

# **Proyecto Ing.Web Carrito de Compras**

## **Documentación Guía N°6**

Juan Jose Guerrero

Mikey Collazos

Asly Camelo

Sebastian Lopez Alvarez

Juan Sebastian Mendoza

Profesora Diana Marcela Toquica

29 de Abril de 2025

Universidad Manuela Beltran

Facultad Ingeniería de Software

## Resumen

“Entre Líneas” es una innovadora plataforma de comercio electrónico centrada en la venta de libros en línea, diseñada para ofrecer una experiencia de usuario fluida, moderna y eficiente. Este proyecto no solo busca digitalizar el acceso a la literatura, sino también optimizar cada etapa del proceso de compra mediante herramientas avanzadas como motores de búsqueda inteligentes, filtros dinámicos y un sistema de pedidos intuitivo.

La construcción del sistema se organiza estratégicamente en cuatro equipos especializados, garantizando un desarrollo integral y colaborativo:

El equipo Front-End transforma las ideas en interfaces visuales interactivas y atractivas, priorizando la usabilidad y el diseño responsivo para múltiples dispositivos.

El equipo Back-End se encarga de la estructura lógica y el manejo de datos, implementando una base de datos relacional robusta y segura que respalda toda la operación del sitio.

El equipo de Integración actúa como puente entre el Front-End y el Back-End, desarrollando APIs que aseguran una comunicación eficiente, fluida y en tiempo real entre ambos entornos.

El equipo de documentación es el guardián del conocimiento técnico del proyecto. Este grupo elabora tanto los manuales de usuario como los documentos técnicos y funcionales, incluyendo la arquitectura del sistema, rutas y procesos implementados.

Con un enfoque claro en la escalabilidad, el rendimiento y la experiencia del usuario, “Entre Líneas” se proyecta como una solución robusta para el mercado digital de libros, abriendo nuevas oportunidades de acceso a la lectura y fomentando el hábito lector en la era tecnológica.

## Abstract

A web-based e-commerce project focused on a bookstore is being designed and developed. This platform includes search bars, filters, the ability to purchase books, order tracking, and a comprehensive book listing. The development is carried out by four teams:

- The **Front-End team** is responsible for designing and building the graphical user interface of the website.
- The **Back-End team** manages data storage using a relational database and handles data processing.
- The **Integration team** connects the Front-End and Back-End through APIs, enabling seamless data transfer.
- The **Documentation team** is in charge of creating both the User and Technical manuals, as well as the general system documentation, including specifications and defined routes.

## Índice General

<b>1 INTRODUCCIÓN</b>	<b>7</b>
<b>2 DEFINICIÓN PROYECTO</b>	<b>7</b>
2.1 Objetivos del proyecto	7
2.1.1 Objetivo General	7
2.1.2 Objetivos Específicos	7
2.2 Ambiente de Ingeniería de Software	7
<b>3 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE</b>	<b>8</b>
3.1 Alcances	8
3.2 Objetivo del software	8
3.3 Descripción Global del Producto	8
3.3.1 Interfaz de usuario	8
3.3.2 Interfaz De Hardware	8
3.3.3 Interfaces de comunicación	9
3.4 Requerimientos Específicos	9
3.4.1 Requerimientos Funcionales del sistema	9
<b>4 DISEÑO</b>	<b>44</b>
4.1 Diseño de Físico de la Base de datos	44
4.2 Diseño de arquitectura funcional	46
4.3 Diseño interfaz y navegación	46
4.4 Especificación de módulos	46
<b>5 PRUEBAS</b>	<b>47</b>
5.1 Elementos de prueba	47
5.2 Especificación de las pruebas	47
5.3 Responsables de las pruebas	48
5.4 Calendario de pruebas	48
5.5 Detalle de las pruebas	48
5.6 Conclusiones de Prueba	48
<b>6 CONCLUSIONES</b>	<b>48</b>
<b>7 BIBLIOGRAFÍA</b>	<b>49</b>
<b>8 ANEXO: MANUAL DE USUARIO</b>	<b>49</b>

## Índice Tablas

<b>Tabla 1: Requerimiento Funcional Crear Cuenta</b>	<b>10</b>
<b>Tabla 2: Requerimiento Funcional Iniciar Sesión</b>	<b>12</b>
<b>Tabla 3: Requerimiento Funcional Editar Perfil del Usuario</b>	<b>14</b>
<b>Tabla 4: Requerimiento Funcional Barra De Búsqueda</b>	<b>15</b>
<b>Tabla 5: Requerimiento Funcional Lista de Libros (Administrador)</b>	<b>17</b>
<b>Tabla 6: Requerimiento Funcional Filtros</b>	<b>18</b>
<b>Tabla 7: Requerimiento Funcional Crear categorías/filtros (Administrador)</b>	<b>20</b>
<b>Tabla 8: Requerimiento Funcional Recomendaciones</b>	<b>21</b>
<b>Tabla 9: Requerimiento Funcional Información del Libros</b>	<b>23</b>
<b>Tabla 10: Requerimiento Funcional Gestión de libros (Usuario)</b>	<b>24</b>
<b>Tabla 11: Requerimiento Funcional Información del libro (Usuario)</b>	<b>26</b>
<b>Tabla 12: Requerimiento Funcional Editar detalles del libro (Administrador)</b>	<b>29</b>
<b>Tabla 13: Requerimiento Funcional Agregar libros (Administrador)</b>	<b>30</b>
<b>Tabla 14: Requerimiento Funcional Eliminar libros (Administrador)</b>	<b>32</b>
<b>Tabla 15: Requerimiento Funcional Cantidad a comprar</b>	<b>33</b>
<b>Tabla 16: Requerimiento Funcional Carrito de compras</b>	<b>35</b>
<b>Tabla 17: Requerimiento Funcional Entrega</b>	<b>36</b>
<b>Tabla 18: Requerimiento Funcional Historial de compra</b>	<b>39</b>
<b>Tabla 19: Requerimiento Funcional Buscar en el registro de compra (Administrador)</b>	<b>40</b>
<b>Tabla 20: Requerimiento Funcional Reseñas</b>	<b>42</b>
<b>Tabla 21: Requerimiento Funcional Procesamiento de pagos</b>	<b>43</b>

## Índice Figuras

## **1 INTRODUCCIÓN**

---

Se diseña y desarrolla un proyecto de comercio web basado en una librería, esta consta de barras de búsqueda, filtros, posibilidad de compras de libros, pedidos y listado de libros, esta se realiza por medio de 4 equipos, el equipo encargado del Front-End diseñará y construirá la parte gráfica de la página, Back-End se encarga del alojamiento de datos en base de datos relacional y traspaso de los mismos, el grupo de Integración se encarga por medio de APIs transferir los datos y unir el Front con el Back y el grupo de Documentación encargado de generar los manuales de Usuario y Técnico, además del archivo general con las especificaciones del sistema y sus rutas creadas.

---

## **2 DEFINICIÓN PROYECTO**

---

### **2.1 Objetivos del proyecto**

#### **2.1.1 Objetivo General**

El objetivo del proyecto es desarrollar una plataforma web de un carrito de compras llamado “Entre líneas” la cual se basará en la venta de libros online, brindando una experiencia de usuario, basado en herramientas de búsqueda, filtros y pedidos

#### **2.1.2 Objetivos Específicos**

- Definir los requerimientos funcionales del sistema, basado en las funcionalidades generales que esté tendrá.
- Diseñar los diagramas necesarios para entender el sistema, la base de datos, la interfaz del usuario y su funcionamiento.
- Establecer la Base de Datos a la par con la interfaz del usuario, teniendo en cuenta que la interfaz debe ser intuitiva y amigable con el usuario.
- Realizar pruebas de APIs con Postman para verificar las rutas que se realizan.
- Integrar la Base de Datos y la interfaz de usuario para su posterior despliegue.
- Documentar todo lo realizado durante el proyecto incluyendo manual de usuario y manual técnico.

### **2.2 Ambiente de Ingeniería de Software**

#### **2.2.1 Metodología de Desarrollo**

### 2.2.2 Las técnicas y anotaciones

Se emplea UML para el modelado de casos de uso, para la base de datos incluye un diagrama físico de la misma, el cual fue elaborado con el IDE “DataGrip”, el cual elabora el diagrama en base del código original de la base de datos.

### 2.2.3 Estándares de Documentación

Se utilizarán las normas APA para la documentación del proyecto, con el objetivo de garantizar un formato claro, estructurado y profesional en la presentación final. El documento deberá ser comprensible y actualizado para ayudar a tener toda la información sobre el proyecto realizado.

Para mantener la documentación actualizada conforme el avance o cambios del proyecto, se elaboraron distintas versiones del documento original.

### 2.2.4 Estándares de apoyo al desarrollo de software

**HTML, CSS y React.js:** Para la construcción de la interfaz de usuario responsiva e intuitiva con el objetivo de mejorar la experiencia del usuario.

**Node.js:** Para manejar el backend, permitiendo la conexión eficiente entre el frontend y el BackEnd. El FrontEnd se comunica con el BackEnd utilizando la función fetch de JavaScript para realizar peticiones HTTP (GET, POST, PUT, DELETE) a las rutas de la API que están conectadas con la base de datos.

**MySQL:** Se utilizará para la construcción de la base de datos, la cual se llama Librería BD. Su objetivo es modelar el flujo de compra de libros en línea. Gestionando usuarios, pedidos, detalles de libros ,ETC.

**Postman o Insomnia:** Se implementará para el testeo de las APIs, permitiendo interactuar con el sistema, evidenciando las fallas existentes para ser corregidas antes de la presentación final.

**GitHub:** Para controlar versiones y gestionar los avances del proyecto. Las ramas las gestionamos en la forma en la cual van trabajando los diferentes grupos de desarrollo, en el caso del ejemplo Backend implementando las API y Frontend actualizando las interfaces. Se tienen diferentes carpetas donde suben los archivos de los respectivos de cada grupo para así mantener un orden y reconocer el progreso de cada grupo

**Despliegue:** El despliegue de la aplicación se planea hacer en un servidor propio, el cual es de el compañero Juan Guerrero. Se busca montar todo el sistema en dicho host para poder acceder a él sin necesidad de mantener instalando node u otros archivos en varios dispositivos



---

### 3 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

---

Adaptación basada en *IEEE Software requirements Specifications Std 830-1998*.

#### 3.1 Alcances

El proyecto consiste en el desarrollo de una plataforma web que permitirá a los usuarios:

- Registrarse y gestionar su perfil.
- Iniciar sesión de forma segura.
- Visualizar un catálogo de libros con recomendaciones personalizadas.
- Realizar búsquedas de libros por diferentes criterios.
- Consultar los detalles de cada libro (descripción, precio, disponibilidad).
- Agregar libros a un carrito de compras y gestionar pedidos.
- Simular un sistema de pago.
- Visualizar el historial de pedidos y compras realizadas.

Además, incluirá un panel de administración que permitirá a los administradores:

- Gestionar la información de los libros (agregar, editar, eliminar).
- Gestionar los pedidos y compras realizadas por los clientes.
- Administrar los perfiles de usuarios registrados.

El sistema contará con un diseño responsivo para adaptarse a diferentes dispositivos, ofreciendo una experiencia de usuario fluida y accesible.

El proyecto abarca:

El diseño y desarrollo del frontend (HTML, CSS, React.js).

La construcción del backend (Node.js, PostgreSQL).

La creación de APIs RESTful para la comunicación entre frontend y backend.

La implementación de pruebas básicas de las APIs mediante Postman.

El control de versiones a través de GitHub.

### 3.2 Objetivo del software

Se describen los objetivos que debe cumplir el software en forma general y específica. Debería señalarse en el objetivo global y correspondientes específicos los siguientes elementos o aspectos:

**INFORMACION** que considera /almacena / gestiona /maneja /etc-el **PROCESO** que apoya/realiza- y el **RESULTADO** que se logra.

Ejemplo: El sistema **manejará** información sobre el **proceso productivo** que permita una **planificación integral del mismo y logra un uso optimo de los recursos utilizados en el proceso.**

### 3.3 Descripción Global del Producto

#### 3.3.1 Interfaz de usuario

Se indican las características lógicas de cada interfaz entre el software y los usuarios. No se refiere al diseño de la interfaz sino a los requerimientos que existen en la empresa respecto a la interfaces de sw.

Por ejemplo puede incluir características de configuración como formato de pantalla, o layout de informes, uso de colores, iconografía, usos de teclas programables, etc.

Se indican también todos los aspectos de optimización, que indique las forma como el software debe y no debe aparecer al usuario.

#### 3.3.2 Interfaz De Hardware

Dado que el sistema es una aplicación web, no se requiere interacción directa con hardware especializado. El acceso al sistema se realizará mediante navegadores web estándar en dispositivos de escritorio y móviles, utilizando periféricos básicos como teclado, mouse o pantalla táctil.

#### 3.3.3 Interfaces de comunicación

El sistema establecerá la comunicación entre el frontend y el backend a través de peticiones HTTP utilizando fetch.

**Tabla 1: Ruta Login**

*Tabla de Ruta para Login con Descripción*

POST	<a href="https://store.racoonlibrary.studio/login">https://store.racoonlibrary.studio/login</a>	Recibe el correo y la contraseña desde un formulario web. Valida los datos contra la base de datos y permite el acceso si las credenciales son correctas.
------	---	---

*Nota: Esta tabla menciona la descripción y ruta del Login con las que Backend trabaja*

**Tabla 2: Ruta Register**

*Tabla de Ruta para Register con Descripción*

POST	<a href="https://store.racoonlibrary.studio/registrar">https://store.racoonlibrary.studio/registrar</a>	Recibe datos como nombre, correo e información del usuario desde un formulario web. Registra y guarda estos datos en la base de datos.
------	---	--

*Nota: Esta tabla menciona la descripción y ruta del Registro con las que Backend trabaja*

### 3.4 Requerimientos Específicos

#### 3.4.1 Requerimientos Funcionales del sistema

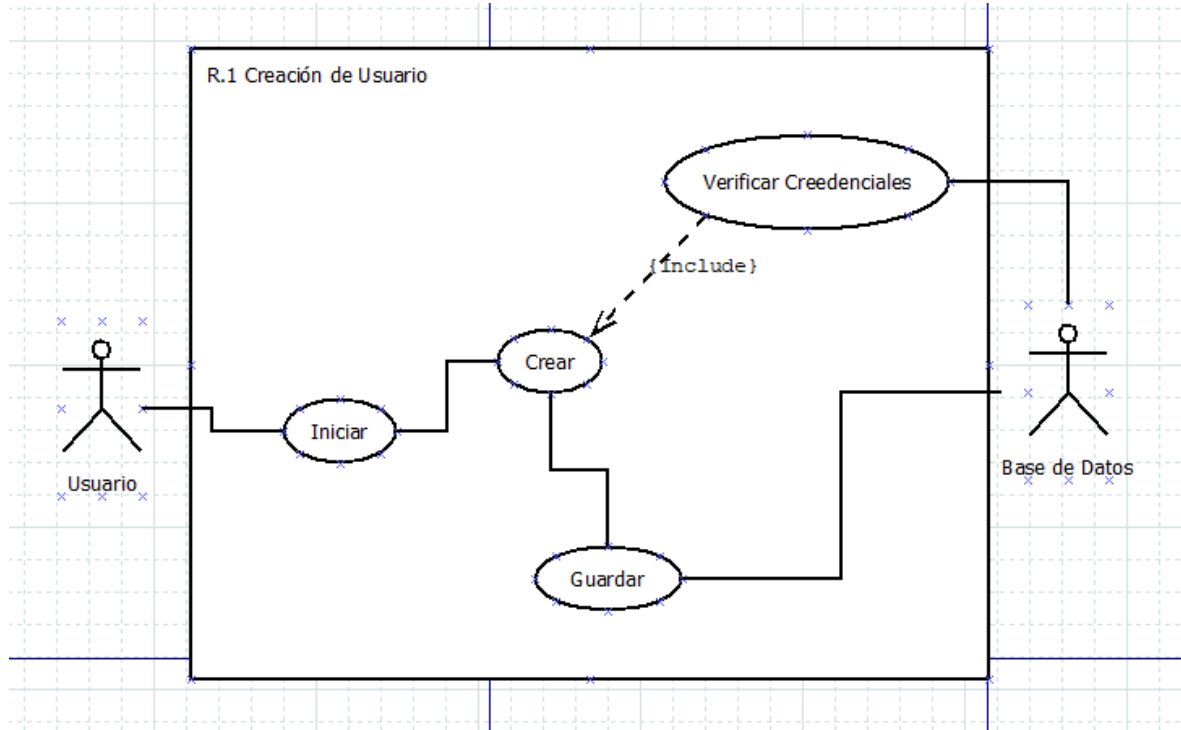
**Tabla 3: Requerimiento Funcional Crear Cuenta**  
*Requerimiento funcional n°1 [creación de usuario]*

Especificación de requerimientos funcionales	
<b>Requerimiento Funcional N°: 01</b>	<b>Nombre</b>
RQF 01	Crear Cuenta
<b>Tipo:</b>	<b>Prioridad:</b>
Necesario	Alta
<b>Descripción:</b>	
El sistema debe permitir al usuario crear una nueva cuenta para el uso dentro de la página debe almacenar el usuario en la base de datos	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"><li>• Los campos no pueden enviarse vacíos</li></ul>	

- El correo debe tener un dominio
- La contraseña debe aceptar todos los caracteres
- El sistema debe redirigir al usuario a la página de login después de crear la cuenta
- El usuario debe aparecer en la base de datos de la página
- Los campos no pueden estar vacíos

***Nota: Esta tabla menciona la descripción y criterios de aceptación para manejar el registro de nuevos usuarios en el sistema***

**Figura 1**  
**Caso de Uso n°1 [creación de usuario]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario con la base de datos al momento de crear y guardar las credenciales creadas en el software*

**Tabla 4: Requerimiento Funcional Iniciar Sesión**

***Requerimiento funcional n°2 [Inicio sesión]***

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:02	Nombre
RQF 02	Iniciar Sesión
Tipo:	Prioridad:
Necesario	Alta
Descripción:	
El sistema permitirá ingresar al aplicativo digitando las credenciales de correo y contraseña	

con las que creó la cuenta el usuario.

