



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN NRC: 10522
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN
DESARROLLO DE APLICACIONES WEB

GRUPO 2

INTEGRANTES

Acosta Moreno Camilo Alejandro

Almeida Logroño Nicolás André

Benavides Mina Luis Mauricio

Díaz Peñafiel Angie Elizabeth

Moreno Cardozo Paola Natalia

Moreno Cardozo Wendy Piedad

Carlos Andres Pillajo Bolagay

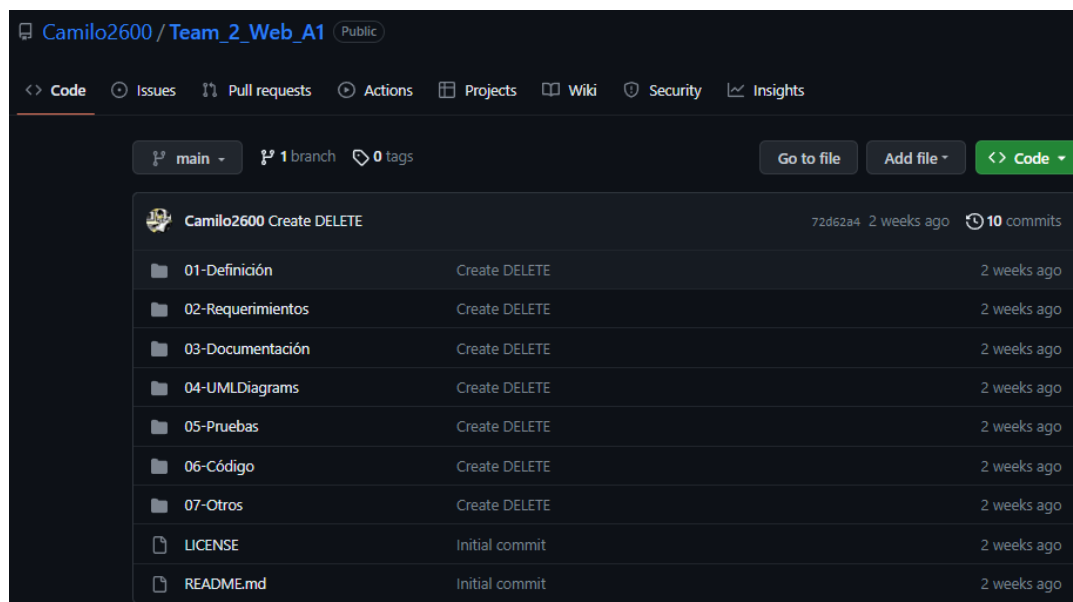
Fecha: 26 de junio de 2023

Implementación de un aplicativo web utilizando la arquitectura Cliente Servidor para la Librería en línea “Mundos de Papel”	3
Estructura del repositorio:	3
Introducción	3
Marco teorico	4
MongoDB Atlas	4
JavaScript	4
Objetivo General	4
Entrevista al Cliente	5
Pregunta 1	5
Pregunta 2	5
Pregunta 3	5
Pregunta 4	5
Pregunta 5	6
Pregunta 6	6
Pregunta 7	6
Pregunta 8	6
Marco de Trabajo	7
Historias de Usuario (Requerimientos)	8
Diagrama de caso de uso	9
Diagrama de clase	11
Diseño de arquitectura	13
Diseño de interfaz UI	14
Link del Video de la Funcionalidad de la Aplicación	15
Conclusiones	15
Recomendación	16
Referencias	17

Implementación de un aplicativo web utilizando la arquitectura Cliente Servidor para la Librería en línea “Mundos de Papel”.

Estructura del repositorio:

https://github.com/Camilo2600/Team_2_Web_A1



Introducción

La librería de Fernando Rocha es un lugar lleno de conocimiento y aventuras literarias que busca proporcionar a sus clientes una experiencia enriquecedora en el mundo de la lectura. Con una amplia gama de libros de distintos géneros y autores, esta librería se ha convertido en un refugio de la literatura y un punto de encuentro para aquellos que buscan sumergirse en historias fascinantes y adquirir conocimientos.

