

Recopilación, Análisis y Gestión de Requerimientos Diseño de la Solución

<<Sistema de gestión de libros>>

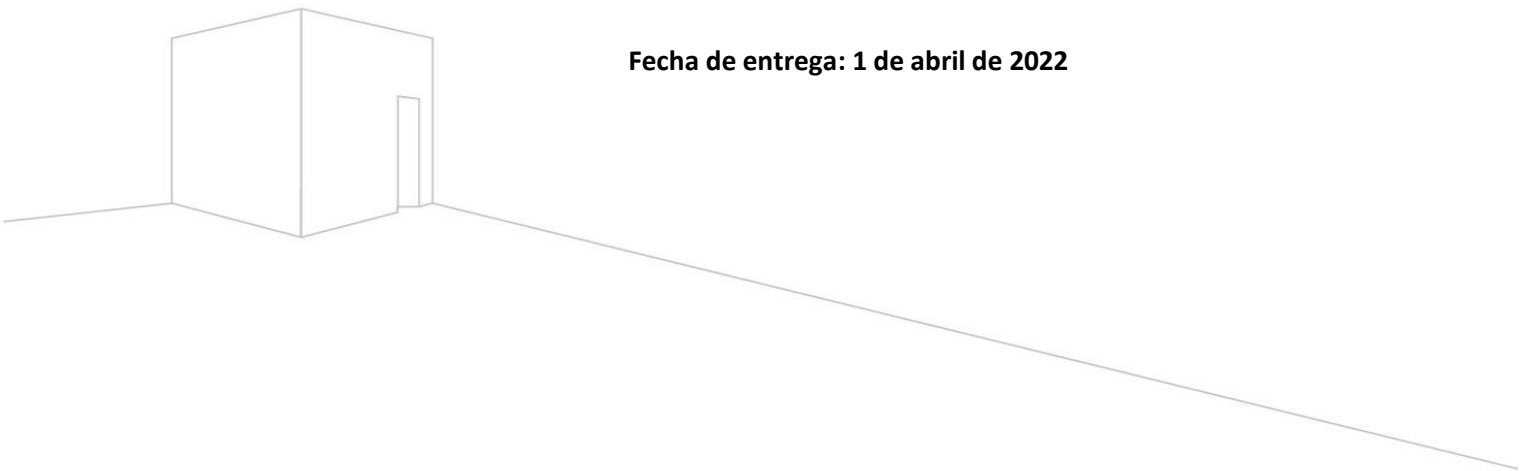
Asignatura: Taller de Integración de Software

Sección: B50

Nombre del académico: Igor Alejandro Cáceres Padilla

Nombre de los integrantes del grupo: Carlos Zuleta, Patricio Koñajagua, Martin Ardiles, Luis Jiménez

Fecha de entrega: 1 de abril de 2022



Contenido

I.	Introducción	3
II.	Casos de Uso	3
a.	Casos de Uso De Alto Nivel	5
b.	Casos de Uso a Nivel Detallado	6
III.	Descripción de Casos de Uso	7
IV.	Diagrama de clases	8
V.	Diagrama de Secuencia	9
VI.	Modelo de Datos	11
VII.	Desarrollo del Prototipo	12
VIII.	Conclusiones	Error! Bookmark not defined.
IX.	Referencias bibliográficas	15
X.	Anexos	Error! Bookmark not defined.

I. Introducción

En base a los requerimientos seleccionados en su presente proyecto y documentados según la norma IEEE830 (que debe considerar como anexo del presente documento), deberá confeccionar los siguientes diagramas que permitirán comprender de mejor forma la manera en que estos requerimientos serán plasmados en el sistema solución.

- Diagramas de casos de uso: Alto nivel, detallado y descripción.
- Diagrama de clases
- Diagrama de secuencia
- Modelo de datos
- Prototipo conceptual