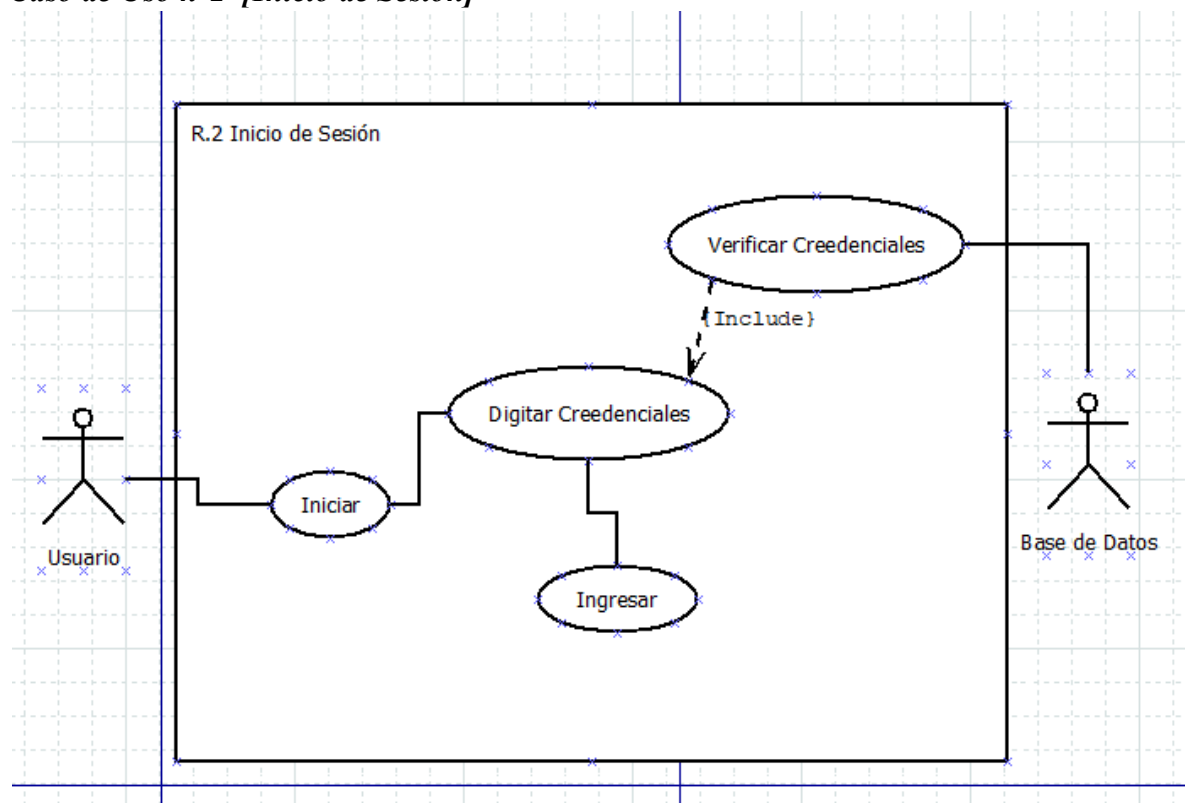
### Criterios de aceptación

- Las credenciales deben ser validadas en la base de datos
- El sistema muestra un error si las credenciales no coinciden en el sistema
- Cuando las credenciales coincida el sistema debe redirigir al usuario al menú principal
- Las cuenta debe bloquearse luego de 4 intentos fallidos de login

*Nota: La tabla referencia los criterios que se debe tener en el proceso de inicio de sesión dentro del sistema*

**Figura 2**

**Caso de Uso n°2 [Inicio de Sesión]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario con la base de datos al momento de iniciar sesión en la página web*

**Tabla 3: Requerimiento Funcional Editar Perfil del Usuario**

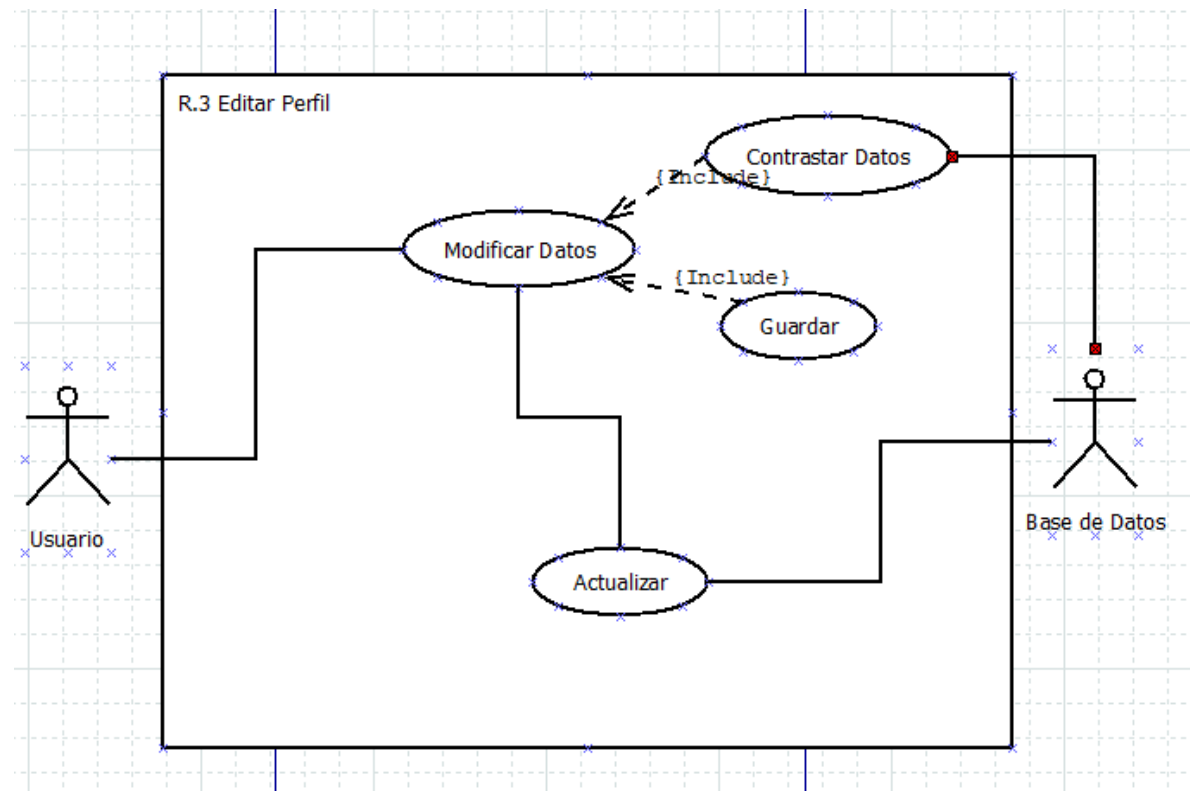
***Requerimiento funcional n° 3 [Editar perfil]***

<b>Especificación de requerimientos funcionales</b>	
<b>Requerimiento Funcional N°:03</b>	<b>Nombre</b>
RQF 03	Editar perfil del usuario
<b>Tipo:</b>	<b>Prioridad:</b>
Necesario	Media
<b>Descripción:</b>	
El sistema permite al usuario ingresar a las configuraciones de su perfil, permitiendo modificar diferentes variables correspondientes al usuario (nombre, correo, contraseña) y variables dependientes del proceso de compra.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Los nuevos datos deben cumplir el mismo formato que los ingresados previamente</li> <li>• Los datos deben ser actualizados en la base de datos del sistema</li> <li>• Mostrar un mensaje de confirmación de actualización de datos</li> <li>• El nuevo correo no debe estar vinculado con ninguna otra cuenta del sistema</li> </ul>	

***Nota: Esta tabla Especifica los criterios para actualizar datos dentro de el sistema***

**Figura 3**

*Caso de Uso n°3 [Editar Perfil]*



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario con la base de datos al momento de editar el perfil creado por el usuario, se deben contrastar datos para que no se repitan y guardar para actualizar en la Base de Datos*

**Tabla 4: Requerimiento Funcional Barra De Búsqueda**

*Requerimiento funcional n°4 [Barra de búsqueda]*

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:04	Nombre
RQF 04	Barra de búsqueda
Tipo:	Prioridad:
Necesario	Alta

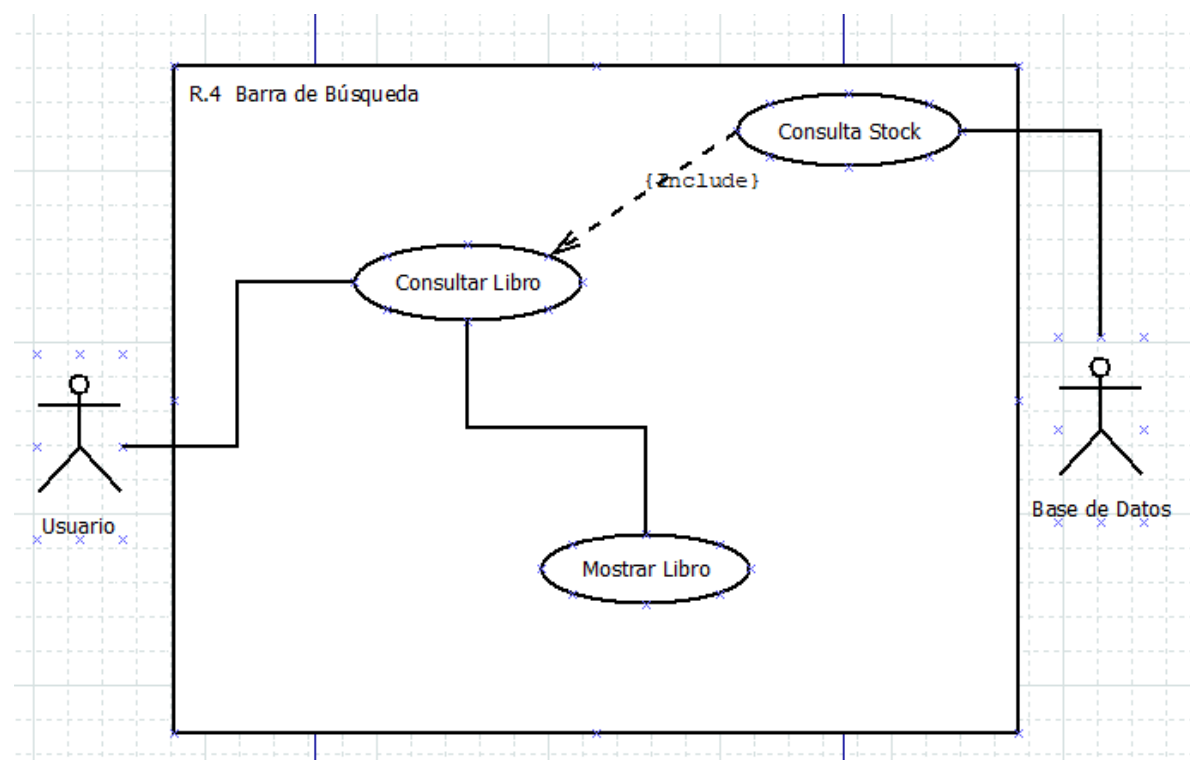


<b>Descripción:</b>
El sistema debe permitir consultas de libros basado en el nombre del mismo, mostrando la descripción.
<b>Criterios de aceptación</b>
<ul style="list-style-type: none"> <li>• El campo de búsqueda no puede enviarse vacío</li> <li>• Realizar la consulta en la base de datos</li> <li>• Mostrar detalles del libro, o libros similares</li> <li>• En dado de no encontrarlo mostrar un mensaje</li> </ul>

*Nota: La tabla expresa los criterios de búsqueda en el sistema*

**Figura 4**

**Caso de Uso n°4 [Barra de Búsqueda]**



Entre Líneas

*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario con la base de datos al momento de buscar un libro por medio de la barra de búsqueda, donde consulta stock y muestra el libro*

**Tabla 5: Requerimiento Funcional Lista de Libros (Administrador)**

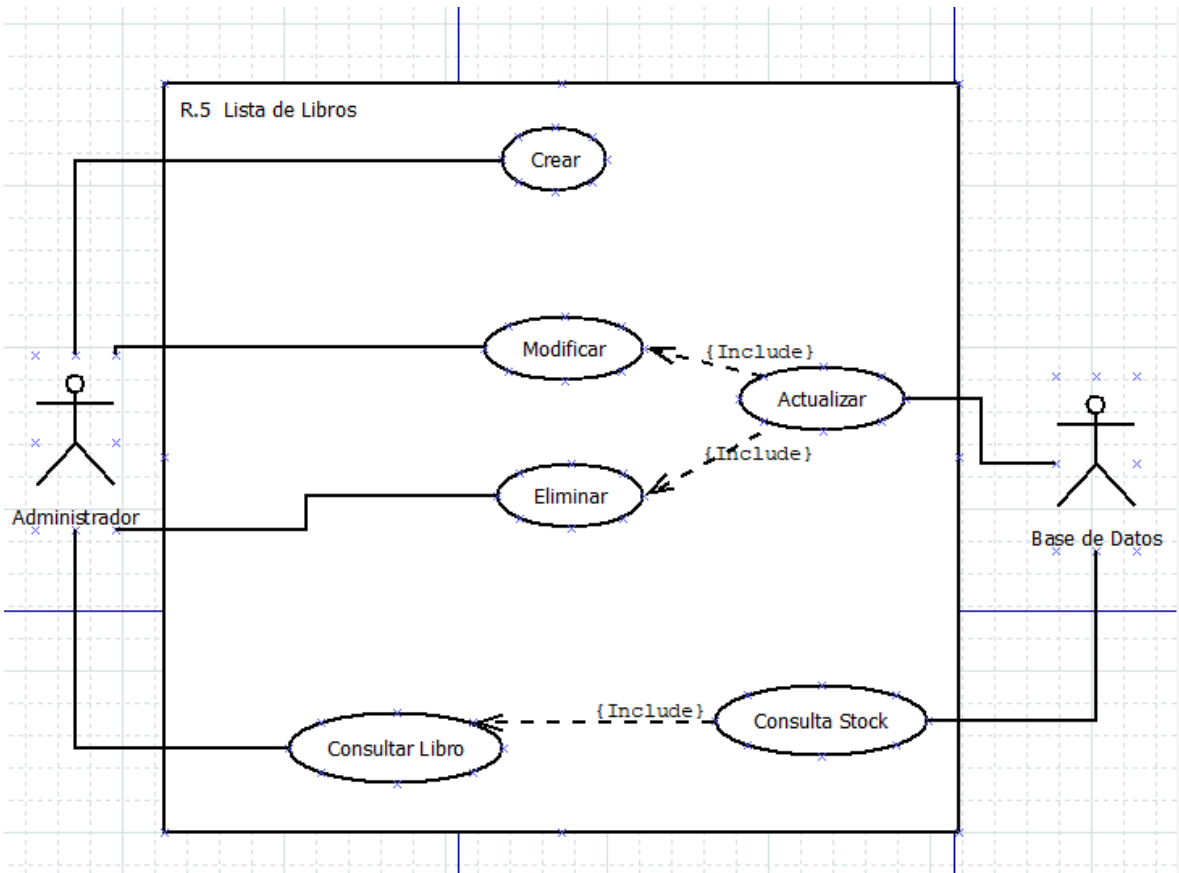
***Requerimiento funcional n° 5[Lista de libros (Admin)]***

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:05	Nombre
RQF 05	Lista de libros (Administrador)
Tipo:	Prioridad:
Necesario	Alta
Descripción:	
El administrador podrá revisar las listas de libros en inventario existente, en ese caso podrá consultarlos, modificarlos o eliminarlos en caso de ser necesarios, todo visualizado en un lista de libros.	
Criterios de aceptación	
<ul style="list-style-type: none"><li>- El sistema deberá confirmar la existencia de los libros en la base de datos.</li><li>- Al actualizar información se deberá actualizar en la base de datos.</li></ul>	

*Nota: La tabla expresa las acciones que puede tener un administrador en el módulo de libros*

Figura 5

Caso de Uso n°5 [Lista de Libros]



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del administrador con la base de datos al momento de crear, modificar, eliminar o consultar un libro mediante lista de libros y comprobar en la base de datos*

Tabla 6: Requerimiento Funcional Filtros

Requerimiento funcional n° 6 [Filtros]

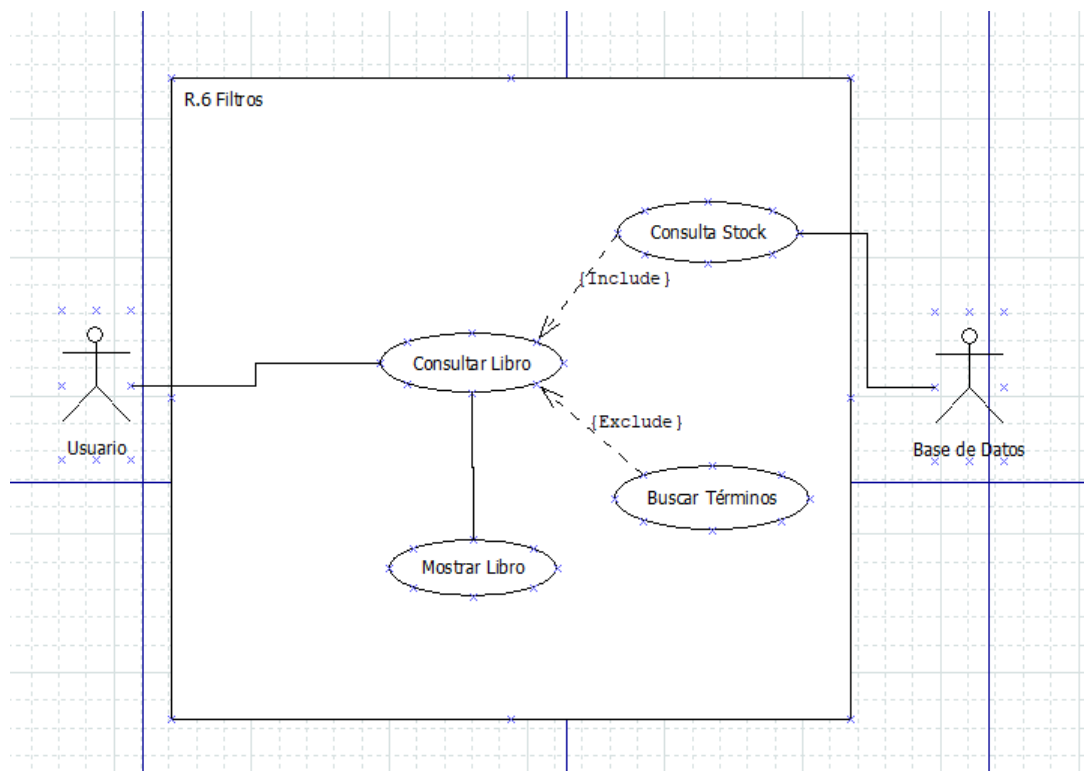
Especificación de requerimientos funcionales	
Requerimiento Funcional N°:06	Nombre
RQF 06	Filtros
Tipo:	Prioridad:

Necesario	Media
<b>Descripción:</b>	
El sistema permitirá realizar consultas en la barra de búsqueda por medio de diferentes filtros generados, en ese caso se buscarán por medio de Autor, Año, Género entre otros...	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• El sistema deberá generar grupos por medio de los filtros escogidos.</li> <li>• El sistema contrastará en la base de datos las consultas por medio de filtros.</li> </ul>	

*Nota: La tabla explica los filtros que se usarán para la barra de búsqueda en los procesos de búsqueda de libros*

**Figura 6**

*Caso de Uso n°6 [Filtros]*



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario y la base de datos al momento de consultar un libro identificando que es posible buscar el libro por medio de Filtros y mostrar el libro*

**Tabla 7: Requerimiento Funcional Crear categorías/filtros (Administrador)**

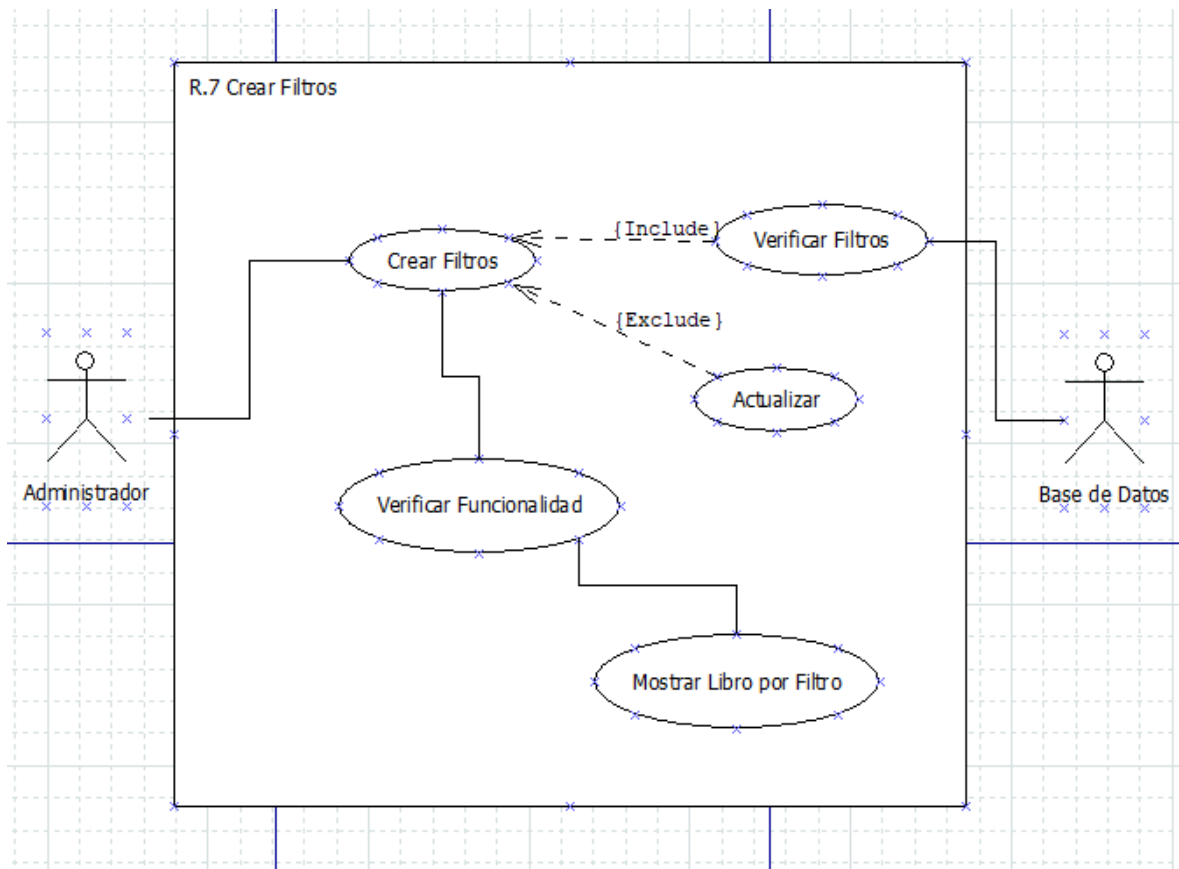
***Requerimiento funcional n° 7 [Crear Filtros]***

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:07	Nombre
RQF 07	Crear categorías/filtros (Administrador)
Tipo:	Prioridad:
Necesario	Media
Descripción:	
El administrador podrá generar filtros en la búsqueda de libros para el usuario, permitiendo tener mejor acceso a las búsquedas y consultas para el usuario.	
Criterios de aceptación	
<ul style="list-style-type: none"> <li>- El sistema deberá revisar si no está creado el filtro en la base de datos antes de agregarlo.</li> <li>- El sistema avisará si es posible o no crear el filtro de búsqueda para el usuario, esto con mensajes por pantalla.</li> </ul>	

*Nota: En esta tabla se expresa la posibilidad de que el Administrador pueda crear filtros en beneficio del usuario y la búsqueda de libros*

**Figura 7**

**Caso de Uso n°7 [Crear Filtros]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del administrador y la base de datos el crear filtros para la barra de búsqueda, como la base de datos verifica los filtros y se actualiza, se realizan las pruebas para que muestre los libros*

**Tabla 8: Requerimiento Funcional Recomendaciones**

**Requerimiento funcional n° 8 [Recomendaciones]**

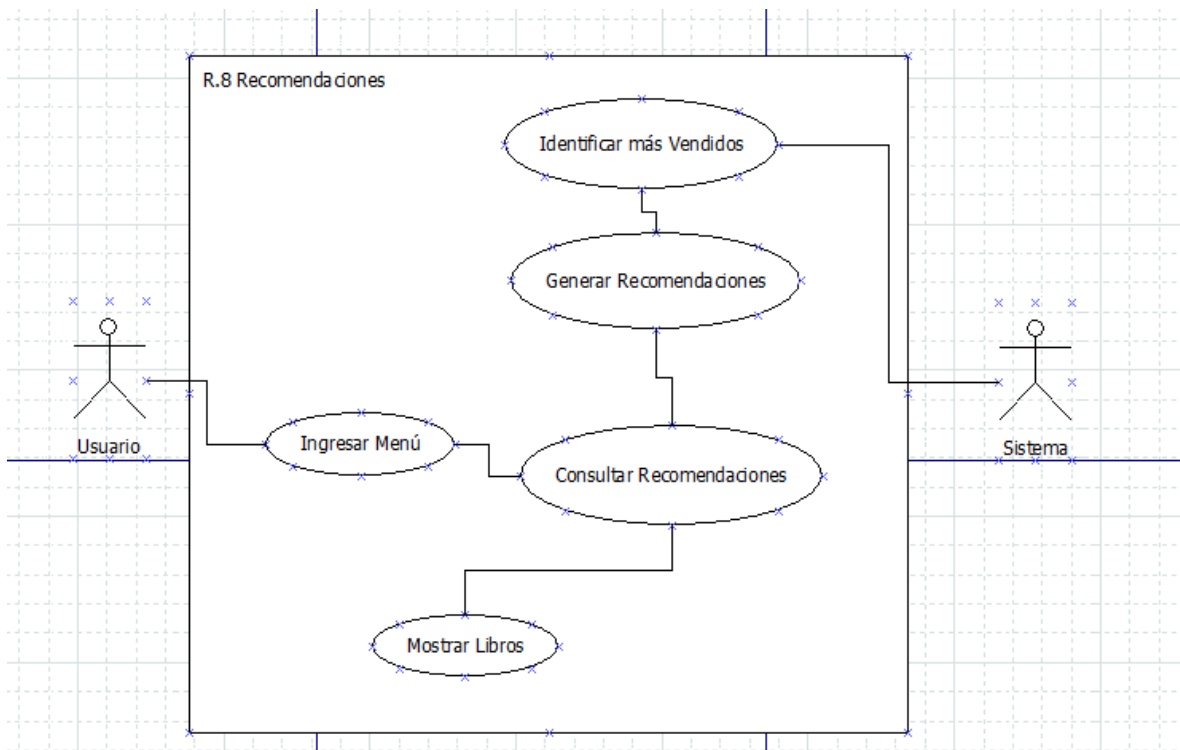
Especificación de requerimientos funcionales	
Requerimiento Funcional N°:08	Nombre
RQF 08	Recomendaciones
Tipo:	Prioridad:

Necesario	Medio
<b>Descripción:</b>	
El sistema identificará los libros más vendidos, los cuales mostrará de primeras a los usuarios en forma de recomendaciones de compra.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>- El sistema deberá revisar los libros más comprados para poder dejarlos en recomendaciones para todos los usuarios.</li> </ul>	

*Nota: En esta tabla se explica cómo el sistema identificará los libros más vendidos para dar recomendaciones a los usuarios para comprar*

**Figura 8**

**Caso de Uso n°8 [Recomendaciones]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario y el sistema, de parte del sistema generar las recomendaciones con libros más vendidos y del usuario consultar las recomendaciones dadas*

**Tabla 9: Requerimiento Funcional Información del Libros**

***Requerimiento funcional n° 9 [Información del libro ]***

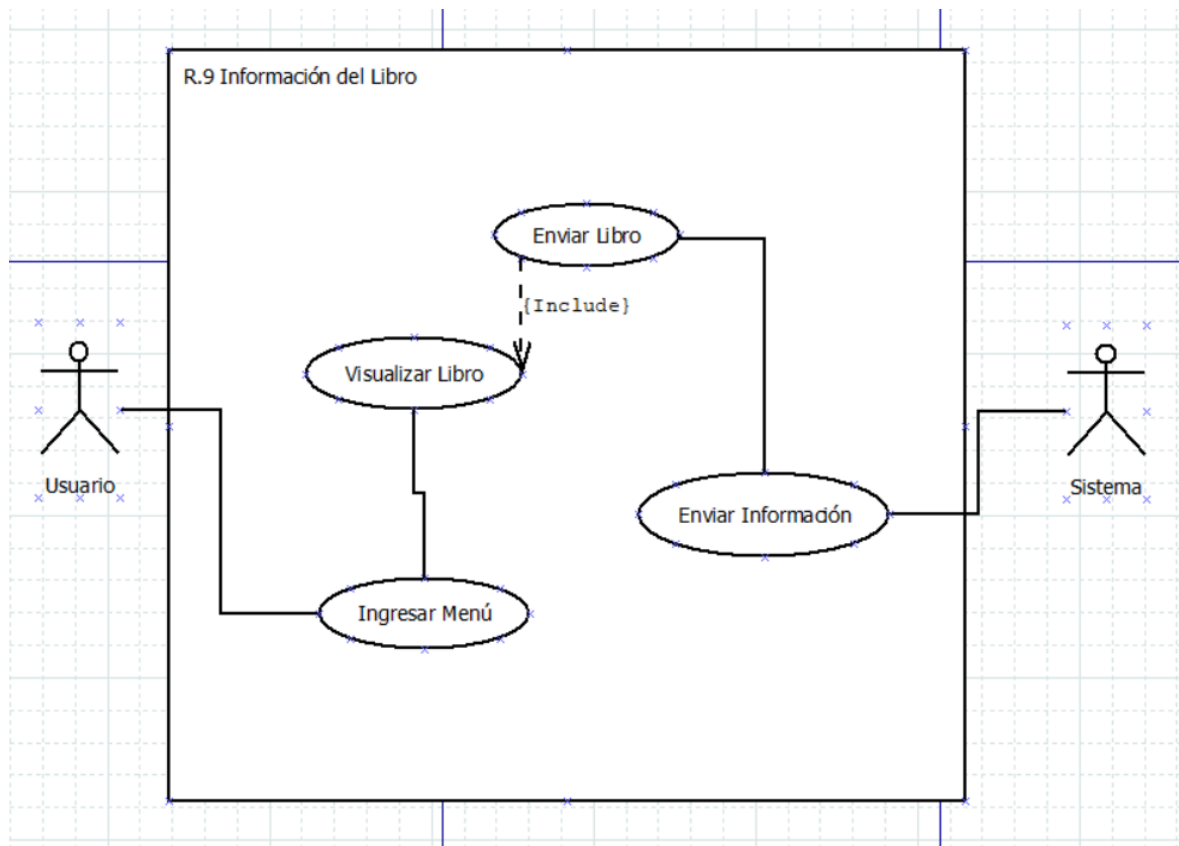
<b>Especificación de requerimientos funcionales</b>	
<b>Requerimiento Funcional N°:09</b>	<b>Nombre</b>
RQF 09	Información del libro
<b>Tipo:</b>	<b>Prioridad:</b>
Necesario	Alta
<b>Descripción:</b>	
El sistema tendrá los libros guardados, en ese caso, aloja la información de cada uno para mostrar su sinopsis y diferentes detalles del libro para poder mostrar al usuario la descripción del libro.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>- El sistema no permitirá dejar espacio en blanco para la descripción del libro.</li> <li>- El sistema deberá tener límite de palabras para la creación de la descripción de los libros.</li> </ul>	

***Nota: La tabla identifica la información del libro guardadas para mostrarlas al usuario***



**Figura 9**

**Caso de Uso n°9 [Información del Libro]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario y el sistema, de parte del sistema enviar la información del libro y enviar el libro que se desea visualizar y por parte del Usuario la consulta del libro*

**Tabla 10: Requerimiento Funcional Gestión de libros (Usuario)**

**Requerimiento funcional n°10 [Gestión de libros (Usuario) ]**

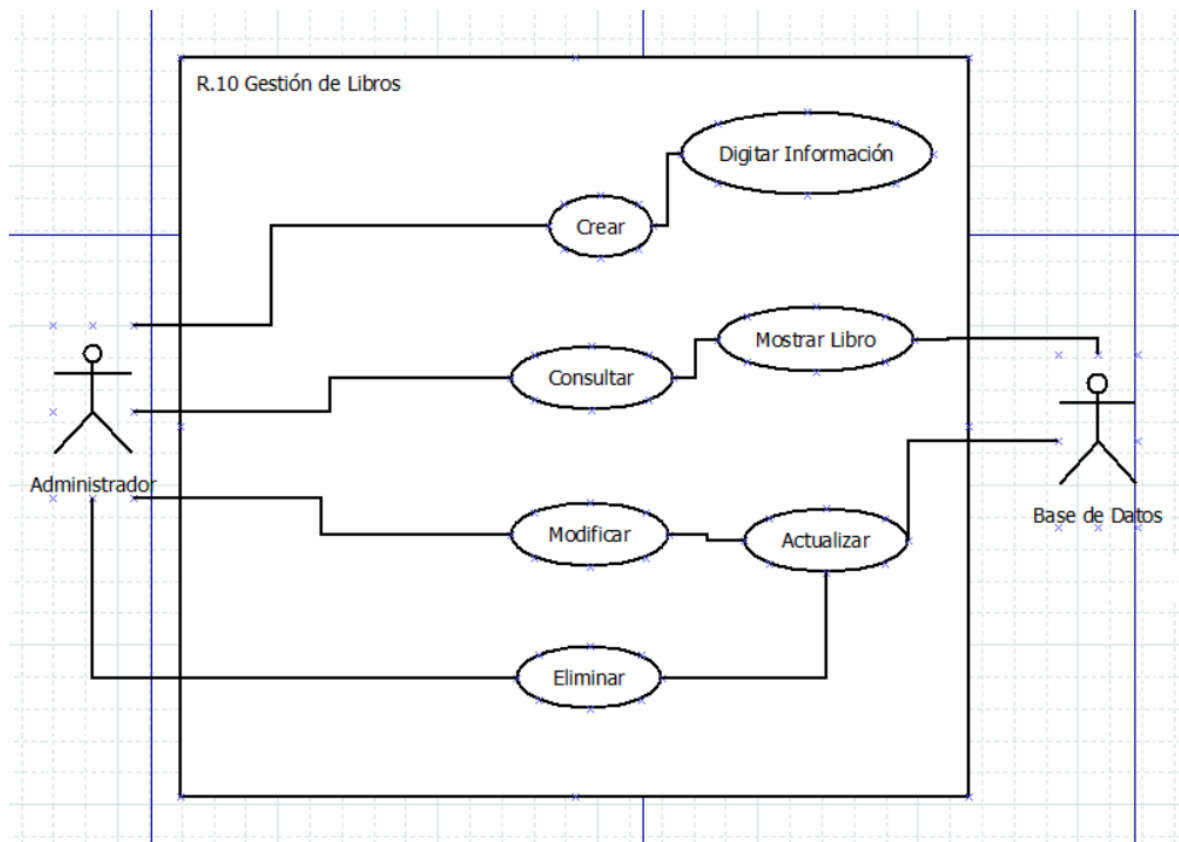
Especificación de requerimientos funcionales	
Requerimiento Funcional N°:10	Nombre
RQF010	Gestión de libros (Usuario)
Tipo:	Prioridad:

Necesario	Alta
<b>Descripción:</b>	
El administrador podrá gestionar el catálogo de libros visibles en la biblioteca, permitiendo modificar el catálogo que podrá ser mostrado para el usuario y disponible para su compra	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>- El sistema debe permitir agregar, editar y eliminar libros en el catálogo</li> <li>- La modificación en el catálogo se debe actualizar en la base de datos</li> <li>- El catálogo actualizado será mostrado a los usuarios al ingresar a la página</li> </ul>	

*Nota: La tabla nos menciona que el usuario deberá de tener la capacidad de poder visualizar únicamente los libros deseados por el administrador de la página, limitando la capacidad de compra pero no siendo el equivalente a borrar o editar los libros de la base de datos*

