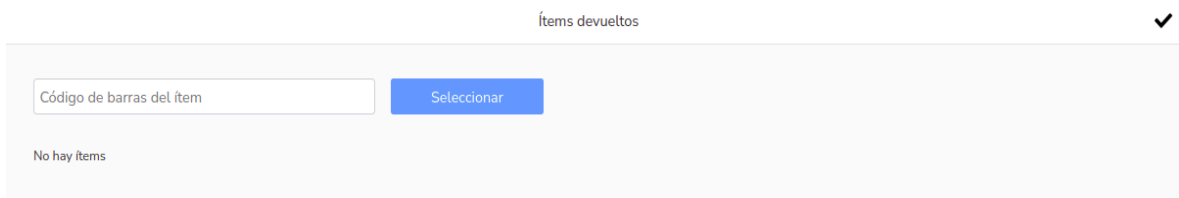
- Éxito: Ítem prestado correctamente. (Close button)
- Éxito: Ítem prestado correctamente. (Close button)
- Error: No se pueden hacer mas préstamos. (Close button)

Figura 4.55 Detalles de ítems prestados

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

### 4.3.5 Devolución

El módulo de devolución de encuentra junto al módulo de préstamo en la barra de tareas, consta de un panel en donde podremos introducir el código del ítem a devolver. También tenemos un botón de check en la parte superior derecha en donde podremos limpiar el panel.



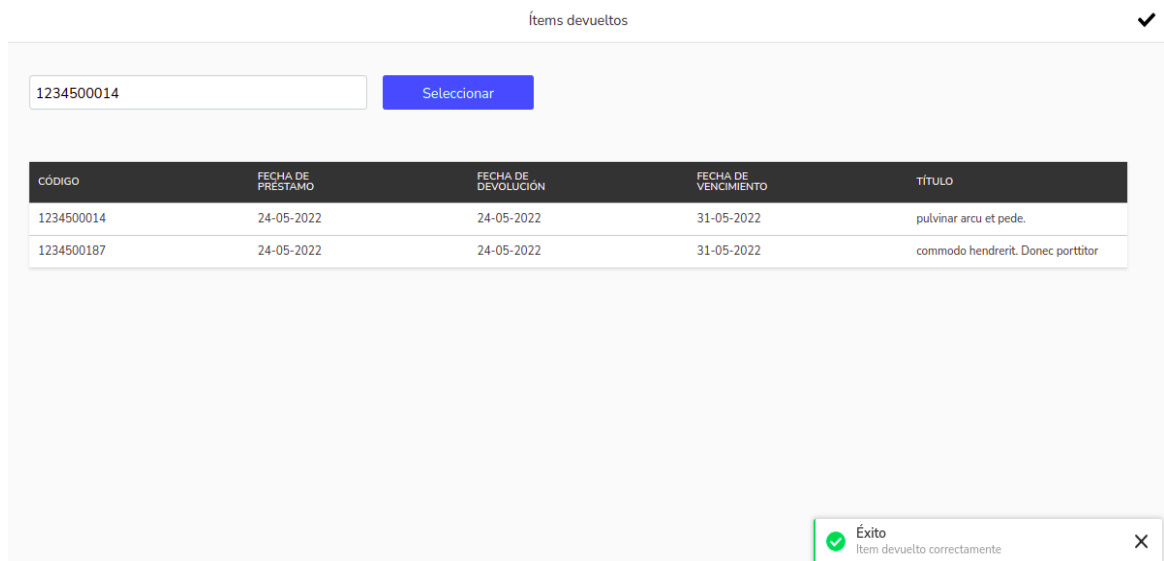
Ítems devueltos ✓

Código de barras del ítem  Seleccionar

No hay ítems

Figura 4.56 Panel de devolución

Al introducir un código de ítem en la caja de texto y luego dar en seleccionar el ítem quedará devuelto siempre y cuando se cumpla la condición de que este ítem se encuentre en un estado de préstamo.



Ítems devueltos ✓

1234500014 Seleccionar

CÓDIGO	FECHA DE PRÉSTAMO	FECHA DE DEVOLUCIÓN	FECHA DE VENCIMIENTO	TÍTULO
1234500014	24-05-2022	24-05-2022	31-05-2022	pulvinar arcu et pede.
1234500187	24-05-2022	24-05-2022	31-05-2022	commodo hendrerit. Donec porttitor

Éxito  
Item devuelto correctamente

Figura 4.57 Detalle ítems devueltos

Si el sistema detecta que el ítem fue devuelto fuera de tiempo se generara una multa y/o un bloqueo de acuerdo con lo establecido en las políticas de circulación.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

### 4.3.6 Configuración

Se accede al módulo de configuración por medio del botón de opciones de la barra de navegación (figura 4.29), este módulo consta de un panel de navegación en donde seleccionaremos el tipo de configuración que queremos modificar.