II. Casos de Uso

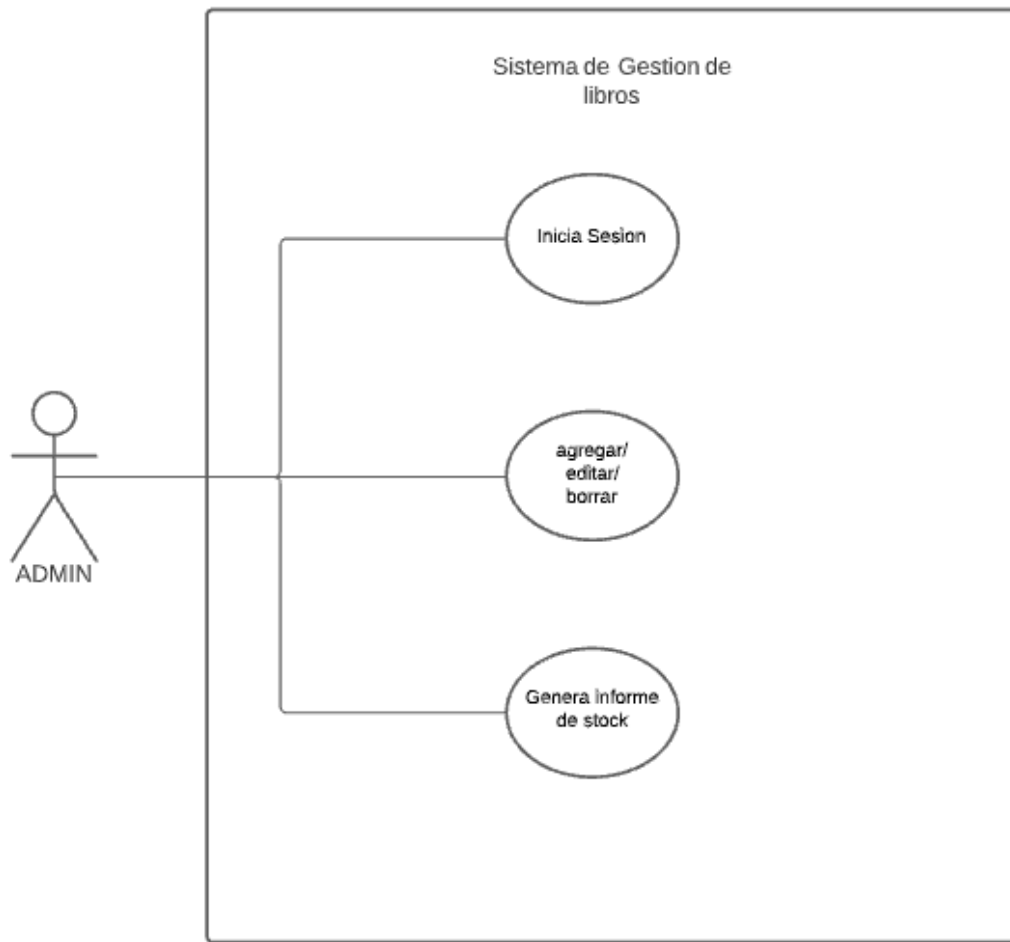
Los casos de uso son una técnica de documentación de requisitos muy extendida, entre otros motivos porque UML le da soporte. Se trata de un enfoque en la manera de documentar requisitos que permite utilizar varios grados de detalle y de formalismo, lo cual los hace adecuados a escenarios muy diversos.

Un caso de uso recoge el contrato entre el sistema y los stakeholders mediante la descripción del comportamiento observable del sistema.

Un actor es una persona, una organización o un sistema informático que tiene capacidad de interactuar con el sistema y que presenta un comportamiento propio. Cada caso de uso tiene un actor llamado actor principal, que es quien utiliza el sistema para satisfacer un objetivo. El caso de uso describe cuál es el comportamiento observable del sistema durante esta interacción.

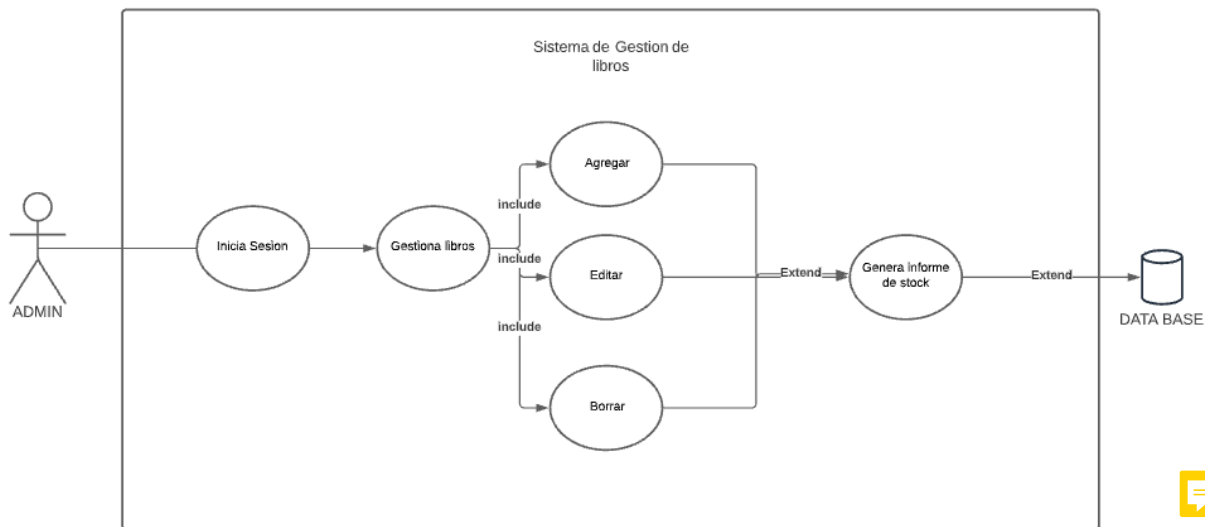
Sin embargo, en un mismo caso de uso, además del actor principal, pueden aparecer uno o más actores de apoyo, también denominados secundarios. Estos son actores externos al sistema que le proporcionan un servicio.