**Figura 10**

*Caso de Uso n°10 [Gestión de libros]*



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del administrador y la Base de Datos para la Gestión de Libros de un catálogo, en este caso al momento de realizar las operaciones se contrastará con las Base de Datos o se guardará*

**Tabla 11: Requerimiento Funcional Información del libro (Usuario)**

*Requerimiento funcional n° 11 [Información del libro (Usuario) ]*

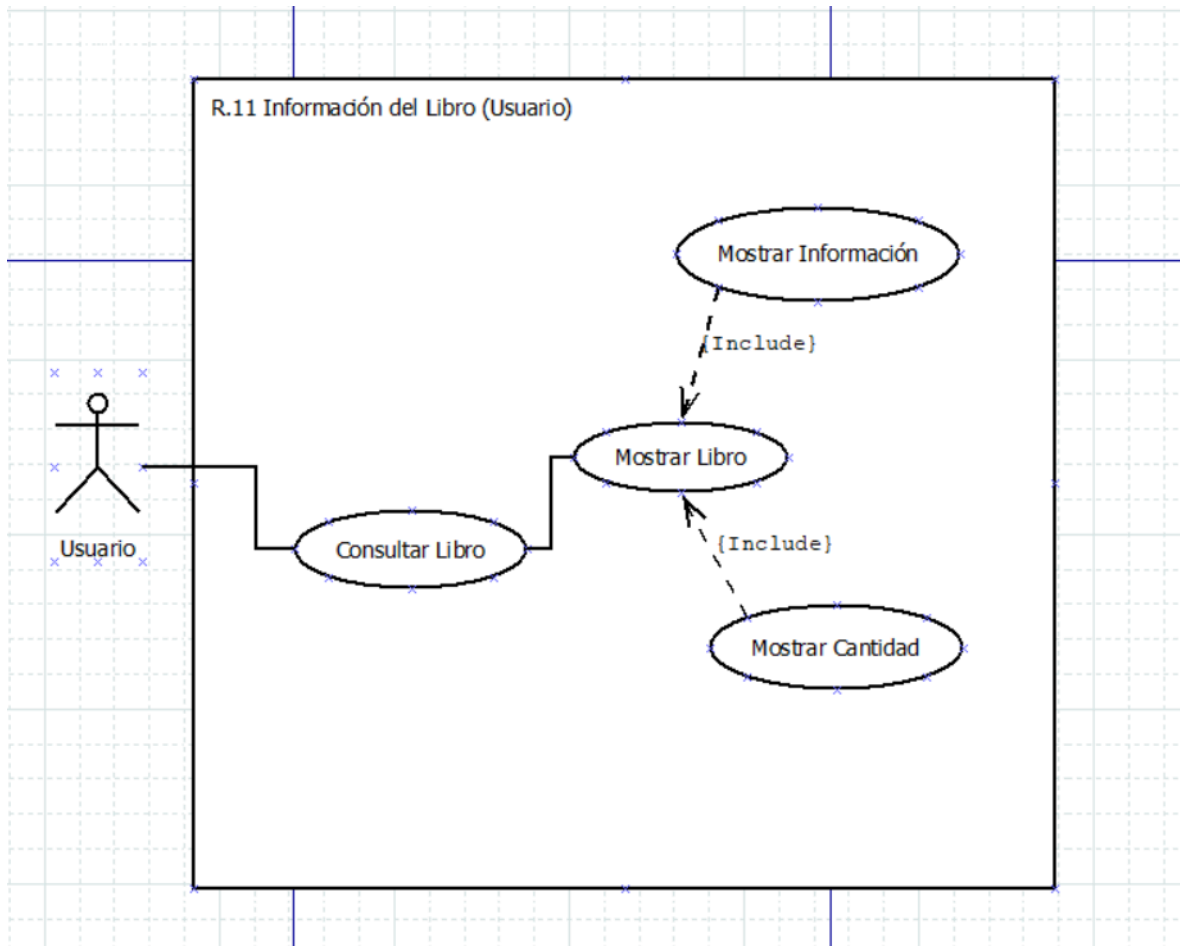
Especificación de requerimientos funcionales	
Requerimiento Funcional N°:11	Nombre
RQF011	Información del libro (Usuario)

<b>Tipo:</b>	<b>Prioridad:</b>
Necesario	Alta
<b>Descripción:</b>	
Muestra los detalles de un libro, tales como : <ul style="list-style-type: none"><li>● Título</li><li>● Autor</li><li>● Género</li><li>● Número de ejemplares</li></ul>	
<b>Criterios de aceptación</b>	
- La página deben mostrar los detalles de cualquier libro registrado en la biblioteca	

*Nota: La tabla nos menciona que cuando el usuario entre a buscar un libro a este se le ha de brindar toda la información relacionada a este incluyendo: “Título, Autor, Género, Número de ejemplares, etc...”*

**Figura 11**

***Caso de Uso n°11 [Información del Libro (Usuario)]***



***Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario con el sistema al momento de consultar un libro y que esté le muestre la información y la cantidad de Stock del libro***

**Tabla 12: Requerimiento Funcional Editar detalles del libro (Administrador)**

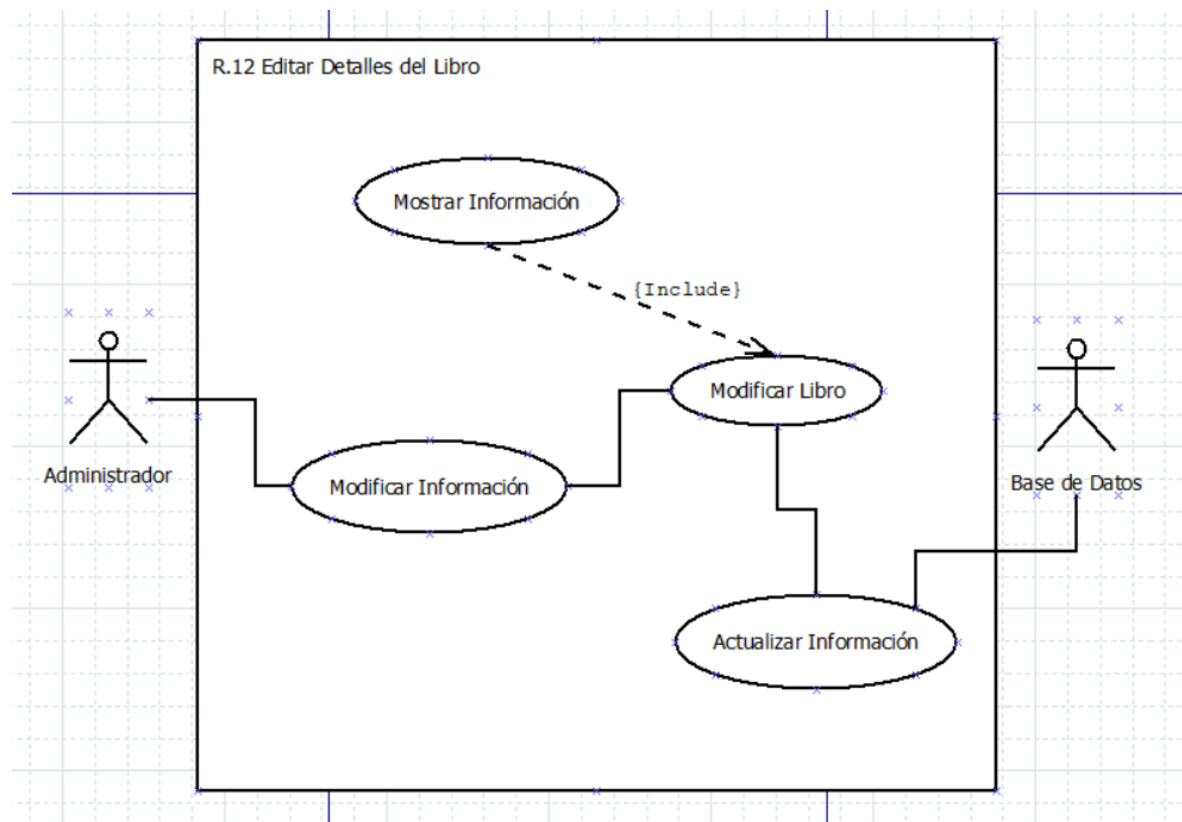
***Requerimiento funcional n° 12 [Editar detalles del libro (Administrador) ]***

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:12	Nombre
RQF012	Editar detalles del libro (Administrador)
Tipo:	Prioridad:
Necesario	Alta
Descripción:	
Permite al administrador modificar la información de un libro en el catálogo, más no eliminarlo	
Criterios de aceptación	
<ul style="list-style-type: none"> <li>- Solo los administradores pueden editar los detalles de lo libros</li> <li>- Los detalles del libro deben ser actualizados en la base de datos</li> </ul>	

***Nota: La tabla nos menciona una de las capacidades del administrador, siendo esta la capacidad de poder editar la información de los libros dentro de la base de datos, esto incluye: “la descripción del libro, su título, autor, categorías, género, imagen de portada o a mostrar de libro”***

**Figura 12**

***Caso de Uso n°12 [Editar Detalles del Libro]***



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del administrador con el sistema y la Base de Datos, en este caso el administrador modifica la información de un libro por lo que se actualiza en la base de datos*

**Tabla 13: Requerimiento Funcional Agregar libros (Administrador)**

***Requerimiento funcional n° 13 [Agregar libros (Administrador) ]***

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:13	Nombre
RQF013	Agregar libros (Administrador)
Tipo:	Prioridad:
Necesario	Alta
Descripción:	
El administrador ha de poder agregar libros a la base de datos, esto incluye, la descripción del libro, su título, autor, categorías, género, imagen de portada o a mostrar de libro	

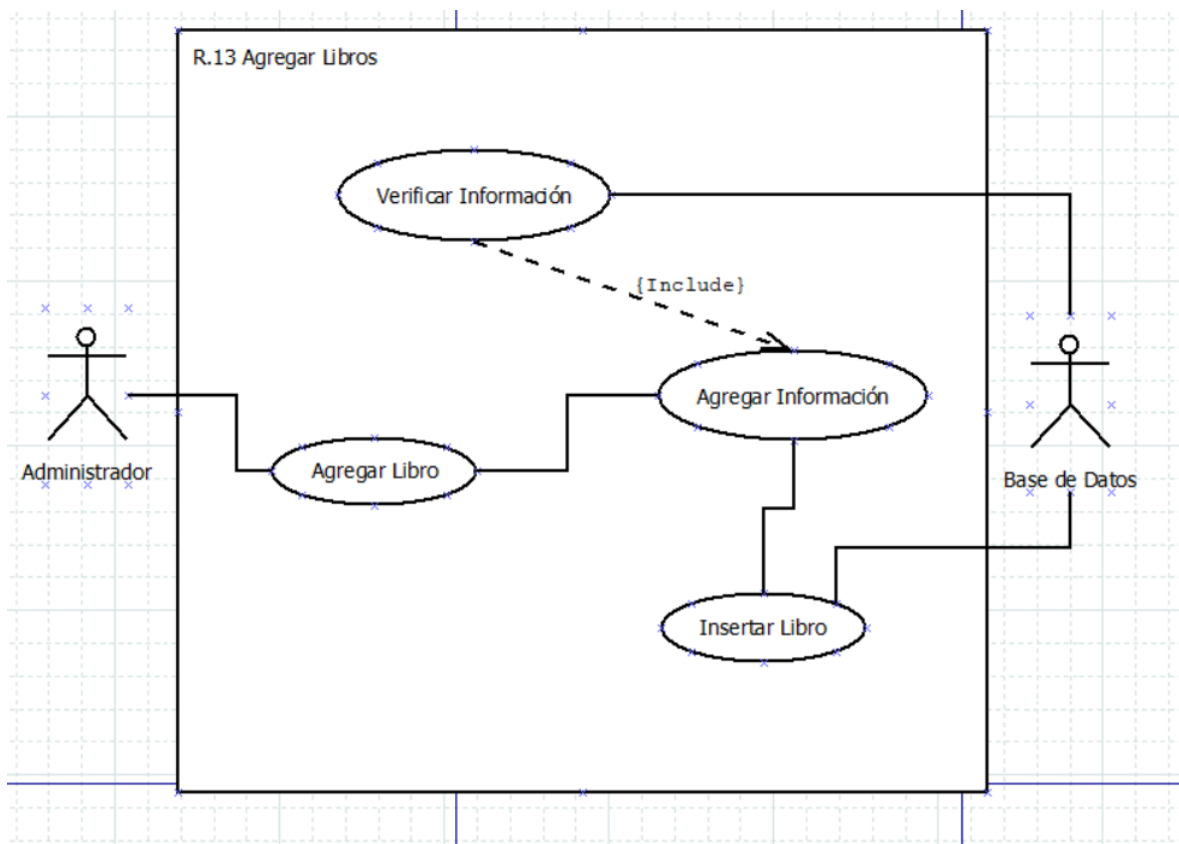
### Criterios de aceptación

- El sistema ha de actualizar y subir la base de datos la información de libro a crear por el administrador

*Nota: La tabla nos menciona una de las capacidades del administrador de la página, siendo esta, la capacidad de poder agregar libros a la pagina, siendo necesariamente agregar “la descripción del libro, su título, autor, categorías, género, imagen de portada o a mostrar de libro”*

**Figura 13**

**Caso de Uso n°13 [Agregar Libros]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del administrador con el sistema y la Base de Datos, en este caso el administrador agrega un libro añadiendo información e insertando el libro en la base de datos, todo es verificado por la base de datos para que sea correcto*



**Tabla 14: Requerimiento Funcional Eliminar libros (Administrador)**

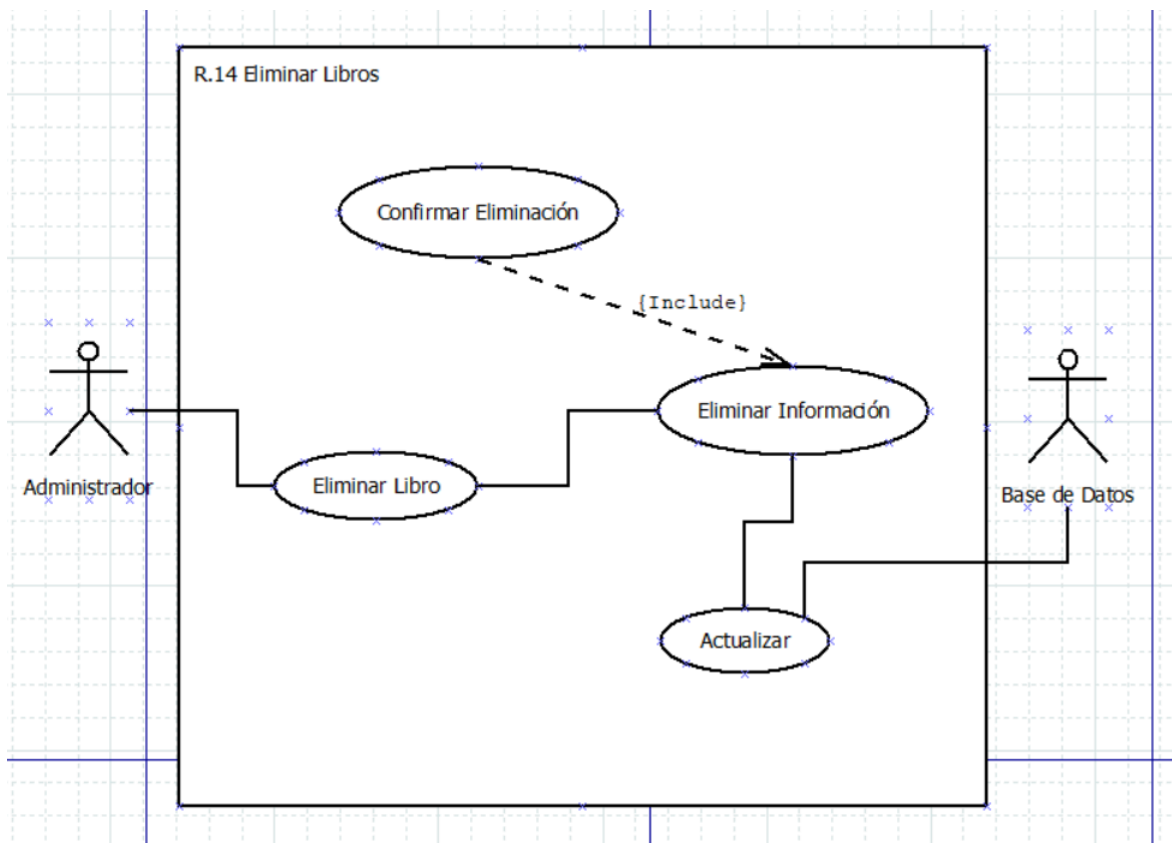
***Requerimiento funcional n° 14 [Eliminar libros (Administrador) ]***

<b>Especificación de requerimientos funcionales</b>	
<b>Requerimiento Funcional N°:14</b>	<b>Nombre</b>
RQF014	Eliminar libros (Administrador)
<b>Tipo:</b>	<b>Prioridad:</b>
Necesario	Alta
<b>Descripción:</b>	
El administrador ha de poder acceder a la base de datos de la librería y eliminar algún libro de ésta, en caso de que ya no desee venderlo o necesitarlo en la librería	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>- El sistema ha de borrar totalmente la información del libro deseado borrar por el administrador</li> </ul>	

***Nota: La tabla nos menciona una de las capacidades del administrador de la página, siendo que esta, la capacidad de poder eliminar libros totalmente de la base de datos***

**Figura 14**

***Caso de Uso n°14 [Eliminar Libros]***



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del administrador con el sistema y la Base de Datos, en este caso el administrador elimina un libro o la información de un libro, para ello se pide que confirme y en ese caso se actualiza en la Base de Datos*

**Tabla 15: Requerimiento Funcional Cantidad a comprar**

***Requerimiento funcional n° 15 [Cantidad a comprar]***

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:15	Nombre
RQF015	Cantidad a comprar
Tipo:	Prioridad:
Necesario	Alta
Descripción:	

El usuario al momento de comprar un libro, ha de poder seleccionar la cantidad a comprar antes de agregarla al carrito de compras