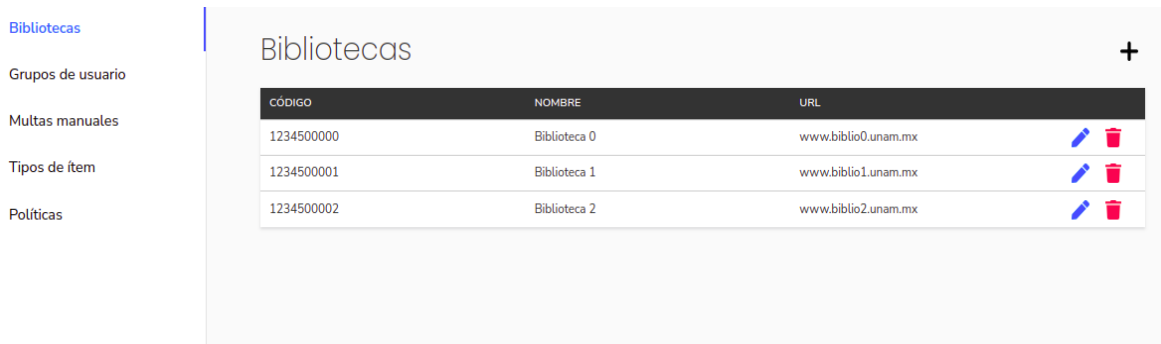


Figura 4.58 Panel configuración y tabla de bibliotecas

Para crear una nueva configuración tendremos que hacer clic en el botón de más del lado derecho del título. Esto nos desplegará el formulario en donde tendremos que llenar los datos correspondientes a cada tipo de configuración.

Crear biblioteca ×

**Código \***

**Nombre \***

**URL**

**Aceptar**

Figura 4.59 Formulario nueva configuración

Para editar un registro se tendrá que seleccionar el ícono de lápiz al lado derecho del registro que se quiere modificar. Esto nos desplegará el formulario correspondiente previamente llenado con los datos del registro.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Editar biblioteca ×

**Código \***  
1234500000

**Nombre \***  
Biblioteca 0

**URL**  
www.biblio0.unam.mx

Aceptar

Figura 4.60 Formulario editar configuración

Para eliminar un registro, se tiene que seleccionar el ícono de bote de basura lo que nos mostrara un modal de confirmación (figura 4.38).

En la configuración de grupos de usuario tendremos el mismo diseño con la diferencia de que el formulario tendrá capos distintos de tipos distintos.

Bibliotecas  
Grupos de usuario  
Multas manuales  
Tipos de ítem  
Políticas

CÓDIGO	NOMBRE	BIBLIOTECA	STAFF	
1234500000	Grupo 0	-	Si	
1234500001	Grupo 1	-	No	
1234500002	Grupo 2	-	No	

Figura 4.61 Tabla de grupos de usuario

En cuanto a las multas manuales y los tipos de ítem, se aplica el mismo diseño que en grupo de usuarios y bibliotecas. En donde cambia ligeramente es en políticas ya que los datos completos no caben en la tabla.

Bibliotecas  
Grupos de usuario  
Multas manuales  
Tipos de ítem  
Políticas

NOMBRE	DESCRIPCIÓN	
Política 0	-	

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

Figura 4.62 Tabla políticas

Para solucionar este problema se agregó un panel de detalle que se expande cuando se selecciona el registro.