Utilizando System Architect u otra herramienta de modelamiento de su preferencia, documente los requerimientos utilizando diagramas de casos de uso.



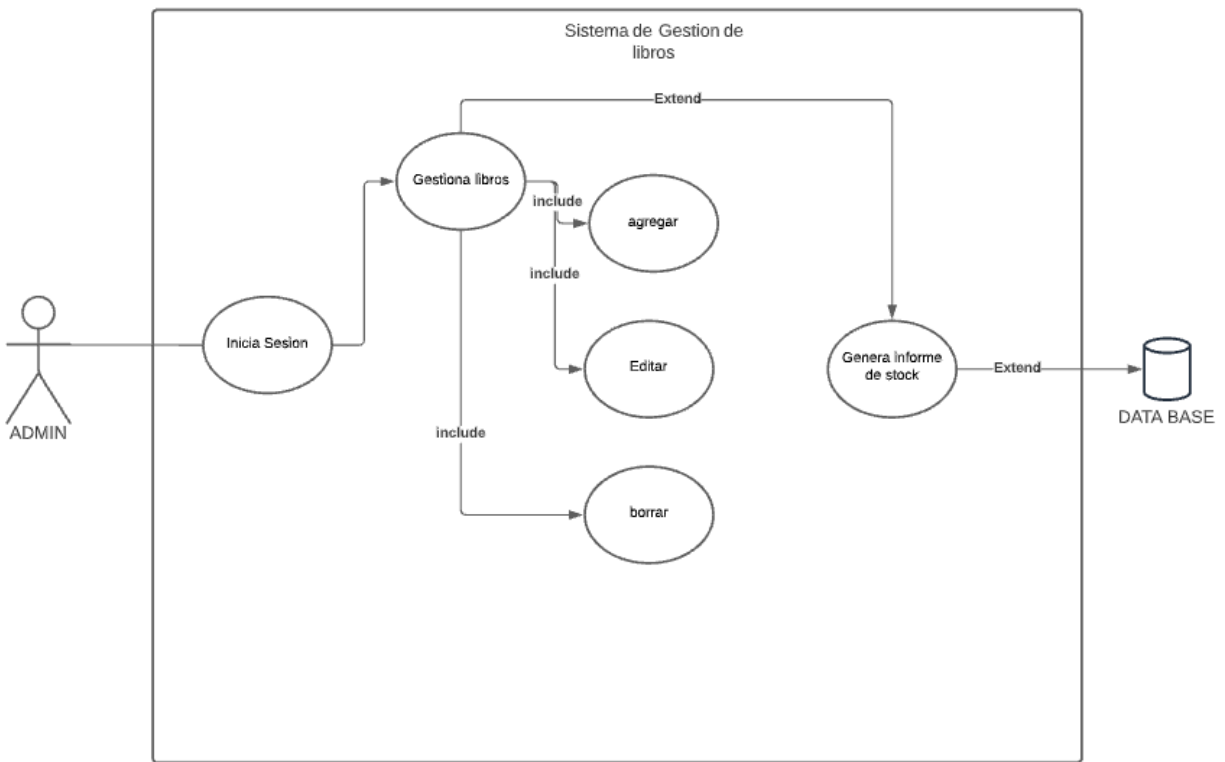
a. Casos de Uso De Alto Nivel

Este diagrama representa la funcionalidad completa de un sistema (o una clase) mostrando su interacción con los agentes externos. Esta representación se hace a través de las relaciones entre los actores (agentes externos) y los casos de uso (acciones) dentro del sistema. Los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esos conjuntos de funcionalidades son representados por los casos de uso. Se pueden visualizar como las funciones más importantes que la aplicación puede realizar o como las opciones presentes en el menú de la aplicación.



b. Casos de Uso a Nivel Detallado

Hacen referencia a la descomposición de los casos de uso del punto anterior. Se dan cuando existe una relación entre dos casos de uso. Dicha relación puede ser de extensión, que en términos de la Orientación a Objetos es una relación de herencia, donde el “sub caso” especializa al caso. También puede ser una relación de “uso”, donde el caso requiere que el sub caso se realice completamente para que él mismo se realice bien y completamente.