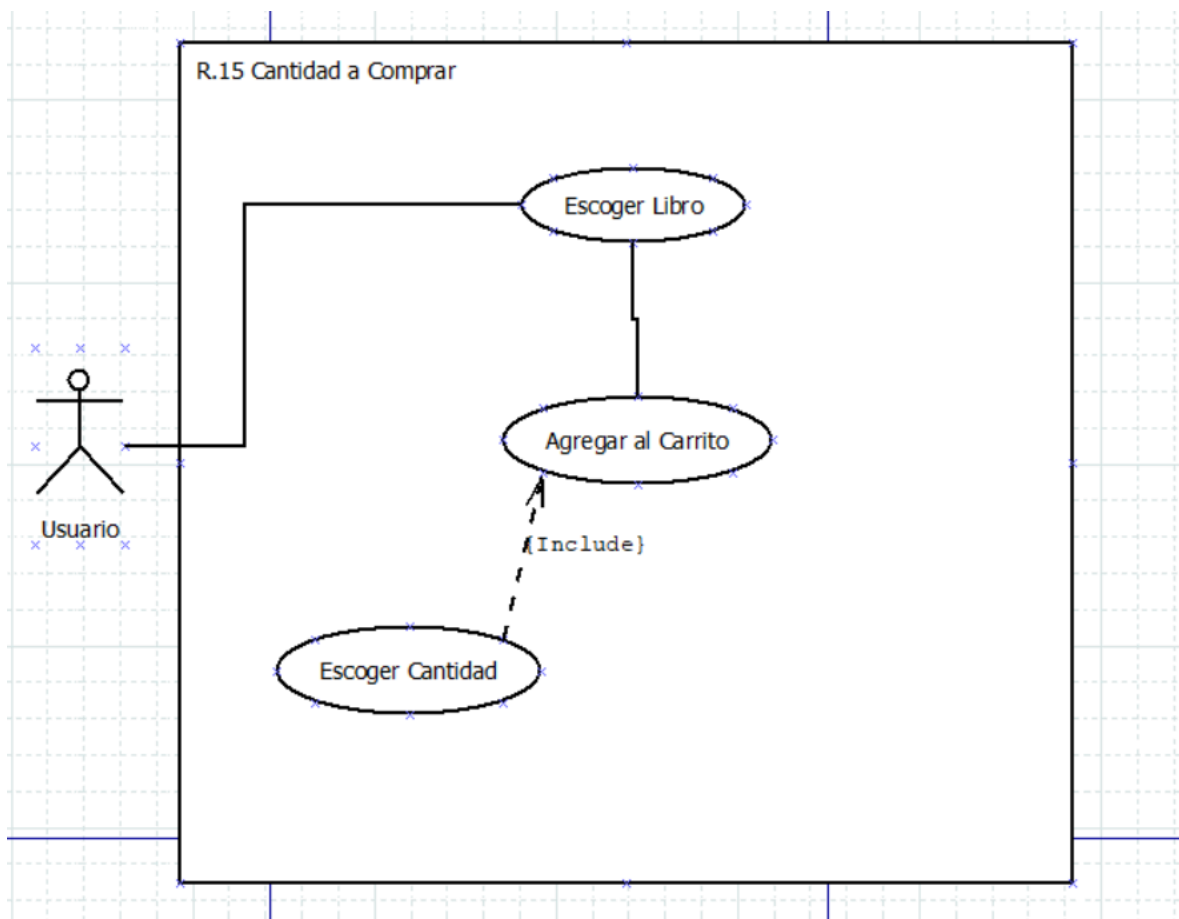
#### Criterios de aceptación

- El sistema ha de actualizar el carrito con la cantidad indicada del usuario

*Nota: La tabla nos menciona que los usuarios han de poder seleccionar la cantidad de libros a comprar antes de agregarlos al carrito*

**Figura 15**

**Caso de Uso n°15 [Cantidad a Comprar]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario al momento de escoger un libro y agregarlo al carrito para comprar, en ese caso debe escoger la cantidad antes de comprarlo*

**Tabla 16: Requerimiento Funcional Carrito de compras**

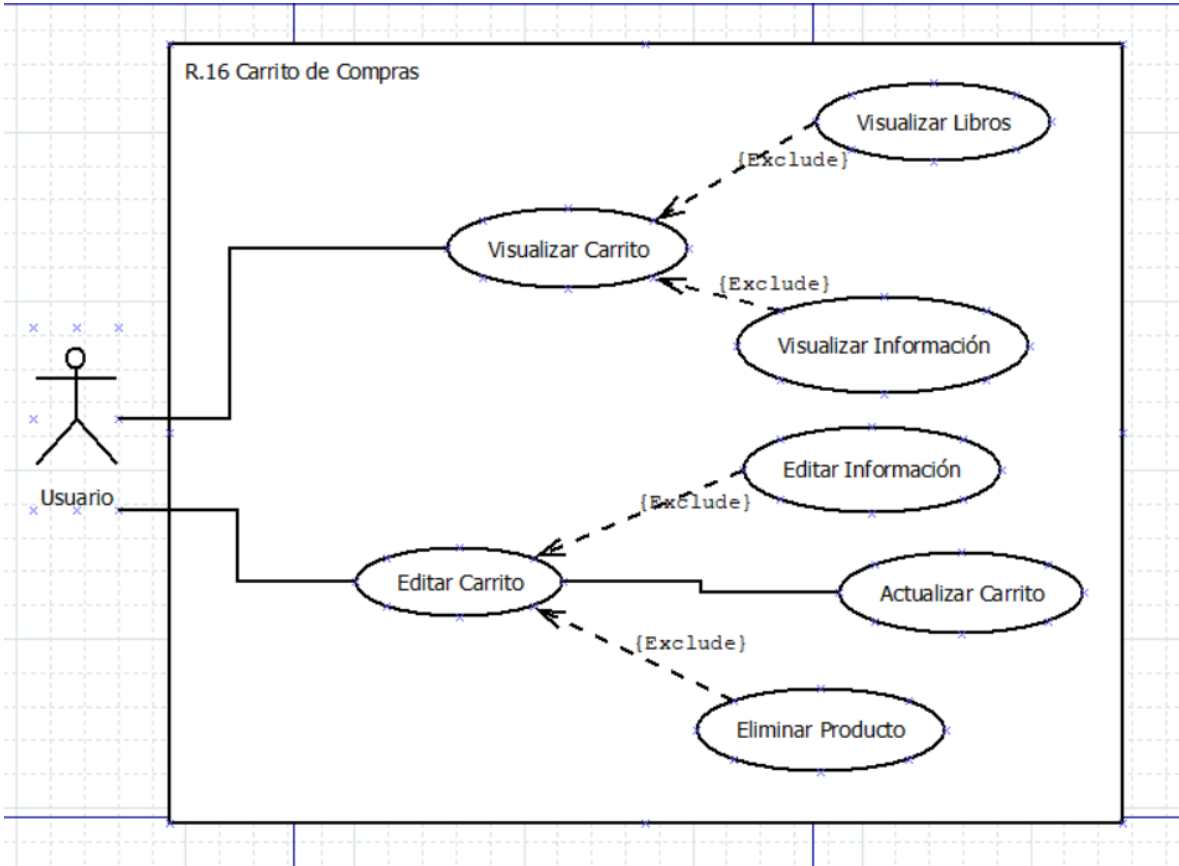
*Requerimiento funcional n° 16 [Carrito de compras ]*

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:16	Nombre
RQF016	Carrito de compras
Tipo:	Prioridad:
Necesario	Alta
Descripción:	
El usuario ha de poder visualizar el contenido dentro de su carrito de compras, al igual que la posibilidad de editar el contenido de esta misma	
Criterios de aceptación	
<ul style="list-style-type: none"><li>- El sistema ha de actualizar el carrito de compras del usuario</li><li>- El sistema ha de guardar la información del carrito de compras para luego ser procesado en la compra</li></ul>	

*Nota: La tabla nos menciona la necesidad de un carrito de compras dentro de la página para almacenar de forma temporal los productos deseados por el usuario, este ha de poder ser editado*

**Figura 16**

*Caso de Uso n°16 [Carrito de Compras]*



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario al momento de entrar al carrito de compras, poder visualizar su contenido y productos, además de poder editar el carrito de compras con poder eliminar sus productos o editar información para la compra, una vez editado se actualiza el carrito de compras*

Tabla 17: Requerimiento Funcional Entrega

***Requerimiento funcional n° 17 [Entrega ]***

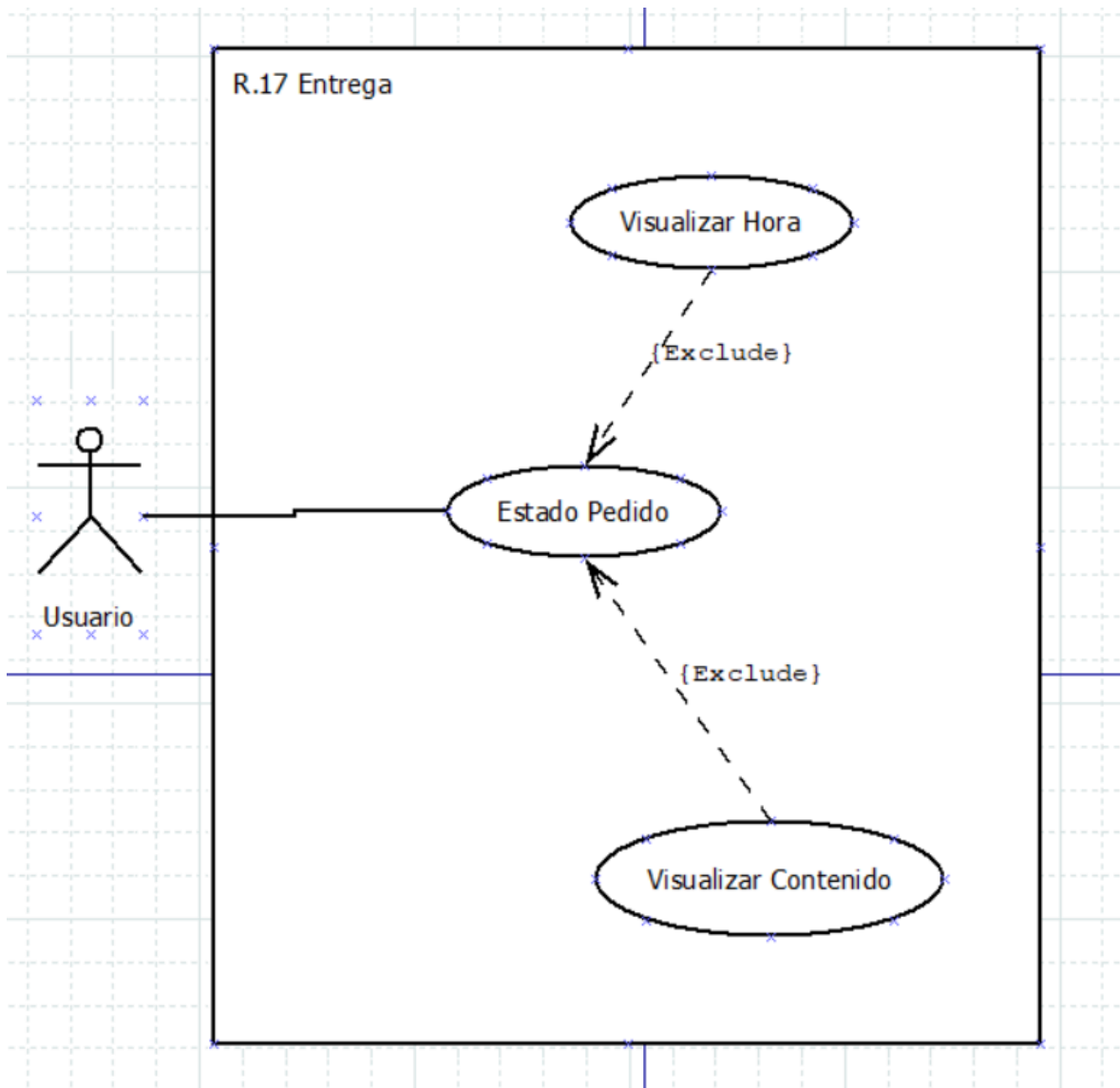
Especificación de requerimientos funcionales	
Requerimiento Funcional N°:17	Nombre
RQF017	Entrega
Tipo:	Prioridad:
Necesario	Alta

<b>Descripción:</b>
El usuario ha de poder ver el estado de su pedido, si este anda en el almacén, en camino, fue regresado, etc.
Criterios de aceptación
<ul style="list-style-type: none"><li>- El administrador ha de poder cambiar el estado de entrega del pedido</li><li>- El sistema ha de actualizar la información la cual únicamente va a ser visible para el usuario</li></ul>

***Nota: La tabla nos menciona el cómo se le ha de mostrar y poder actualizar el estado de entrega de un producto al usuario, no siendo necesario un trackeo, si no un simple cambio de estado en el nombre: “en camino, fue regresado, etc”***

**Figura 17**

**Caso de Uso n°17 [Entrega]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario al momento de revisar en qué estado se encuentra su pedido para la entrega del producto o libro que compró*

**Tabla 18: Requerimiento Funcional Historial de compra**

***Requerimiento funcional n° 18 [Historial de compra]***

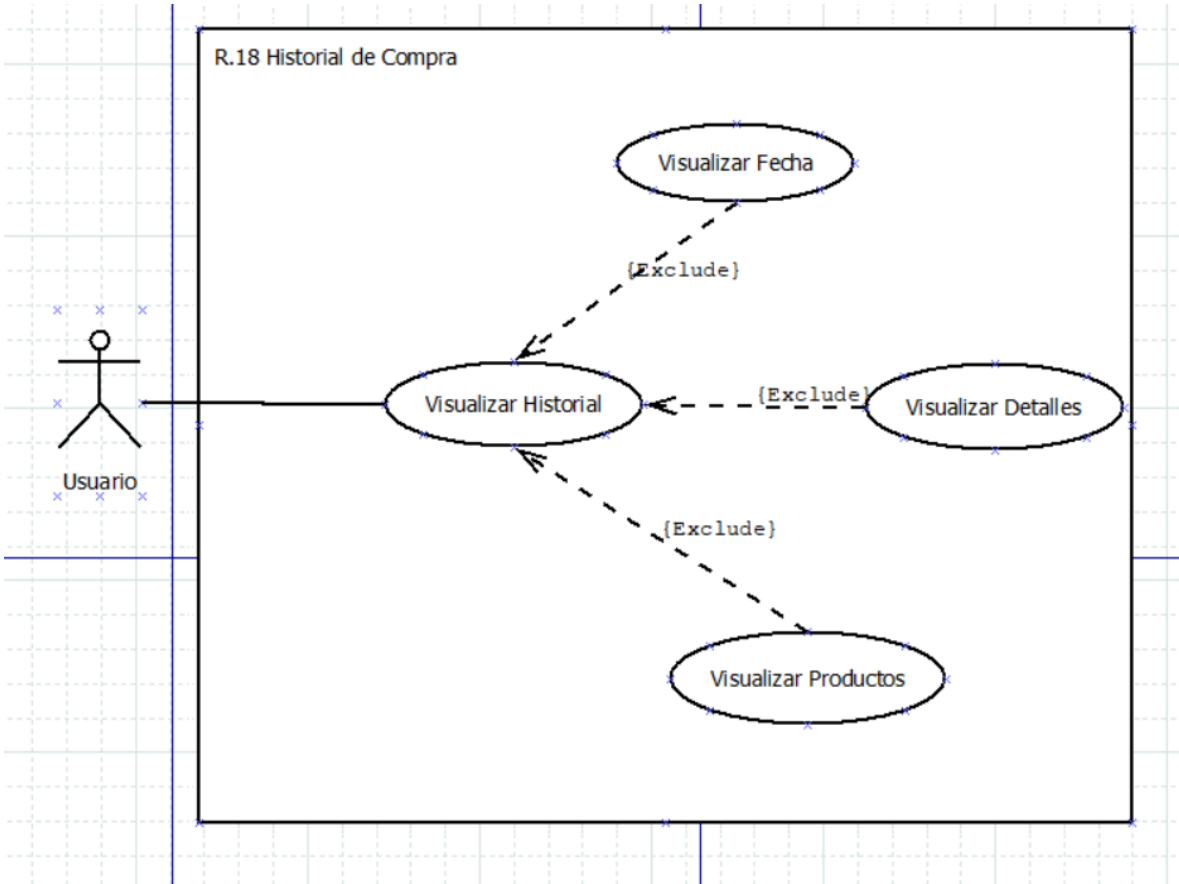
<b>Especificación de requerimientos funcionales</b>	
<b>Requerimiento Funcional N°:18</b>	<b>Nombre</b>
RQF018	Historial de compra
<b>Tipo:</b>	<b>Prioridad:</b>
Necesario	Alta
<b>Descripción:</b>	
El usuario ha de poder ver su historial de compras que ha tenido en la página	
<b>Criterios de aceptación</b>	
- El sistema ha de mostrar todas las compras que ha tenido el usuario	

***Nota: la tabla nos menciona que todos los usuarios han de tener la capacidad de poder ver su historial de compras único***



Figura 18

Caso de Uso n°18 [Historial de Compra]



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario al momento de revisar el historial de compras que ha realizado al tener una cuenta con el sitio web*

Tabla 19: Requerimiento Funcional Buscar en el registro de compra (Administrador)

*Requerimiento funcional n° 19 [Buscar en el registro de compra (Administrador)]*