The screenshot shows a web application interface. On the left is a sidebar with navigation links: Bibliotecas, Grupos de usuario, Multas manuales, Tipos de ítem, and Políticas (highlighted in blue). The main content area is titled 'Políticas' and contains a table with the following data:

NOMBRE	DESCRIPCIÓN	
Política 0	-	 
Biblioteca	Préstamos	Sanción
-	5	7 día(s)
Grupo de usuario	Periodo de préstamo	Renovaciones
-	7 día(s)	2
Tipo de ítem	Multa	Periodo de gracia
-	10.00	1 hora(s)

Figura 4.63 Tabla detalle de política

## Capítulo V. Pruebas y mantenimiento

### 5.1 Implantación

#### Prerrequisitos

Para poder alojar una copia funcional del sistema es necesario contar con un servidor que tenga instalado lo siguiente:

- NodeJS
- PostgreSQL
- Npm

#### Primeros pasos

Para configurar el api inicialmente se tendrá que clonar el repositorio de Git, con el siguiente comando se descargará una copia del proyecto en una máquina local

```
git clone https://github.com/luis-tapiaa/circula.git
```

Después de que el proyecto este descargado mediante la terminal se deberá instalar las dependencias, con el siguiente comando

```
npm install
```

En la raíz del proyecto se debe crear un archivo con el nombre `.env`, dentro de este archivo estarán todas las variables entorno del sistema. Deberán incluirse las siguientes variables:

Opción	Descripción
<b>CLOUDINARY_API_KEY</b>	Colocar el API KEY de la cuenta de cloudinary
<b>CLOUDINARY_API_SECRET</b>	Colocar el API SECRET de la cuenta de cloudinary
<b>CLOUDINARY_NAME</b>	Colocar el nombre de la cuenta de cloudinary



**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

<b>DURATION</b>	Colocar la duración del token
<b>POSTGRES_URL</b>	Colocar la URL de la base de datos
<b>TOKEN_KEY</b>	Colocar la clave para el token

*Tabla 5.1 Variables de entorno API*

El formato para la URL de PostgreSQL deberá contener el host, el usuario, el password, el puerto y la base de datos.

```
postgresql://[usuario[:password]@][host][:puerto]/[db]
```

Para la configuración de la base de datos se debe crear un usuario en el servidor con el mismo nombre, y posteriormente se debe migrar la base de datos. En la raíz del proyecto está ubicado el script de la base de datos el cual se ejecuta con

```
pg_dump query.sql
```

El script contiene datos iniciales los cuales van a permitir utilizar el sistema por primera vez. Una vez realizados los pasos anteriores se podrá levantar el servidor con el siguiente comando

```
npm start
```

Para instalar el servidor web se tiene que copiar el repositorio con el siguiente comando

```
git clone https://github.com/luis-tapiaa/tesis.git
```

Después de descargar el proyecto, mediante la terminal se deberá instalar las dependencias con el comando

```
npm install
```

Luego se tiene que compilar el proyecto para crear una versión que pueda ser utilizada en producción usando el comando

```
npm run build
```

Esto nos generara una carpeta llamada build la cual podremos desplegar en cualquier servidor de nuestra preferencia.

## **5.2 Pruebas.**

Para poder realizar las pruebas se tuvo que instalar el sistema en un servidor de desarrollo y se ejecutaron diversas pruebas que consistieron en lo siguiente:

### **5.2.1 Operaciones de escritura desde el servidor de base de datos**

- Se dieron de alta bibliotecas, las cuales permiten crear usuarios y grupos de usuarios.
- Agregar grupos de usuario, lo cual permite el manejo de privilegios dentro del sistema en usuarios.
- Dar de alta usuarios, con los cuales permite crear préstamos y verificar que el código de usuarios.
- Se debe añadir registros bibliográficos para poder dar de alta ejemplares.
- Se dieron de alta ítems, lo cual permite hacer préstamos, devoluciones y renovaciones.
- Dar de alta políticas, permitirá definir las restricciones de los préstamos, devoluciones y renovaciones.
- Dar de alta préstamos, para poder realizar préstamos y devoluciones editando el registro.
- Se dieron de alta los conceptos de las multas en caso de incumplimiento de las políticas.
- Una vez con creados estos registros ya se puede proceder a crear registros de préstamos.

Durante la ejecución se encontraron errores en cuanto a las estructuras de los registros, por ejemplo, al momento de eliminar una biblioteca se eliminaban por consecuencia todos los registros que estuviesen asociados a dicha biblioteca, este error era consecuencia de que la estructura estaba en la restricción CASCADE, por lo que se procedió a editar las tablas para que estuviesen por defecto.