III. Descripción de Casos de Uso

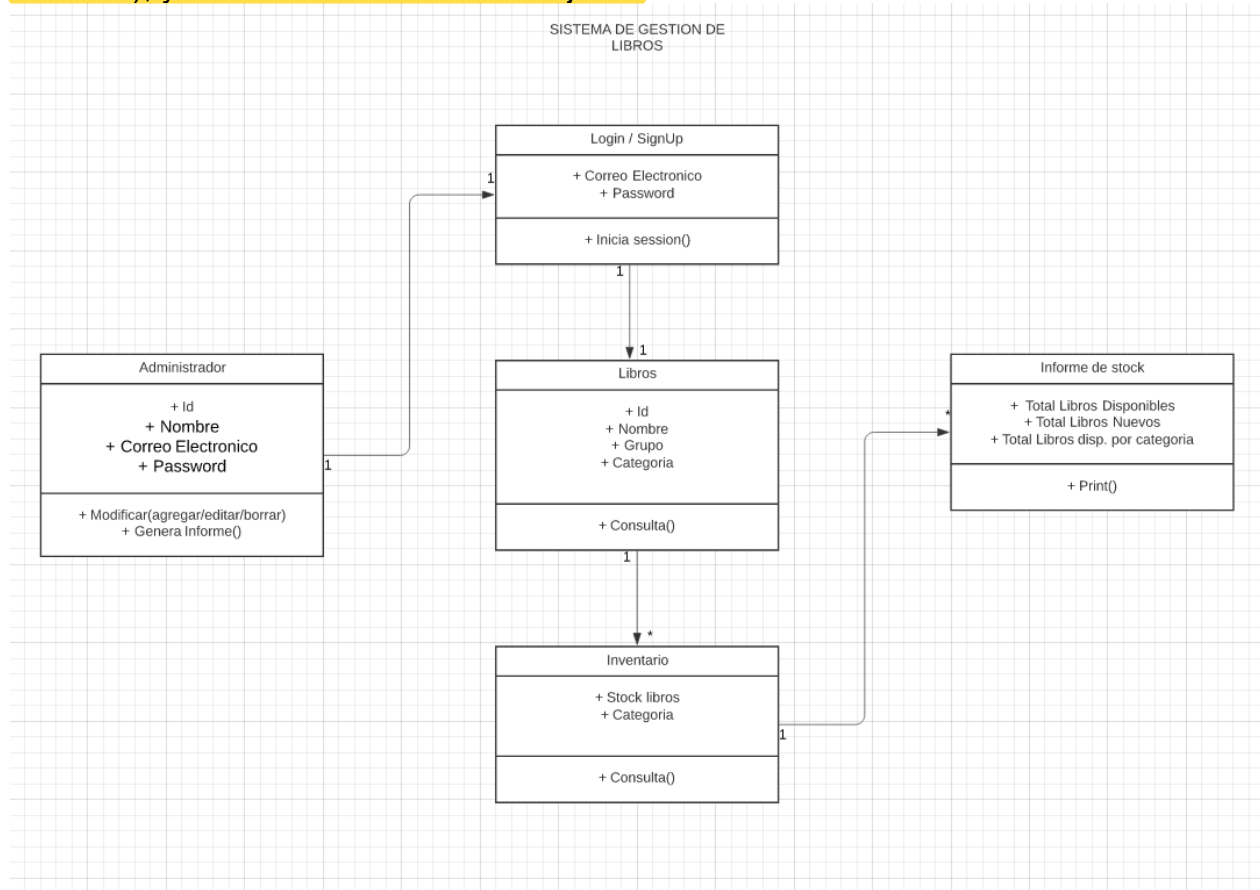
Este formato muestra una descripción para ayudar a comprender los Casos y Sub-Casos de Uso. También hace referencia a los requerimientos consignados en el documento de Requerimientos, con los cuales tiene relación, respete el siguiente formato para documentar cada caso de uso (se completaron los campos a modo de ejemplo):



Caso de uso	Sistema de Gestión de libros
Actor principal	Administrador
Ámbito	Administración y Gestión de libros
Nivel de objetivo	Usuario Administrador
Stakeholders e intereses	Administrador: Requiere que el sistema de gestión de libro sea lo más eficiente posible en control de inventario y búsqueda de libros
Precondición	Administrador habilitado para utilizar la interfaz web
Garantías mínimas	
Garantías en caso de éxito	Se realizará la gestión de libros de forma expedita y administrador podrá ampliar el stock de libros.
Escenario principal de éxito	<ol style="list-style-type: none"> 1) El Administrador indica inicio sesión 2) El Administrador registra los libros y cantidad de ellos, en el sistema uno a uno a través del lector de código. 3) El sistema muestra en pantalla descripción del producto, y si está disponible y/o no disponible en área visible 4) Cuando el administrador ingresa el último libro indica "fin de la actualización" 5) sistema calcula total de los libros disponible, detallado para generar un informe de stock de libros disponibles para su lectura. 6) El Administrador consulta a la base de datos, para confirmar la actualización de stock de libros 7) El sistema emitirá el informe detallado de libros disponibles para su lectura.
Extensiones	2a. Se puede registrar el producto del libro en forma manual digitando el código del libro.