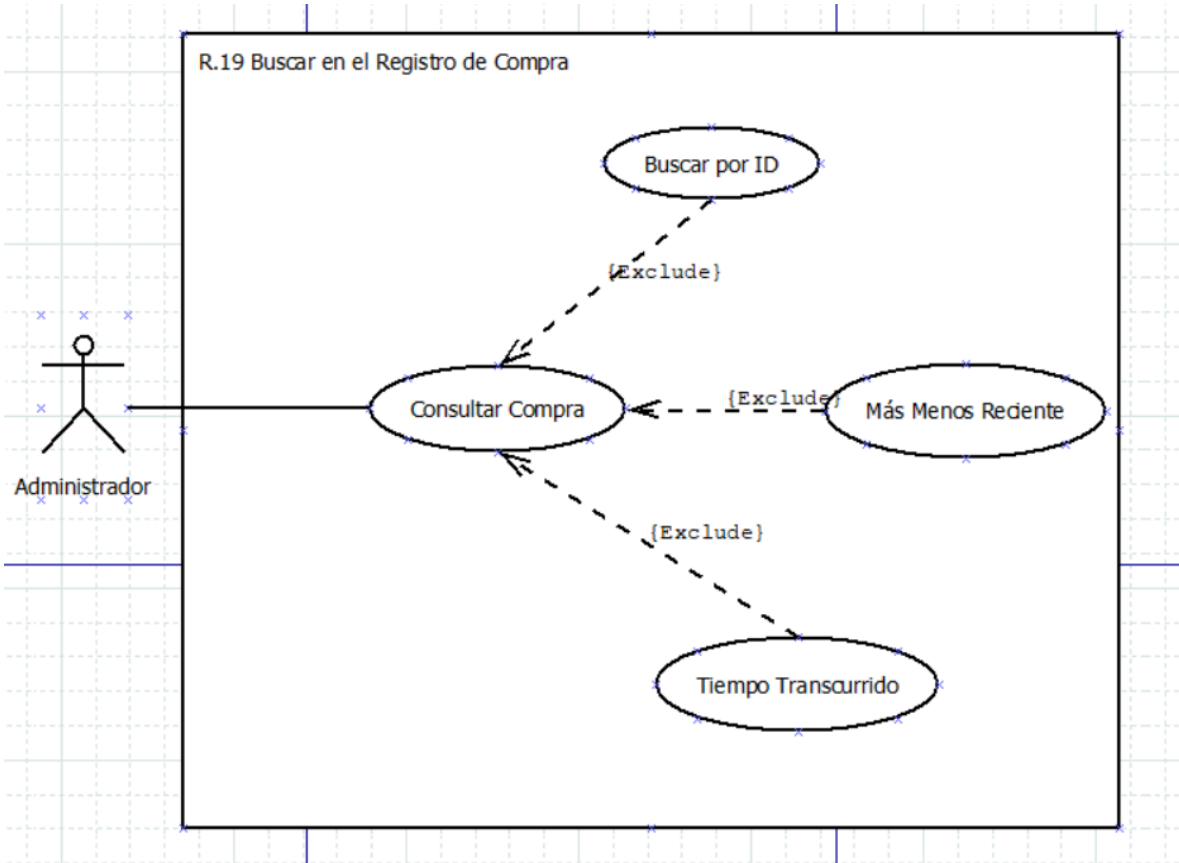
Especificación de requerimientos funcionales	
Requerimiento Funcional N°:19	Nombre

RQF019	Buscar en el registro de compra (Administrador)
<b>Tipo:</b>	<b>Prioridad:</b>
Necesario	Alta
<b>Descripción:</b>	
El administrador ha de poder consultar los registros de las compras de los usuarios en orden de más reciente a menos reciente, en un lapso de 1 mes y/o buscar por ID del recibo	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>- El sistema ha de cargar los últimos registros de compras</li> <li>- El sistema ha de permitir buscar por un registro en específico con su ID</li> </ul>	

*Nota: La tabla nos menciona que el administrador de la página ha de tener la capacidad de poder buscar los registros de compras hechos en su página desde el menos reciente al más reciente*

Figura 19

Caso de Uso n°19 [Buscar en el Registro de Compra]



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario al momento de consultar las compras que se han realizado por medio de clientes en la página*

Tabla 20: Requerimiento Funcional Reseñas

Requerimiento funcional n° 20 [Reseñas]

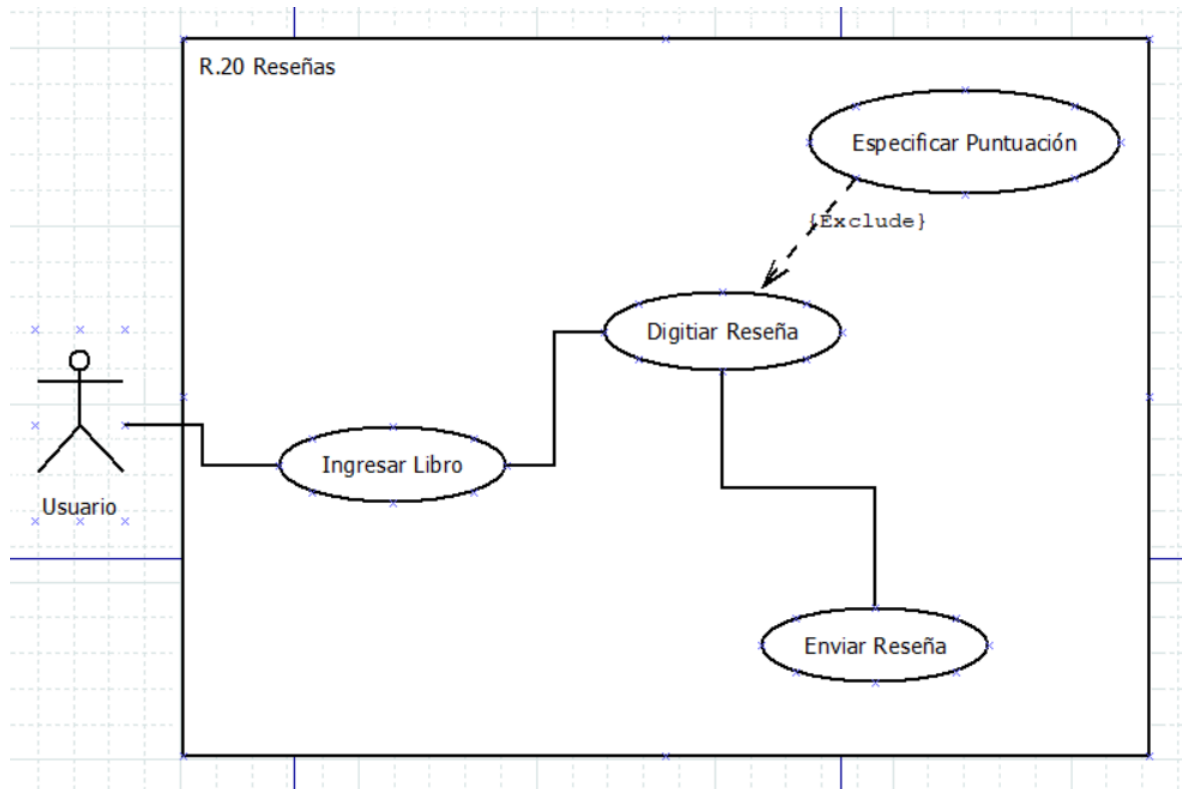
Especificación de requerimientos funcionales	
Requerimiento Funcional N°:20	Nombre
RQF020	Reseñas
Tipo:	Prioridad:

Necesario	Alta
<b>Descripción:</b>	
El usuario ha de poder dejar una reseña en el libro deseado o que esté mirando, solo si este tiene una cuenta dentro de la página y su sesión está activa en el momento de crear la reseña	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>- La reseña digitada no debe ser vacía</li> <li>- La reseña del usuario se ha de guardar correctamente en la base de datos</li> <li>- La reseña del usuario se ha de mostrar a demás usuarios inclusive a él mismo</li> </ul>	

*Nota: La tabla nos menciona las capacidades del usuario dentro de la página al momento de escribir una reseña en algún libro*

**Figura 20**

**Caso de Uso n°20 [Reseñas]**



*Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario al momento de realizar las reseñas necesarias para los libros que quiera dentro de la página*

**Tabla 21: Requerimiento Funcional Procesamiento de pagos**

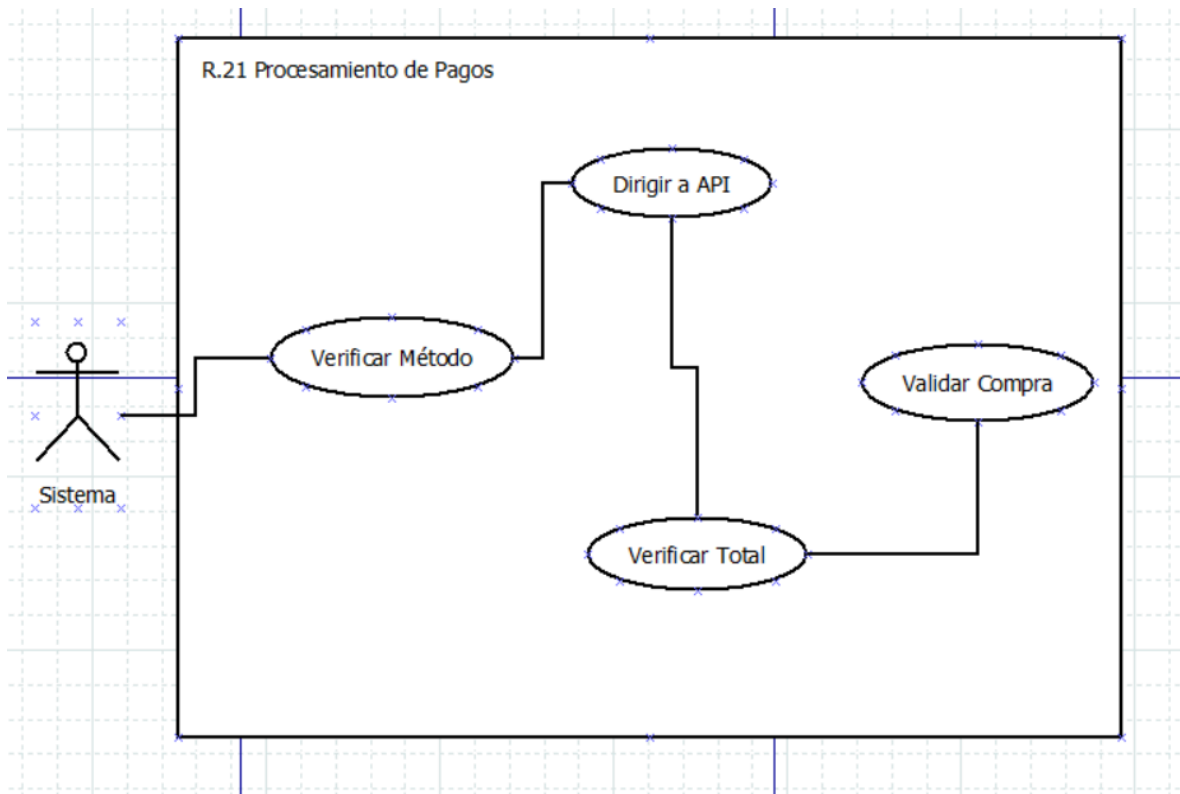
***Requerimiento funcional n° 21 [Procesamiento de pagos]***

Especificación de requerimientos funcionales	
Requerimiento Funcional N°:21	Nombre
RQF021	Procesamiento de pagos
Tipo:	Prioridad:
Necesario	Alta
Descripción:	
El sistema ha de procesar los pagos y verificar la validez del método de pago seleccionado por el usuario	
Criterios de aceptación	
<ul style="list-style-type: none"><li>• El pago debe ser válido</li><li>• En caso de que el pago sea negado se le ha de informar al usuario que su pago no fue procesado</li></ul>	

*Nota: Esta tabla menciona que la pasarela de base de datos usada ha de procesar correctamente los pagos y que ha de ocurrir en caso de que falle*

**Figura 21**

***Caso de Uso n°21 [Procesamiento de Pagos]***



***Nota: Esta figura ilustra mediante caso de uso la funcionalidad e interacción del usuario al momento de realizar la compra, en este caso se muestra como el sistema se comporta redirigiendo a la API encargada de generar el pago y validar la compra***

---

## 4 DISEÑO

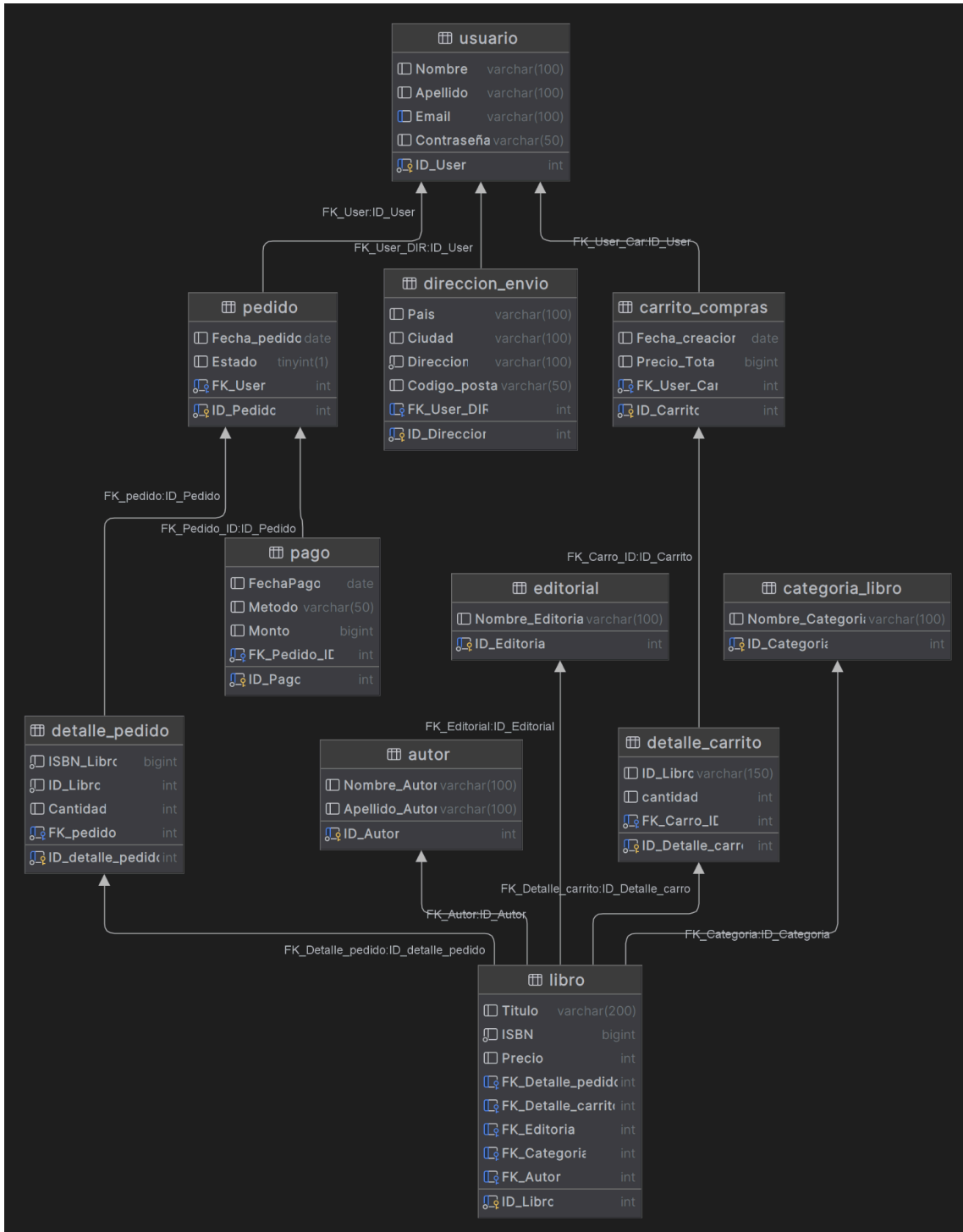
---

### 4.1 Diseño de Físico de la Base de datos

La base de datos se llama Librería BD, será desarrollada en MySQL. Su objetivo es modelar el flujo de compra de libros en línea. Gestionando usuarios, pedidos, detalles de libros ,ETC, para l cual se implementarán las siguientes entidades:

- Tabla usuario
- Restricciones: El campo Email es único.
- Tabla pedido
- Tabla pedido
- Tabla pago
- Tabla Detalle\_Pedido
- Tabla Direccion\_Envio
- Tabla Carrito\_Compras
- Tabla Detalle\_Carrito
- Tabla Editorial
- Tabla Categoria\_Libro
- Tabla Autor
- Tabla Libro

**Figura 2**  
Diagrama Base de Datos



*Nota: Autoría propia hecho con DataGrip IDE*



## 4.2 Diseño de arquitectura funcional

Su objetivo es realizar un diseño de la arquitectura funcional del sistema.

Para éste punto se deben especificar las interrelaciones entre todos los módulos del sw (propios y aquellos utilizados desde librerías) y la dependencia de ellos. Para realizar esta actividad se puede construir un Árbol de descomposición funcional, como se muestra en el ejemplo para un sistema de biblioteca en la página de éste documento. Es importante que considere que los últimos niveles de descomposición del árbol corresponderá a los módulos que serán codificados (por ejemplo, librerías, procedimientos - funciones), por lo tanto es importante el nivel de detalle al que se llegue sin olvidar "*implementar módulos reutilizables, maximizar la cohesión y minimizar el acoplamiento*"

Los primeros niveles de descomposición de éste árbol debe ser consistente con el Diagrama de Casos de Uso o el Diagrama de flujos de datos, según corresponda.

## 4.3 Diseño interfaz y navegación

El diseño de la interfaz de usuario debe considerar un diseño estándar que será respetado en todas las pantallas. En el diseño se considera la organización y el aspecto de la interfaz. El aspecto considera muchos elementos, entre ellos, los colores, imágenes de fondo, uso de iconos entre otros.

La organización de una pantalla considera la ubicación de cada uno de los tipos de elementos de la interfaz, considerando por ejemplo las siguientes áreas: (ver ítem 19.2, página )

- De ingresos de datos
- De Botones de opción general
- De botones de opciones específicas a la ventana
- De Menús
- De títulos
- De Barras de Herramientas
- De pie de página
- De Encabezados
- De Logos

El diseño de menú/ navegación considera las opciones / medios que tendrá el usuario para acceder a la funcionalidad del Sw,

Debe considerar:

- Nombre de ítem y opciones representativas para el usuario
- Organización/ jerarquía representativas para el usuario
- Facilidad de acceso a opciones relacionadas

La jerarquía de menú solo representa los anidamientos y agrupaciones de las opciones de menú y el mapa de navegación representa las opciones que tendrá el usuario para "navegar / recorrer" dentro de las distintas opciones (ver Diagrama para representar la jerarquía de menú, página ).