### **5.2.2 Operaciones de escritura desde el sistema CIRCULA**

- Se hicieron varios intentos en el módulo de inicio de sesión para verificar que el sistema no de acceso a usuarios sin privilegios, así como se probó que la sesión se almacene dentro del navegador y tenga la duración elegida.
- Se hicieron búsquedas de usuarios y registros bibliográficos para verificar que se pueda encontrar algún registro solo con los datos indispensables.
- Se realizaron operaciones de escritura como creación, edición, y eliminación sobre los usuarios, registros bibliográficos e ítems.
- Se realizaron diversos prestamos, devoluciones y renovaciones para verificar que las políticas se apliquen de manera correcta.
- Se realizaron devoluciones extemporáneas para verificar que las multas y bloqueos sean aplicados de manera correcta según lo estipulado en las políticas.
- Se realizaron diversas operaciones de lectura y escritura sobre las configuraciones del sistema las cuales incluyen las bibliotecas, grupo de usuario, tipos de ítem, multas manuales y políticas de circulación.

Todas las pruebas realizadas anteriormente fueron cumplidas exitosamente, pero se realizaron modificaciones en la interfaz del sistema para que estas operaciones fueran implementadas de manera más intuitiva y amigable con los usuarios

### **5.2.3 Concurrencia**

Las pruebas de concurrencia realizadas fueron con 5 usuarios concurrentes, dichos usuarios voluntarios a los cuales se les dio una capacitación del uso del software, durante el proceso el sistema se comportó de manera eficaz y rápida, cumplimiento la integridad y la confidencialidad de los registros. Con este bloque cumplido se tiene la expectativa de que con hardware más vigoroso podemos lograr el objetivo de 20 usuarios concurrentes empleando el software.

## **5.3 Mantenimiento**

Durante la creación del proyecto, de los diferentes módulos y componentes. El sistema fue sufriendo cambios, ya que al estar realizando el proyecto se

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

encontraban errores, por lo que durante este proceso se le dio un mantenimiento correctivo y perfectivo al sistema.

En cuanto el proyecto este montado se le dará al usuario una capacitación sobre el uso del sistema, además se le proporcionará el código en caso de que a futuro se realice una nueva versión del sistema por si se desea mejorar el rendimiento o en caso de que se requieran nuevos módulos, se puedan integrar de manera simple.

El mantenimiento de la base de datos se recomienda efectuar cada semestre para verificar la integridad de los registros y se debe crear un respaldo de la base de datos para cualquier escenario que se pudiese presentar ajeno al sistema.

## **Conclusiones**

Las bibliotecas deben de seguir un proceso de evolución, por lo que necesitan renovarse cada cierto periodo de tiempo. En este caso se llevó a cabo el desarrollo del software para automatizar el proceso de circulación bibliográfica.

La circulación bibliográfica consiste en realizar prestamos del material bibliográfico, así como la renovación de estos, su devolución y reserva. Los prestamos pueden ser a domicilio, interbibliotecas y para su consulta en sala. La circulación bibliográfica también comprende la administración de multas y sanciones en caso del incumplimiento de las políticas de circulación por parte de los usuarios de la biblioteca.

Las bibliotecas de la UNAM viven con la problemática de que el sistema que utilizan para llevar a cabo la administración de la circulación bibliográfica y otras tareas propias de una biblioteca es un sistema de uso comercial (Aleph) por el cual se llegan a pagar hasta 5 millones de pesos para poder dar este servicio.

Por esa razón se propuso la modernización tecnológica en el servicio de circulación bibliográfica por medio del desarrollo de un software con la finalidad de disminuir costos cubriendo las necesidades del procedimiento de circulación bibliográfica.

El objetivo principal del presente trabajo es cubierto porque se empleó tecnología de última generación como es ReactJS, GraphQL y Express.

Para el logro de este fin se tuvo que hacer el levantamiento del sistema a través de una serie de entrevistas a personal bibliotecario y de sistemas de la Dirección General de Bibliotecas de la UNAM con lo cual se pudo detectar la necesidad y la importancia de la circulación bibliográfica en la UNAM.

Se utilizó el método de desarrollo en cascada ya que al comenzar el desarrollo se definieron claramente los requisitos y al ser sólo un módulo el desarrollo no fue tan complejo.

Con el desarrollo del sistema se pretende disminuir considerablemente los costos del mantenimiento del software ya que para este fin se podría instalar este software

en el sistema bibliotecario de la UNAM o en las bibliotecas que así lo requieran, con inversiones mínimas de aproximadamente 26 mil pesos que sería la inversión económica para poderlo implementar.