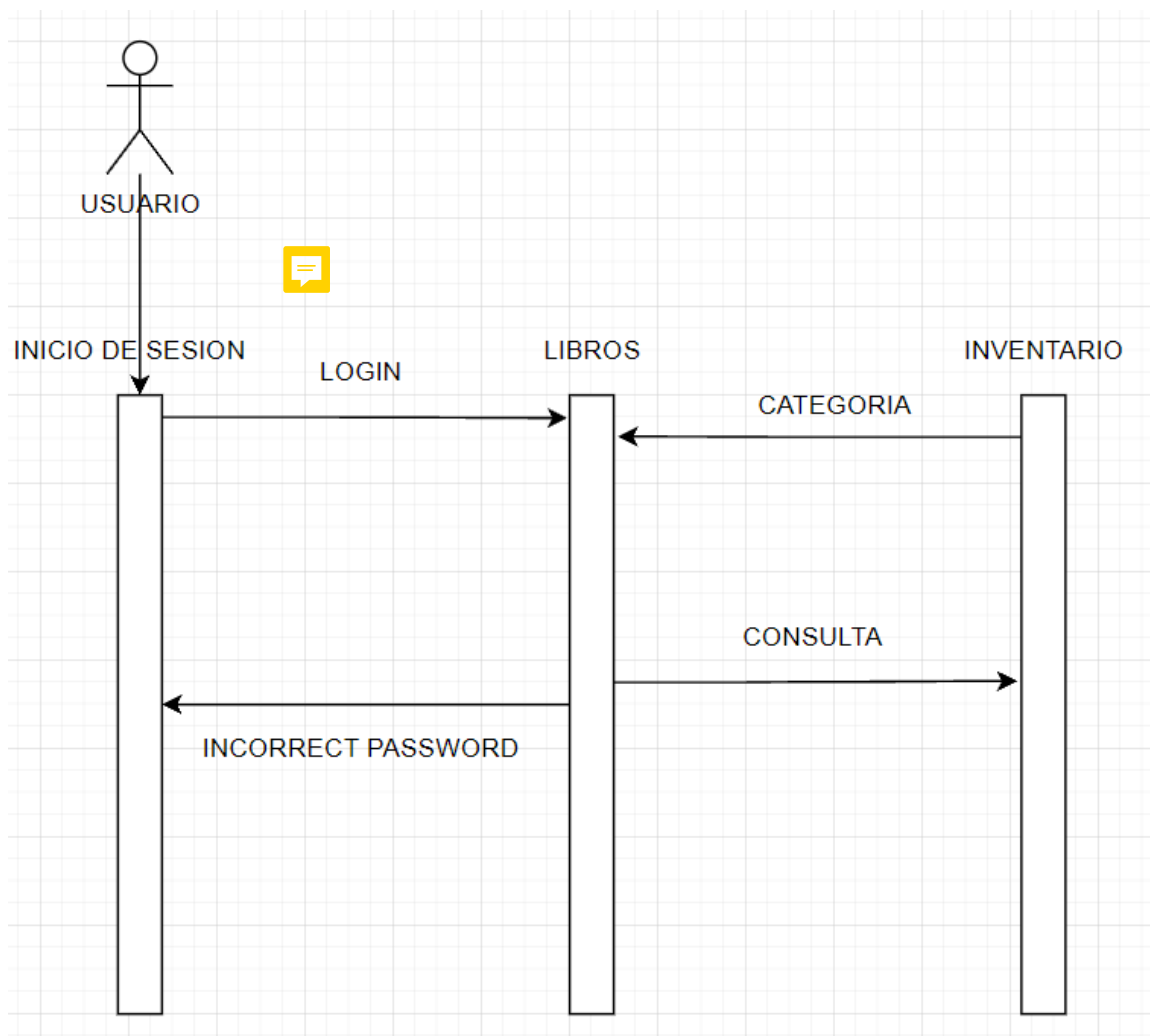
IV. Diagrama de clases

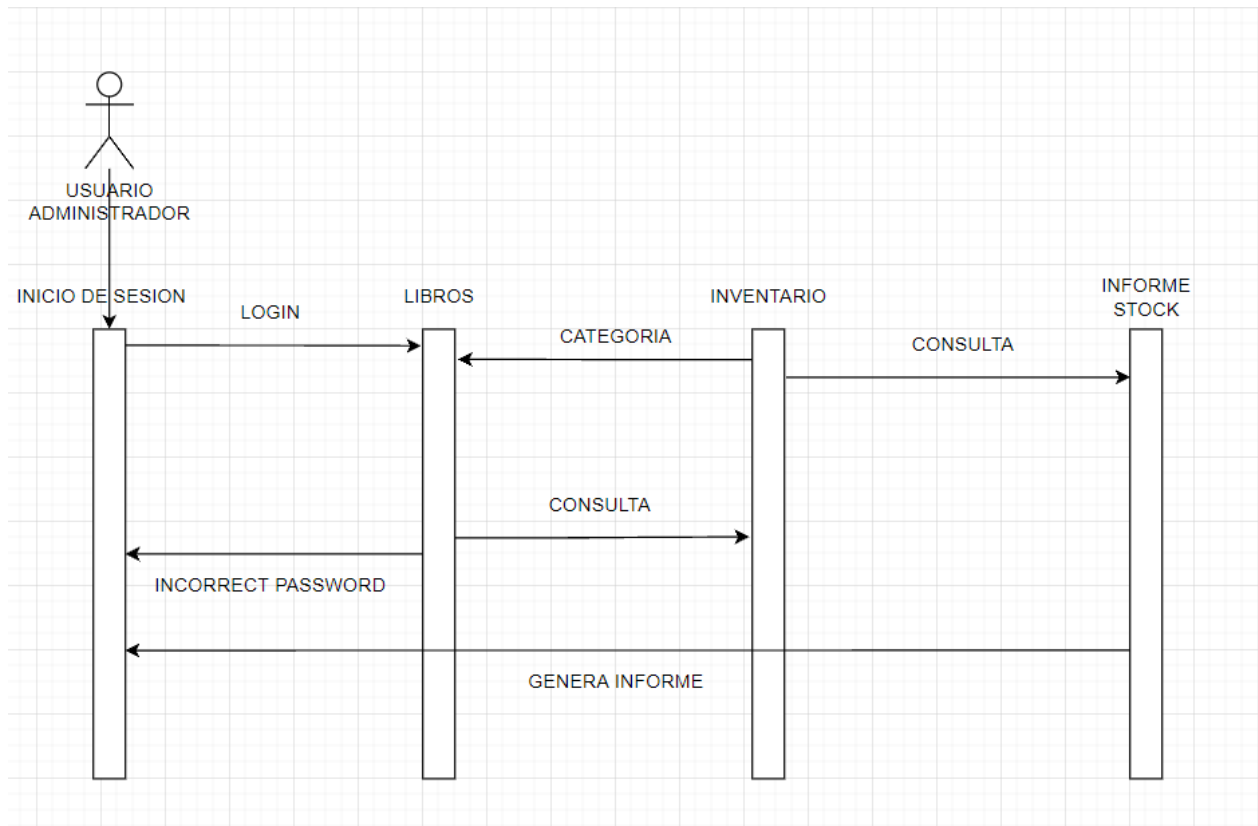
El diagrama de clases es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre sus objetos.



V. Diagrama de Secuencia

Un diagrama de secuencia nos permite observar la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada colaboración (en su aspecto mínimo para caso de uso). A menudo es útil para complementar a un diagrama de clases, pues el diagrama de secuencia se podría describir de manera informal como "el diagrama de clases en movimiento", por lo que ambos deben estar relacionados entre sí (mismas clases, métodos, atributos...). Mientras que el diagrama de casos de uso permite el modelado de una vista negocio del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