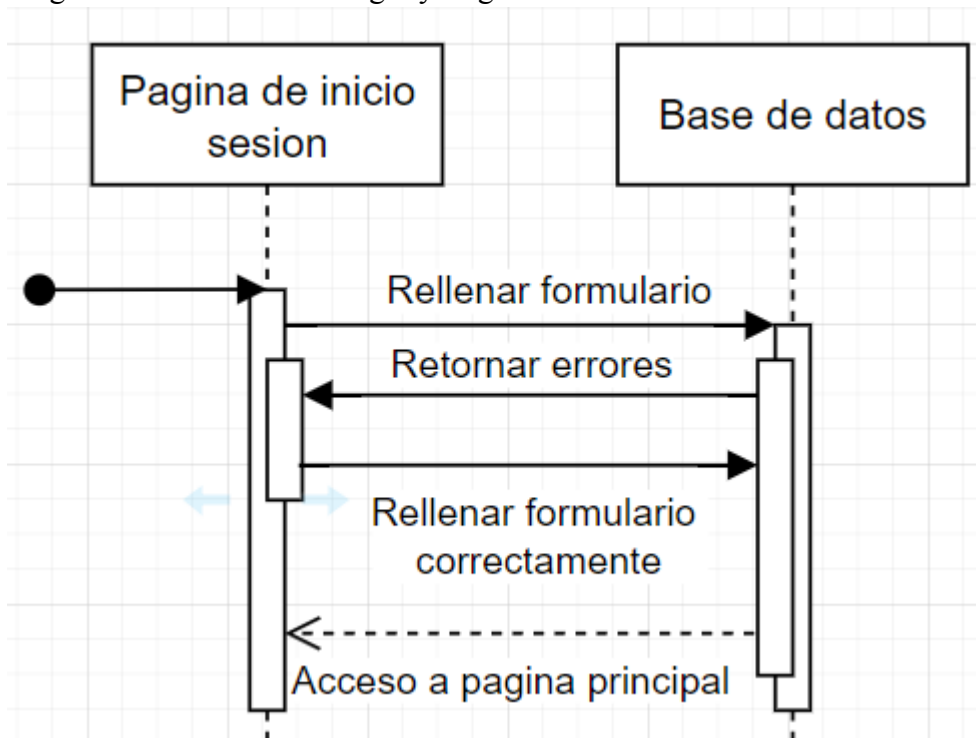
#### 4.4 Estructura del sistema

Para la estructuración del sistema se plantearon diagramas de secuencia, flujo y actividades. Buscando explicar y facilitar el manejo de datos de los módulos, por parte del sistema, para los futuros usuarios de este.

##### Módulo Login y registro:

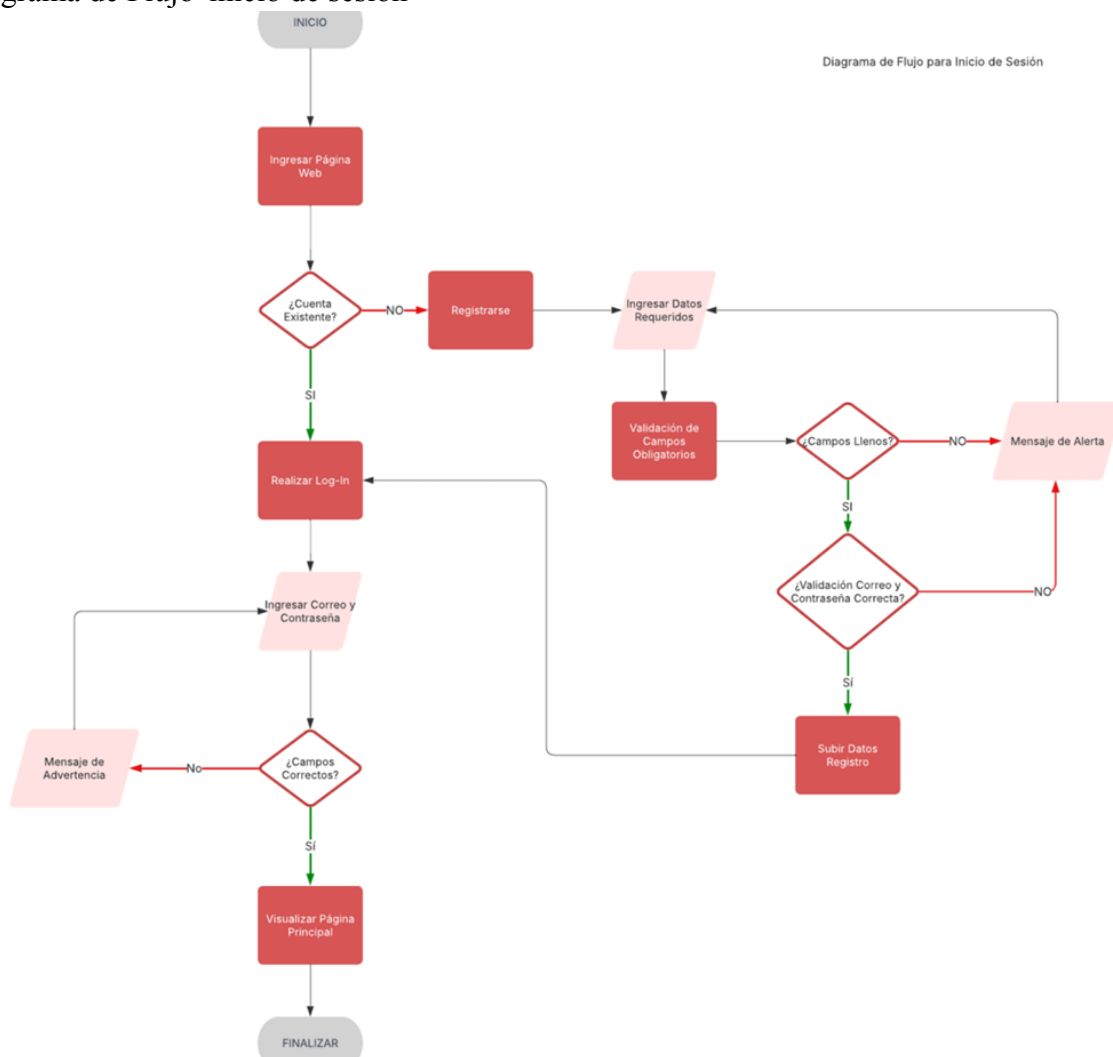
##### *Figura*

Diagrama de actividades “Login y Registro”



*Nota: Autoría propia hecho con Draw.io*

**Figura 3**  
Diagrama de Flujo “inicio de sesion”

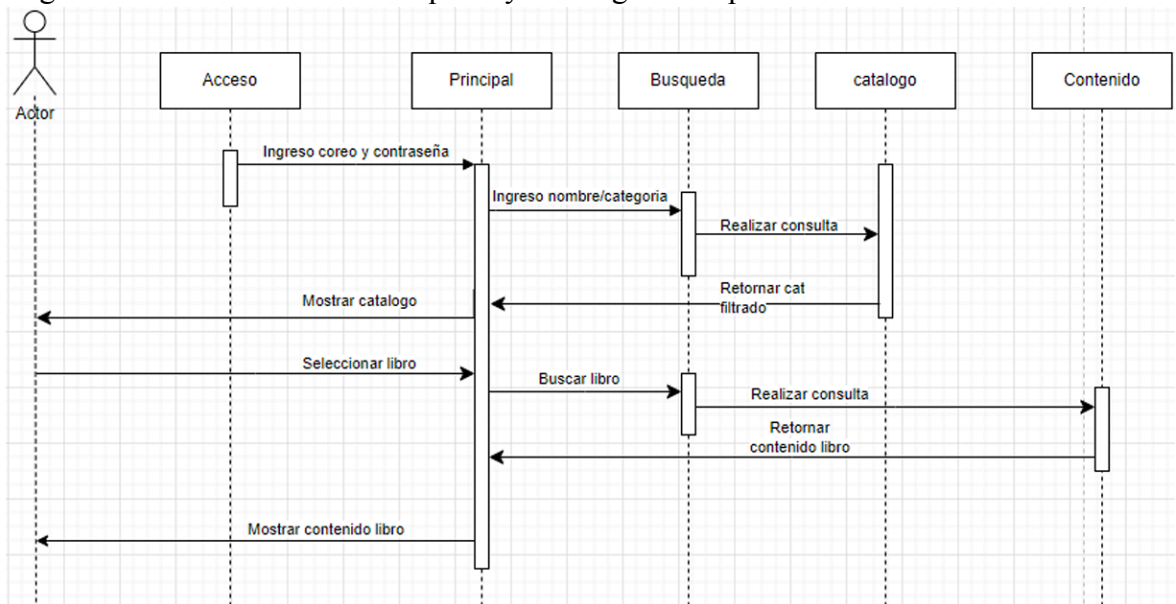


**Nota:** Autoría propia hecho con Whimsical

**Módulo principal:**

**Figura 5**

Diagrama de actividades de “Búsqueda y Catálogo Principal”

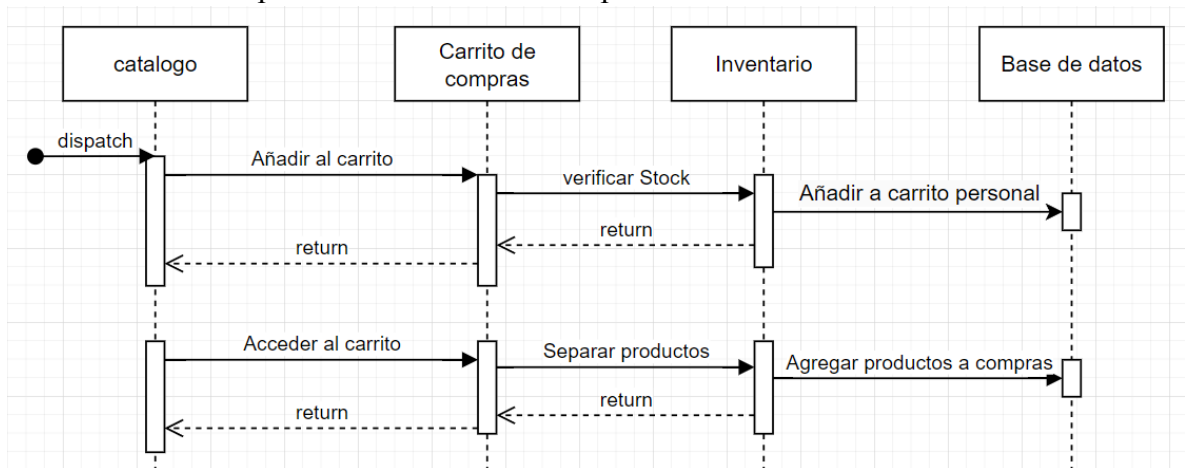


*Nota: Autoría propia hecho con Draw.io*

**Módulo compras:**

**Figura**

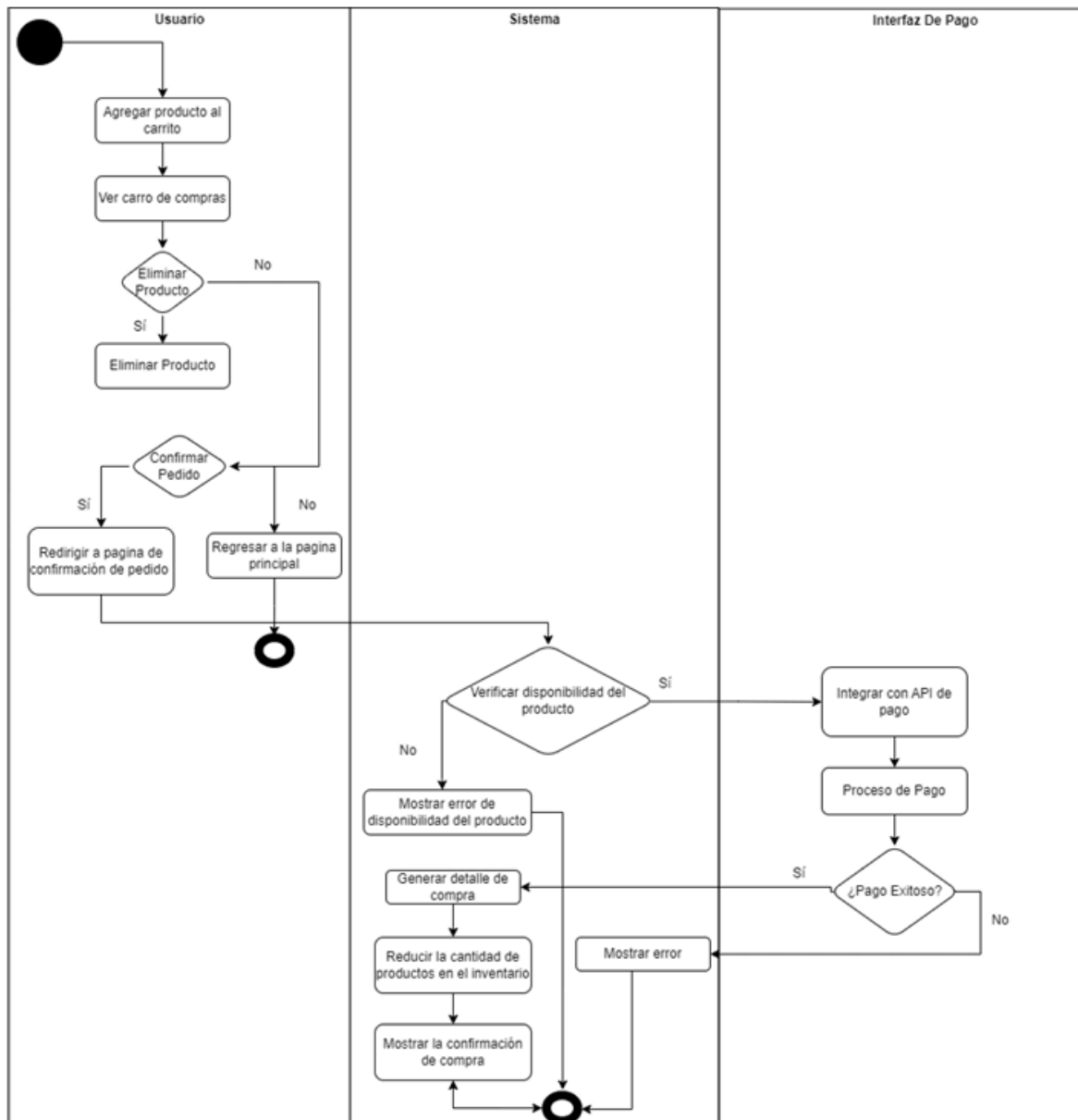
Diagrama de actividades “procesos de carrito de compra”



*Nota: Autoría propia hecho con Draw.io*

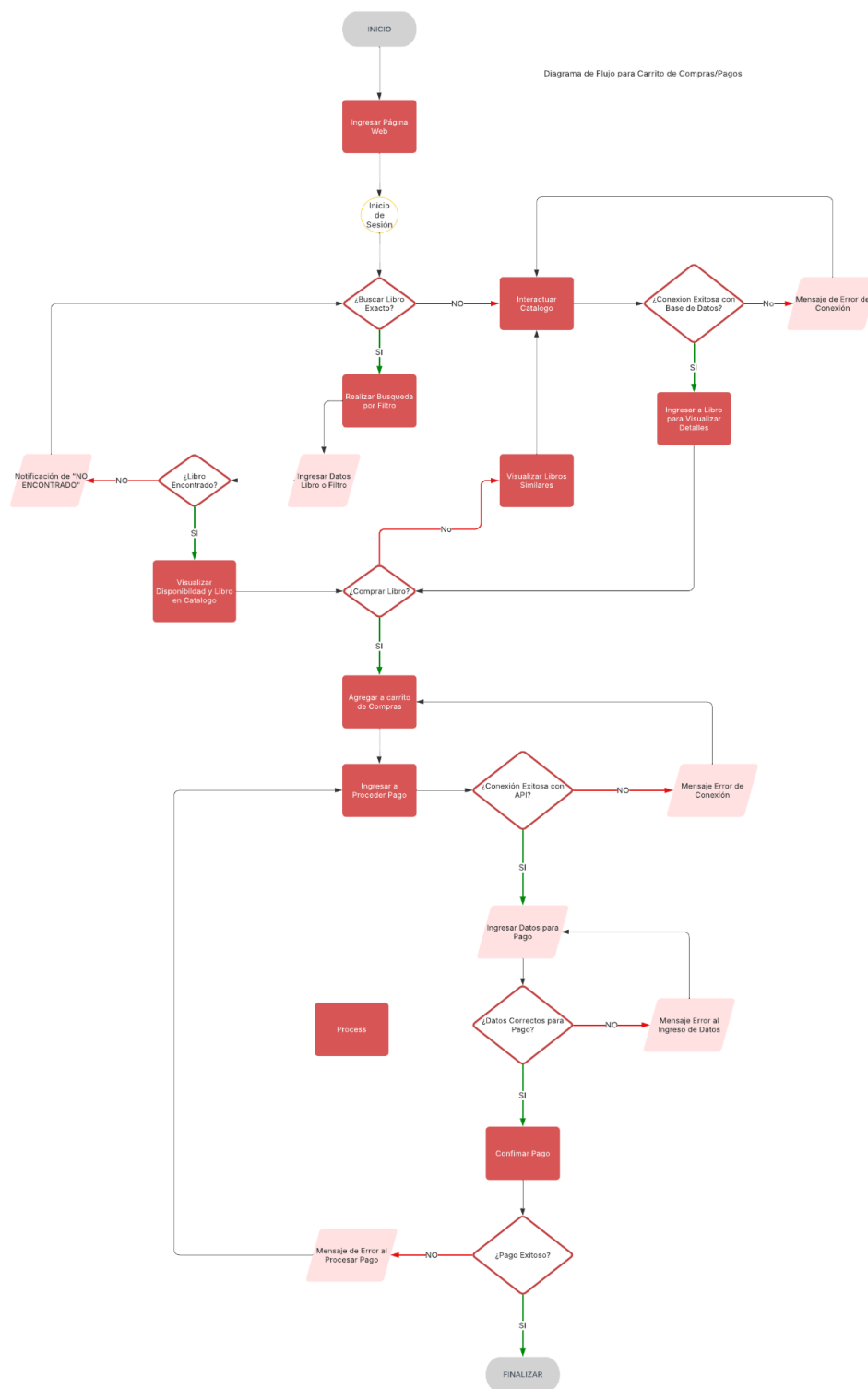
**Figura**

Diagrama de actividades “procesos de carrito de compra”



*Nota: Autoría propia hecho con Draw.io*

**Figura**  
Diagrama de Flujo “Carrito de compras”



**Nota:** Autoría propia hecho con Whimsical

---

## 5 PRUEBAS

---

### 5.1 Elementos de prueba

#### Módulo de Registro:

Para realizar las pruebas en el módulo de registro, se ingresaron los datos ( "Laura", "Gomez", "[laura@example.com](mailto:laura@example.com)", "contraseña123", "3123456789") para cumplir con los campos requeridos (nombre, apellido, email, password, telefono).