En las bibliotecas se generan repostes del uso del sistema, pero no se analizan los datos que se generan con el uso del sistema, por esta razón un siguiente alcance para este sistema podría ser la aplicación de minería de datos para la obtención de conocimiento a partir del uso del sistema o inteligencia artificial para garantizar y mejorar los requerimientos de los usuarios de las bibliotecas.

En un futuro también se pueden implementar las siguientes características las cuales, si bien no son indispensables, ayudarían a mejorar el servicio que se puede proporcionar por medio de este sistema.

- **Modificación masiva de usuarios e inventario:** Permitir la modificación masiva de registros a través de archivos CSV ya que la única manera de almacenar registros de manera masiva es a través de scripts de la base de datos.
- **Consulta del estado de los prestamos:** Crear un módulo en donde los usuarios no administradores puedan consultar el estado de sus préstamos y multas.
- **Reserva de material bibliográfico:** Permitir la reserva de materiales para que el usuario pueda acceder primero a ellos cuando estén disponibles.

Finalmente, el sistema cubre el objetivo planteado y puede ser implementado en el sistema bibliotecario de la UNAM.

## Referencias

- Biblioteca Universitaria. (2008). Sistema ALEPH 500. *Biblioteca Universitaria*, 49-56.
- Cardador, A. L. (2014). *Desarrollo de aplicaciones web distribuidas*. Andalucía: IC Editorial.
- Dirección General de Bibliotecas. (29 de Agosto de 2017). *Acerca de la Dirección General de Bibliotecas*. Obtenido de Dirección General de Bibliotecas: <http://dgb.unam.mx/index.php/quienes-somos/acerca-de-la-dgb>
- Drasner, S. (21 de Enero de 2020). *Advice from a GraphQL Expert*. Obtenido de Netlify: <https://www.netlify.com/blog/2020/01/21/advice-from-a-graphql-expert/>
- Granados, L. (2014). *Desarrollo de aplicaciones web en el entorno servidor*. Andalucía: IC Editorial.
- Leyva Ramírez, E. (2019). *Memoria UNAM 2019*. Ciudad de México: UNAM.
- Marqués, M. (2011). *Bases de datos*. Castelló: Universitat Jaume I de Castelló.
- Mozilla Foundation. (23 de Marzo de 2019). *DOM*. Obtenido de Documentacion web de MDN: <https://developer.mozilla.org/es/docs/DOM>
- Mozilla Foundation. (6 de Agosto de 2020). *Introduction to client-side frameworks*. Obtenido de Learn web development: [https://developer.mozilla.org/en-US/docs/Learn/Tools\\_and\\_testing/Client-side\\_JavaScript\\_frameworks/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction)
- Mozilla Foundation. (8 de Agosto de 2020). *Tecnología para desarrolladores web*. Obtenido de Documentación Web de MDN: <https://developer.mozilla.org/es/docs/Web>
- Mulay, P., Paliwal, S., Iyengar, V., Pillai, S., & Rao, A. (2019). Resolving multiple copies problem in unique-titles from biblio-records available through KOHA library management system. *Library Hi Tech News*, 12-16.

Oppel, A., & Sheldon, R. (2010). *Fundamentos de SQL*. Ciudad de México: McGraw-Hill.

Pascual, J. R. (1 de Diciembre de 2019). *Arquitectura de Servicios Web*. Obtenido de Disrupción Tecnológica: <https://www.disrupciontecnologica.com/arquitectura-de-servicios-web>

PostgreSQL. (18 de Abril de 2018). *PostgreSQL: Acerca de*. Obtenido de PostgreSQL: <https://www.postgresql.org/about/>

Pressman, R. (2010). *Ingeniería del software Un enfoque Práctico*. Ciudad de México: McGraw Hill.

REST. (9 de Octubre de 2015). Obtenido de Sistemas Distribuidos: <http://dsanchezzz.blogspot.com/2015/10/rest.html>

Ricardo, C. M. (2009). *Bases de datos*. Ciudad de México: McGraw-Hill.

Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos*. Madrid: McGraw Hill.

Villarello Reza, R. (2011). A 100 años de la fundación de la UNAM: un sumario de la historia de su Sistema Bibliotecario. *Biblioteca Universitaria*, 123-143.

Voutssás, J. (1987). Definición de un sistema de circulación o préstamo automatizado para una biblioteca. *Boletín informativo de la Dirección General de Bibliotecas UNAM*, 10-19.

Voutssás, J. (2019). *Los inicios de la automatización de bibliotecas en México*. Ciudad de México: UNAM.