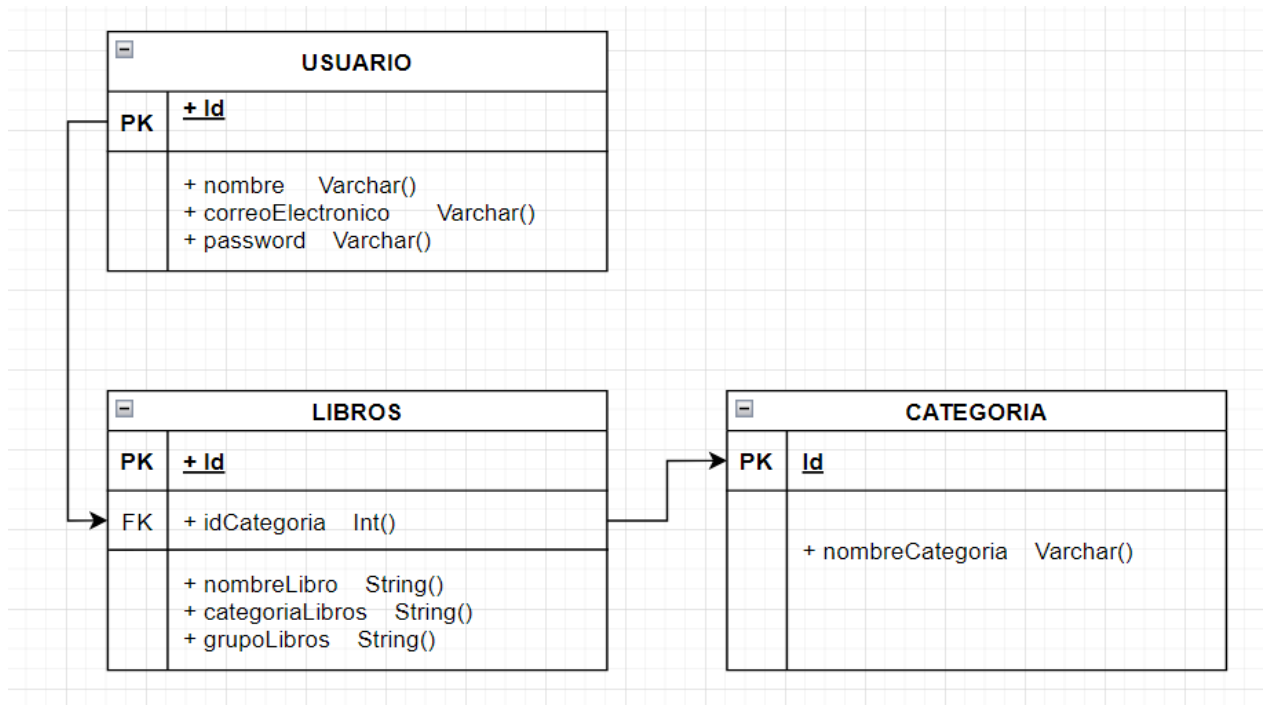




VI. Modelo de Datos

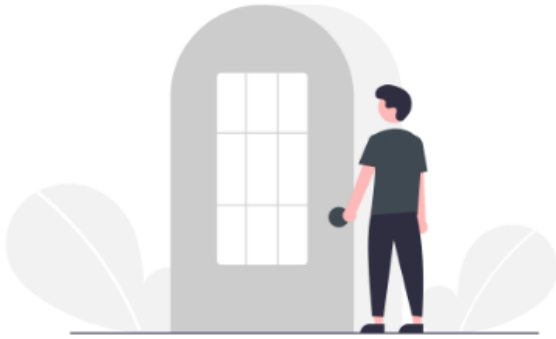
Un modelo de datos les permite indicar la forma en que serán almacenados los datos en el sistema los que posteriormente podrán ser procesados para la obtención de distintas dimensiones de información. En este nivel solo se les solicita el modelo de datos lógico y físico que cubran los requerimientos de información de sus stakeholder's.

Para la confección del modelo deberá utilizar aplicaciones por usted conocidas como WorkBench de MySQL o SQL DataModeler de Oracle.



VII. Desarrollo del Prototipo


Iniciar Sesión



Email

Contraseña

☐ Recuerdame [¿Olvidaste La Contraseña?](#)

 Iniciar Con Google

Iniciar Sesión

FILTRAR PRODUCTOS

ID: Todos


Nombre: Todos






Grupo: Todos

Categoría: Todos

☐ Asendente ☒ Desendente

Limpiar Filtros Aplicar

Dashboard **Productos** Categorías [+ Añadir Producto](#)  [Perfil](#)

	Nombre	Id	Grupo	Categoría	Stock	Vendidos	Modificar	Generar Informe
<input type="checkbox"/>	 Aaaaaaaa	1	Grupo1	Comedia	45	5 Vendido!	<button>Añadir ejemplar</button>	<button>Eliminar Unidad</button>
<input type="checkbox"/>	 Bbbbbbbb	2	Grupo3	Terror	45	5 Vendido!	<button>Añadir ejemplar</button>	<button>Eliminar Unidad</button>
<input type="checkbox"/>	 Cccccc	3	Grupo2	Suspenso	45	5 Vendido!	<button>Añadir ejemplar</button>	<button>Eliminar Unidad</button>
<input type="checkbox"/>	 Dddddd	4	Grupo1	Aventura	45	5 Vendido!	<button>Añadir ejemplar</button>	<button>Eliminar Unidad</button>
<input type="checkbox"/>	 Eeeeeee	5	Grupo5	Romanse	45	5 Vendido!	<button>Añadir ejemplar</button>	<button>Eliminar Unidad</button>