A pesar de su éxito y popularidad, la librería ha estado gestionando sus actividades utilizando métodos tradicionales, como hojas de cálculo y tablas dinámicas en Excel. Consciente de la necesidad de optimizar y agilizar sus operaciones, Fernando Rocha ha decidido embarcarse en un emocionante proyecto de desarrollo de aplicaciones web. El objetivo es implementar un

software personalizado que brinde a la librería una solución integral y eficiente para la gestión de inventario, clientes y otras actividades fundamentales.

Marco teorico

MongoDB Atlas

MongoDB Atlas es una base de datos mundial basada en la nube y completamente administrada de MongoDB que combina modelos de datos similares a JSON, indexación y búsqueda avanzadas, y escalabilidad elástica, a la vez que automatiza las tareas administrativas que llevan mucho tiempo.

JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Objetivo General

Desarrollar una aplicación web para la librería de Daniel Cardoso, utilizando la arquitectura cliente-servidor y la base de datos MongoDB Atlas, que permita gestionar de manera eficiente el inventario, clientes y demás actividades relacionadas, mejorando así la productividad y la experiencia del usuario.

Objetivos específicos

- Identificar los requisitos del usuario: Realizar un análisis exhaustivo de las necesidades y requerimientos de la librería y sus usuarios, incluyendo la gestión de inventario, clientes, generación de informes, entre otros aspectos relevantes.

- Diseñar la arquitectura cliente-servidor: Definir una estructura adecuada para la aplicación web, que permita la comunicación eficiente entre el cliente (interfaz de usuario) y el servidor (backend), garantizando un flujo de información fluido y seguro.
- Desarrollar diagramas UML y diagramas de clases: Crear diagramas UML que representen de manera visual las distintas entidades y relaciones de la aplicación, como el diagrama de casos de uso, diagrama de clases, entre otros. Estos diagramas ayudan a comprender la estructura y el flujo de la aplicación, facilitando el desarrollo y la comunicación entre el equipo de desarrollo y el cliente.

Entrevista al Cliente



Pregunta 1

¿Cuántos libros diferentes maneja actualmente en su inventario?

Respuesta: Actualmente manejamos alrededor de 5,000 libros diferentes.

Pregunta 2

¿Cuál es el rango de precios de los libros en su librería?

Respuesta: Nuestros libros tienen precios que oscilan entre \$10 y \$50.

Pregunta 3

¿Con qué frecuencia realiza compras de nuevos libros para su inventario?

Respuesta: Realizamos compras de nuevos libros aproximadamente una vez al mes.

Pregunta 4

¿Cómo le gustaría que se registre la disponibilidad de los libros en el inventario? ¿Por cantidad o por estado (disponible/no disponible)?

Respuesta: Preferiría que se registre por cantidad, es decir, la cantidad de ejemplares disponibles para cada libro.

Pregunta 5

¿Tiene algún requisito específico en cuanto a la interfaz del software de inventario?

Respuesta: Me gustaría que la interfaz sea intuitiva y fácil de usar, con una navegación sencilla para poder encontrar rápidamente los libros en el inventario.

Pregunta 6

¿Desea tener la opción de agregar imágenes de portada para cada libro en el inventario?

Respuesta: No es necesario para mí incluir imágenes de portada en el inventario.

Pregunta 7

¿Necesita alguna funcionalidad adicional, como la capacidad de realizar pedidos a proveedores desde el software de inventario?

Respuesta: En este momento no necesito esa funcionalidad, pero podría considerarla en el futuro.

Pregunta 8

¿Hay algún otro detalle importante que debemos tener en cuenta para desarrollar el software de inventario según sus necesidades?

Respuesta: No hay ningún detalle adicional en este momento. Lo más importante para mí es tener un software sencillo y eficiente para el control de inventario de mi librería.