Estos datos fueron modificados en diferentes envíos, para testear posibles errores, y complicaciones en el funcionamiento.

#### Modulo de Login:

### 5.2 Especificación de las pruebas

#### Módulo de Registro y Login:

Método	Ruta	Descripción
POST	<a href="http://localhost:3000/registro">http://localhost:3000/registro</a>	Al momento de realizar la prueba de integración de la ruta POST, manda error ya que la tabla USUARIOS de la DB le falta el campo Número de teléfono. En el formulario de registro el atributo ID del campo contraseña y apellido no coincide con los de la DB, esto genera errores al momento de realizar consultas SQL a través del servidor.
POST	<a href="http://localhost:3000/login">http://localhost:3000/login</a>	Al momento de realizar la prueba de integración de la ruta POST, obtiene los datos del correo y la contraseña para contrastarlos con los guardados en la base de datos.

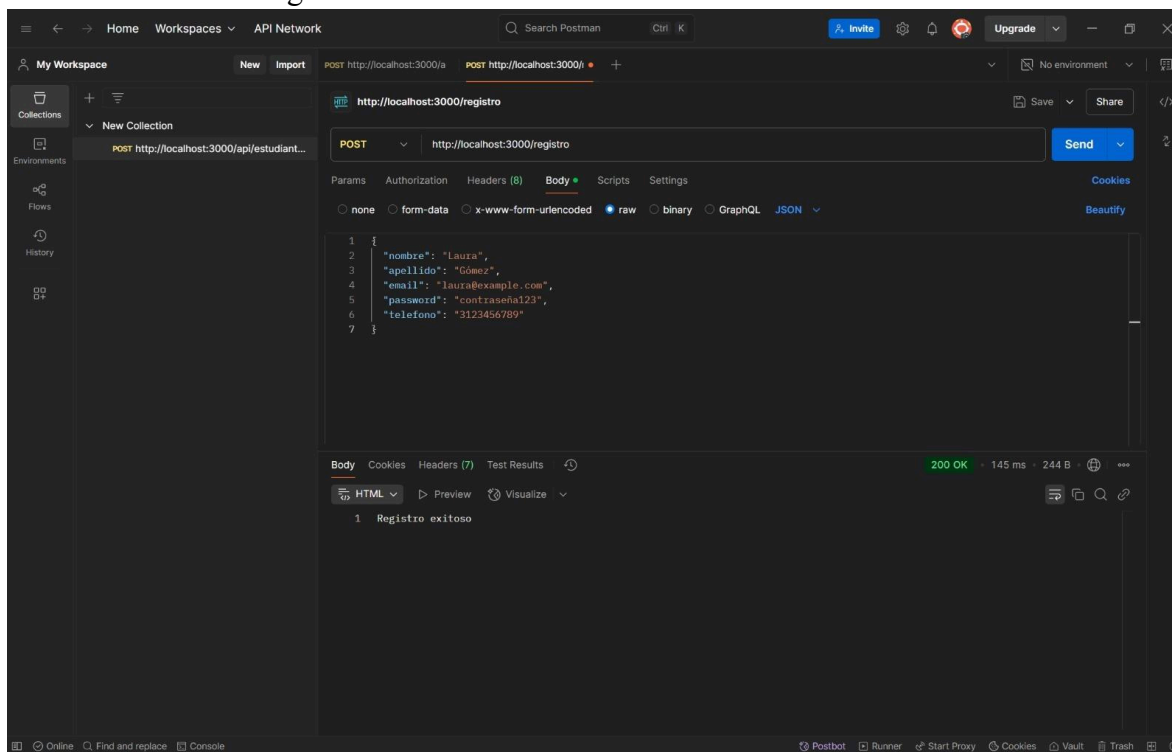
### Módulo de Carrito:

Método	Ruta	Descripción
GET	<a href="http://localhost:3300/carrito">http://localhost:3300/carrito</a>	
POST	<a href="http://localhost:3300/carrito">http://localhost:3300/carrito</a>	
DELETE	<a href="http://localhost:3300/carrito/1">http://localhost:3300/carrito/1</a>	
PUT	<a href="http://localhost:3300/carrito/1">http://localhost:3300/carrito/1</a>	

## 5.2.1 Pantallazos Pruebas Unitarias

**Figura 3**

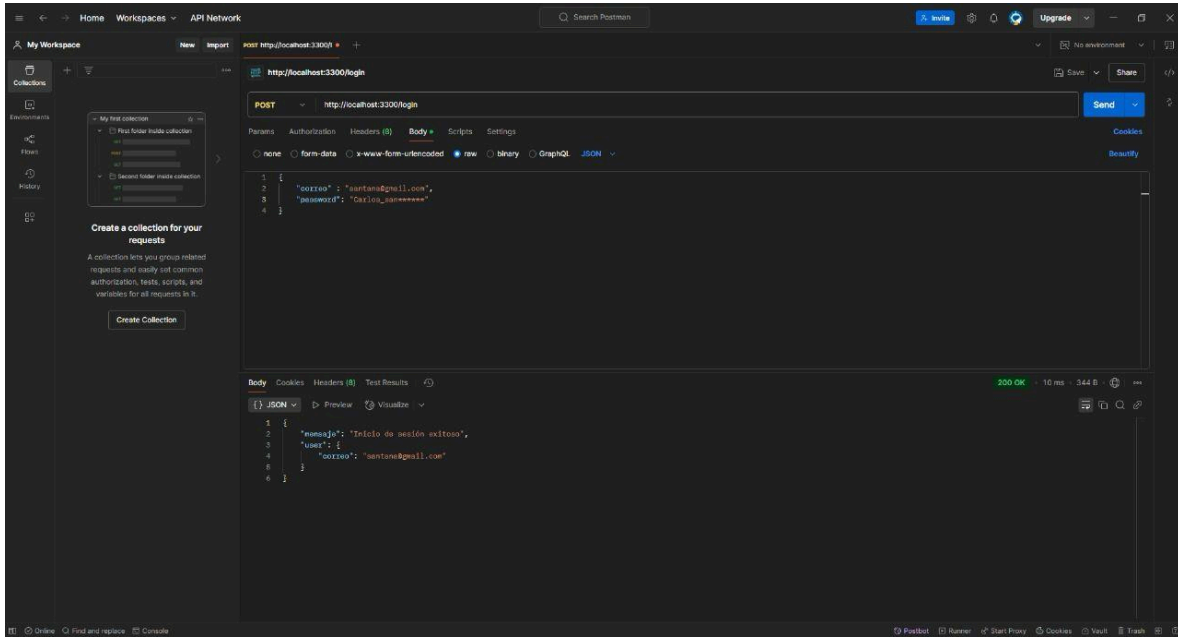
Prueba Unitaria de Registro



**Nota:** Prueba realizada en Postman, se observa que se conecta mediante el puerto 3000 en el Endpoint “<http://localhost:3000/registro>” registrando los datos exitosamente.

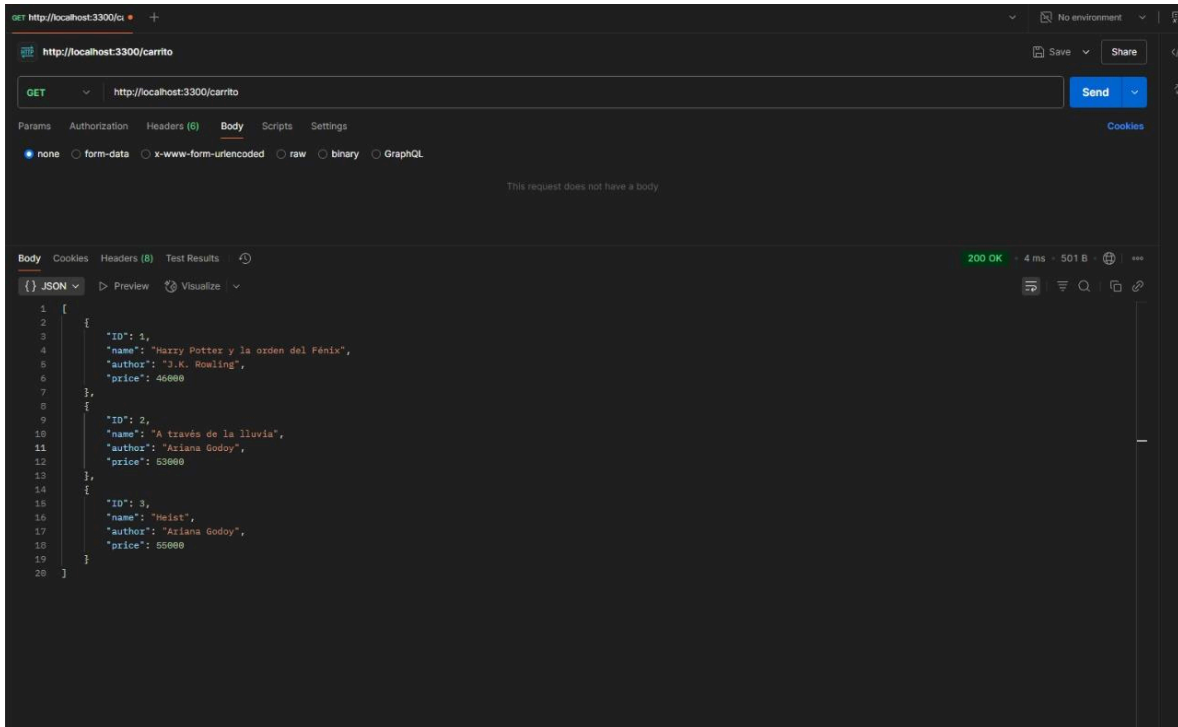


**Figura 4**  
Prueba Unitaria de Login



**Nota:** Prueba realizada en Postman, se observa que se conecta mediante el puerto 3000 en el Endpoint ["http://localhost:3300/login"](http://localhost:3300/login) registrando los datos exitosamente.

**Figura 5**  
Prueba Unitaria de Carito GET

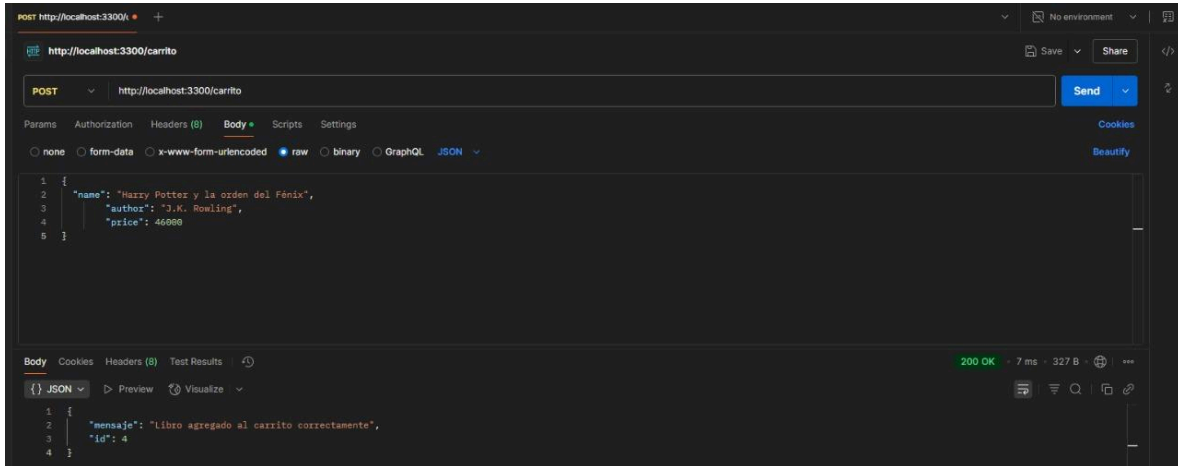


Entre Líneas

***Nota: Prueba realizada en Postman, se observa que se conecta mediante el puerto 3300 en el Endpoint "<http://localhost:3300/carrito>" obteniendo los datos guardados en la base de datos mostrados en la prueba.***

**Figura 6**

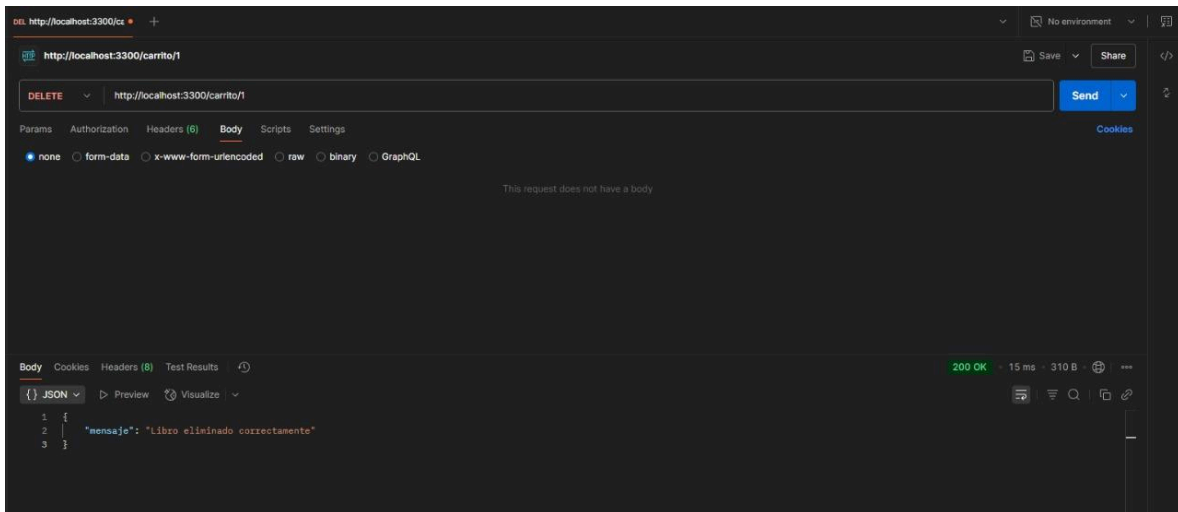
Prueba Unitaria de Carito POST



***Nota: Prueba realizada en Postman, se observa que se conecta mediante el puerto 3300 en el Endpoint "<http://localhost:3300/carrito>" mostrando los datos que se encuentran en la base de datos.***

**Figura 7**

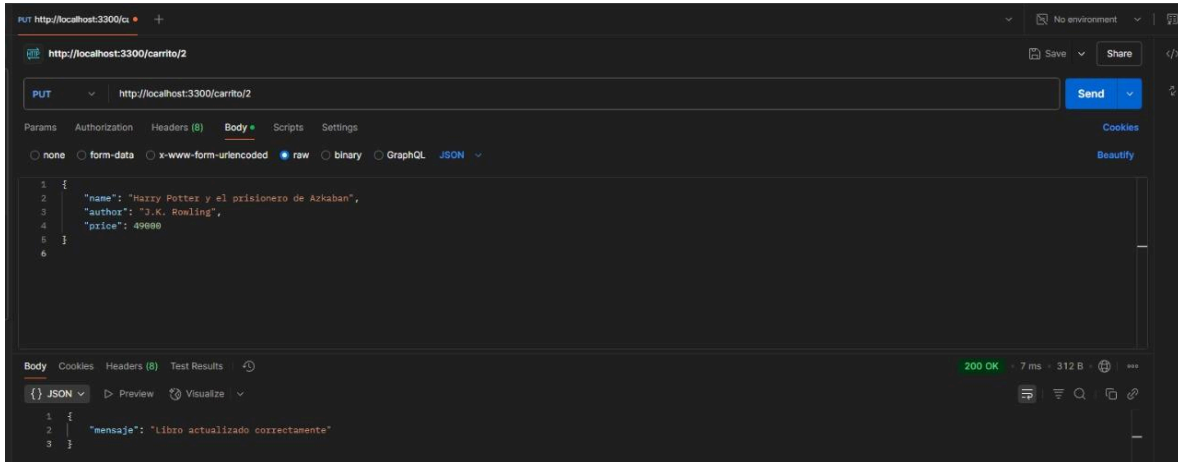
Prueba Unitaria de Carito DELETE



***Nota: Prueba realizada en Postman, se observa que se conecta mediante el puerto 3300 en el Endpoint "<http://localhost:3300/carrito/1>" borrando los datos registrados en la base de datos mediante el ID registrado.***

**Figura 8**

Prueba Unitaria de Carito PUT



*Nota: Prueba realizada en Postman, se observa que se conecta mediante el puerto 3300 en el Endpoint ["http://localhost:3300/carrito/2"](http://localhost:3300/carrito/2) modificando los datos ingresados identificando los campos por medio del ID en la base de datos.*

### 5.3 Responsables de las pruebas

#### Registro:

- Grupo de Integración (Juan David Serrano, Miguel Angel Romero, Laura Gaona Castaño, Andres Leonardo Pinzon Benites)

#### Login:

- Grupo de Integración (Juan David Serrano, Miguel Angel Romero, Laura Gaona Castaño, Andres Leonardo Pinzon Benites)

#### Carrito CRUD:

- Santiago Perez, Andres Forero

### 5.4 Conclusiones de las Pruebas Unitarias

Concluya respecto al proceso y énfasis de las pruebas realizadas, así como en los resultados.

---

## 6 CONCLUSIONES

---

En primera instancia el alumno debe hacer la contrastación de los objetivos del proyecto y del sistema planteados y alcanzado al final del proyecto.

Se planean conclusiones respecto al ajuste de las herramientas, lenguajes o metodologías utilizadas y la planificación inicial del proyecto.

Para terminar se incluyen conclusiones generales del proyecto desde los puntos de vista:

- Académico
- Personal

---

## **7 BIBLIOGRAFÍA**

---

Formato de referencias y bibliografía según los estándares de biblioteca.

---

## **8 ANEXO: MANUAL DE USUARIO**

---

Según se requiera.

---

## **9 ANEXO: MANUAL TÉCNICO**

---