FILTRAR PRODUCTOS

buscar en Inventario

ID

Todos

Nombre

Todos

Grupo

Todos

Categoría


Todos

☐ Asendente

☒ Desendente

Limpiar Filtros

Aplicar





Dashboard

Productos

Categorías

+ Añadir Producto



Perfil 

Añadir Libro

Nombre

Grupo

Categoría

Cantidad

CANCELAR

APLICAR

VISTA PREVIA

Portada / Caratula

SUBIR

FILTRAR PRODUCTOS

buscar en Inventario

ID

Todos

Nombre

Todos

Grupo

Todos

Categoría


Todos

☐ Asendente

☒ Desendente

Limpiar Filtros

Aplicar





Dashboard

Productos

Categorías

+ Añadir Producto



Perfil 

Modificar Libro

Nombre

Grupo

Categoría

Cantidad

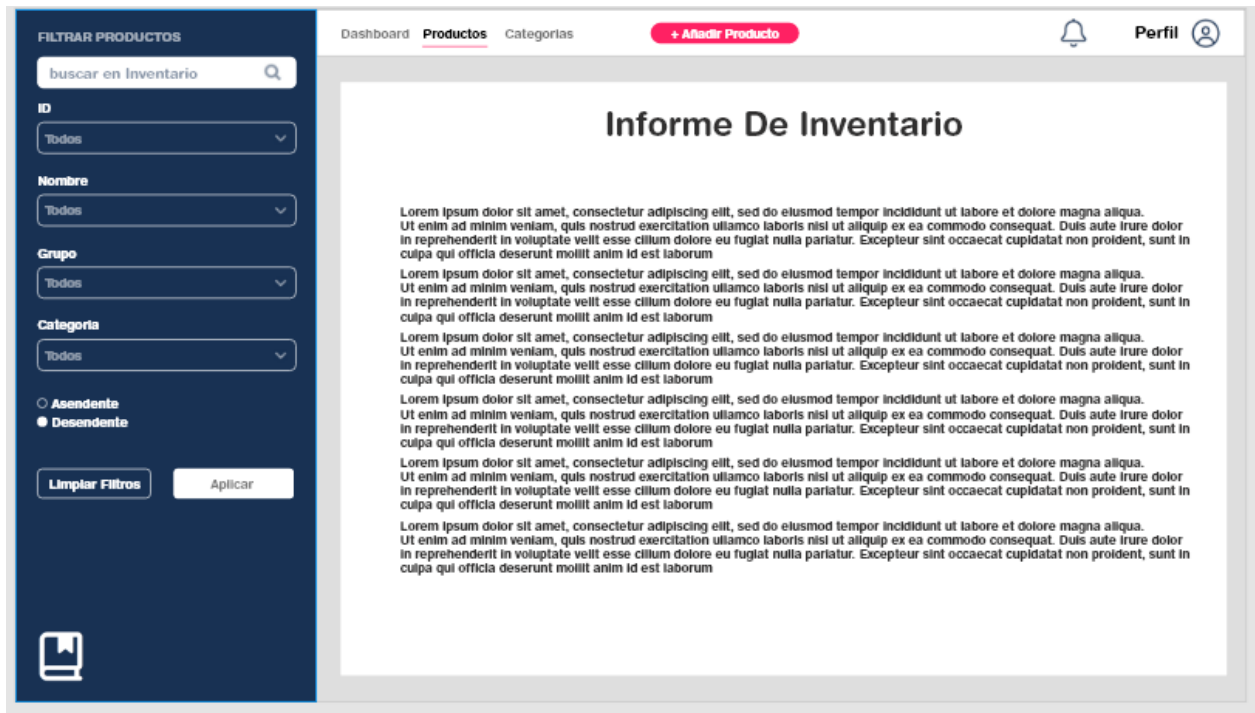
CANCELAR

APLICAR

VISTA PREVIA

Portada / Caratula

SUBIR



VIII. Conclusión

Se Puede concluir que la creación y el armado de diagramas de clases, diagramas de datos, entre otros diagramas serán y son de ayuda para entender o dar a explicar mejor el sistema que crearemos.

No obstante, también cabe aclarar que la realización de estos es bastante **engorrosa** y tardada pero aun así es de bastante ayuda al momento de organizar el sistema.

IX. Referencias bibliográficas

Diagramas hecho en : <https://bit.ly/3NGysN1> - <https://bit.ly/3u18bS7>

Bennet S. (2006). *Análisis y diseño orientado a objetos de sistemas usando UML*.

McGraw Hill

Gutierrez C. (2011). *Casos prácticos de UML*. Complutense.

Pressman R. (2010). *Ingeniería de Software: Un enfoque práctico*. McGraw Hill

Casas R. (2014). *Diseño conceptual de bases de datos en UML*. UOC