Link:

https://github.com/Camilo2600/Team_2_Web_A1_y_A2/blob/main/07-Otros/Acta%20Reuni%C3%B3n%20con%20cliente%20V01.pdf

Marco de Trabajo

La metodología Scrum es una elección común para la gestión de historias de usuario debido a varios beneficios clave que ofrece:


- **Flexibilidad y adaptabilidad:** Scrum se basa en iteraciones cortas y regulares llamadas Sprints. Esto permite una mayor flexibilidad para adaptarse a los cambios en los requisitos y requisitos a medida que se desarrollen las historias de usuario. El equipo puede realizar ajustes rápidos y tomar decisiones basadas en la retroalimentación continua.
- **Enfoque en el valor del negocio:** Scrum se centra en maximizar el valor entregado al cliente. Las historias de usuario se priorizan según su valor y se entregan en incrementos funcionales, lo que permite obtener retroalimentación temprana y garantizar que las características más valiosas se implementen primero.
- **Colaboración y autogestión:** Scrum promueve la colaboración estrecha entre el equipo de desarrollo, el Product Owner y el Scrum Master. El equipo se autoorganiza y trabaja en estrecha colaboración para definir, estimar y desarrollar las historias de usuario. Esto fomenta la responsabilidad compartida y la toma de decisiones conjuntas.


Matriz de Marco de Trabajo


ITEM	PROBLEMA	QUE (NECESIDAD)	PARA QUE (SOLUCIÓN)	PARA QUIEN (USUARIO)	COMO (DESCRIPCIÓN DE TAREAS)	HECHO POR (PROG. RESP.)	CUANTO TIEMPO (ESTIMADO EN HRS)	FECHA DE ENTREGA	PRIORIDAD	STATUS	PRUEBA (COMO SE VERIFICA)	COMENTARIOS	NOMBRE DE HISTORIA
REQ001	Ingresar al sistema	Acceder a la aplicación	Para acceder a lista de libros	Administrador / Personal	Ingreso y validación: Usuario (nombre y apellido) y contraseña	Camilo Acosta, Mauricio Benavides	10	En proceso	Alta	En proceso	Mostrar al Administrador/Personal: Sus datos son correctos/ Sus datos son incorrectos, vuelva a intentar	Se debe validar que el usuario y la contraseña sean validos.	Login
REQ002	Permitir registrar nuevo usuario	Registrar en el sistema un nuevo usuario	Almacenar un nuevo usuario en el sistema.	Administrador / Personal	Registrar su usuario en la aplicación con su nombre y apellido, y crear una contraseña nueva	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Si el usuario deja un campo vacío se visualizará un mensaje de alerta indicando "Datos Incompletos".	Se debe validar que el usuario llene todos los campos	Registrarse
REQ003	Permitir añadir productos de la librería	Gestionar el ingreso de productos	Controlar el ingreso de los productos.	Administrador / Personal	Añadir la información de todos los campos del producto y guardamos en la base de datos.	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Verificar la información ingresada, en el caso que algún campo se quede vacío o se haya llenado incorrectamente presentará una alerta indicando "Verificar la información ingresada"	Se debe validar que los datos ingresados sean correctos	Añadir Productos
REQ004	Permitir eliminar productos de la librería	Eliminar un producto	Eliminar un producto en la base de datos.	Administrador / Personal	Elegimos el producto que queremos eliminar. Confirmamos que la información del producto que vamos a eliminar sea correcta. Eliminar en la base de datos.	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Verificar la información que se va a eliminar, se presentará una alerta indicando "Está seguro que quiere eliminar el producto"	Se debe validar que los datos eliminados sean correctos	Eliminar Productos
REQ005	Permitir actualizar productos de la librería	Actualizar un producto	Actualizar un producto en la base de datos.	Administrador / Personal	Seleccionar el producto y actualizamos la información de los campos del producto y guardamos en la base de datos. Actualizar la base de datos.	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Verificar la información que se va a actualizar, se presentará una alerta indicando "Está seguro que quiere actualizar el producto"	Se debe validar que los datos actualizados sean correctos	Actualizar Productos
REQ006	Permitir visualizar la disponibilidad de un producto	Gestionar la disponibilidad de un producto	Mostrar la disponibilidad de cada producto	Administrador	Escogemos el producto y marcamos disponible o no disponible para visualizar el estado del libro	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Verificar la información de estado del producto, se presentará una alerta indicando "Está seguro que quiere modificar el estado del producto"	Se debe validar que los datos actualizados sean correctos	Disponibilidad de Productos

Link: https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/02-Requerimientos

Historias de Usuario (Requerimientos)

HISTORIA DE USUARIO (HU)			
ITEM REQ003	USUARIO Administrador/ Personal	STATUS En proceso	
TIEMPO 5	PRIORIDAD Alta	PROG. RESP. Camilo Acosta, Mauricio Benavides	
QUE Gestionar el ingreso de productos	PARA QUE Controlar el ingreso de los productos.	COMO Añadir la información de todos los campos del producto y guardamos en la base de datos.	
NOMBRE HISTORIA Añadir Productos			
PRUEBA Se quede vacío o se haya llenado incorrectamente presione		COMENTARIOS Se debe validar que los datos ingresados sean correctos	

HISTORIA DE USUARIO (HU)			
ITEM REQ004	USUARIO Administrador/ Personal	STATUS En proceso	
TIEMPO 5	PRIORIDAD Alta	PROG. RESP. Camilo Acosta, Mauricio Benavides	
QUE Eliminar un producto	PARA QUE Eliminar un producto en la base de datos.	COMO Elegimos el producto que queremos eliminar. Confirmamos que la información del producto que vamos a eliminar sea correcta. Eliminar en la base de datos.	
NOMBRE HISTORIA Eliminar Productos			
PRUEBA Se eliminar, se presentará una alerta indicando "Está seguro que quiere eliminar el producto"		COMENTARIOS Se debe validar que los datos eliminados sean correctos	

HISTORIA DE USUARIO (HU)			
ITEM REQ005	USUARIO Administrador/ Personal	STATUS En proceso	
TIEMPO 5	PRIORIDAD Alta	PROG. RESP. Camilo Acosta, Mauricio Benavides	
QUE Actualizar un producto	PARA QUE Actualizar un producto en la base de datos.	COMO Seleccionar el producto y actualizamos la información de los campos del producto y guardamos en la base de datos. Actualizar la base de datos.	
NOMBRE HISTORIA Actualizar Productos			
PRUEBA actualizar, se presentará una alerta indicando "Está seguro que quiere actualizar el producto"		COMENTARIOS Se debe validar que los datos actualizados sean correctos	


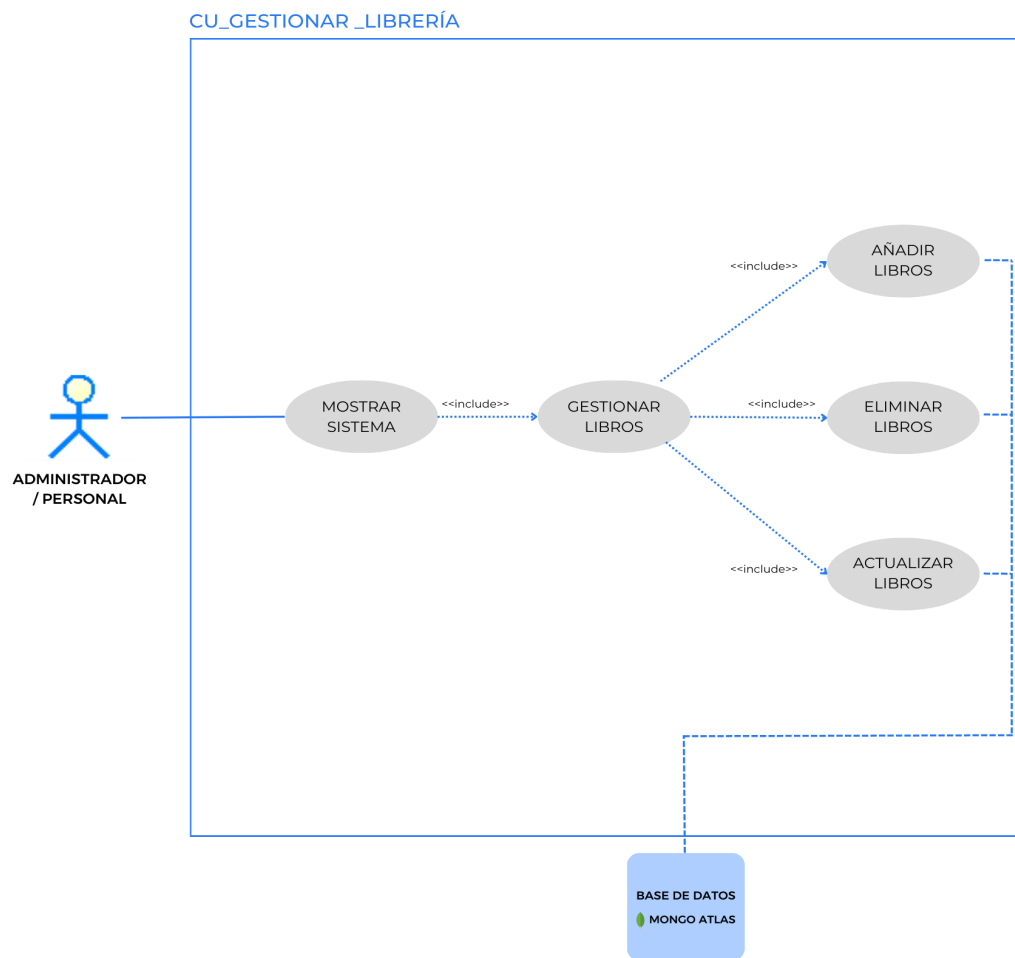
HISTORIA DE USUARIO (HU)			
ITEM REQ006	USUARIO Administrador	STATUS En proceso	
TIEMPO 5	PRIORIDAD Alta	PROG. RESP. Camilo Acosta, Mauricio Benavides	
QUE Gestionar la disponibilidad de un producto	PARA QUE Mostrar la disponibilidad de cada producto	COMO Escogemos el producto y marcamos disponible o no disponible para visualizar el estado del libro	
NOMBRE HISTORIA Disponibilidad de Productos			
PRUEBA Seleccionar el producto, se presentará una alerta indicando "Está seguro que quiere modificar el estado del producto"		COMENTARIOS Se debe validar que los datos actualizados sean correctos	

Diagrama de caso de uso

Un diagrama de casos de uso es una representación visual que describe la interacción entre los actores (usuarios o sistemas externos) y un sistema. Proporciona una visión general de los diferentes casos de uso o funcionalidades que un sistema ofrece a sus actores. Aquí tienes una breve descripción de un diagrama de casos de uso:

- **Actores:** Los actores son entidades externas que interactúan con el sistema. Pueden ser usuarios, otros sistemas o dispositivos.
- **Casos de Uso:** Un caso de uso representa una funcionalidad específica que el sistema ofrece a los actores. Describe una secuencia de acciones que el sistema realiza en respuesta a una solicitud del actor.
- **Relaciones:** Las relaciones en un diagrama de casos de uso indican la interacción entre los actores y los casos de uso. Estas relaciones pueden ser de inclusión, extensión o generalización, y ayudan a representar la lógica y la fluidez de la interacción.
- **Escenario:** Cada caso de uso puede tener varios escenarios que describen diferentes situaciones o flujos de eventos que pueden ocurrir. Estos escenarios capturan las acciones realizadas por el actor y las respuestas del sistema.
- **Objetivo:** El objetivo principal de un diagrama de casos de uso es proporcionar una descripción clara y visual de las funcionalidades principales del sistema y cómo interactúan con los actores.

Link: https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/04-Diagramas



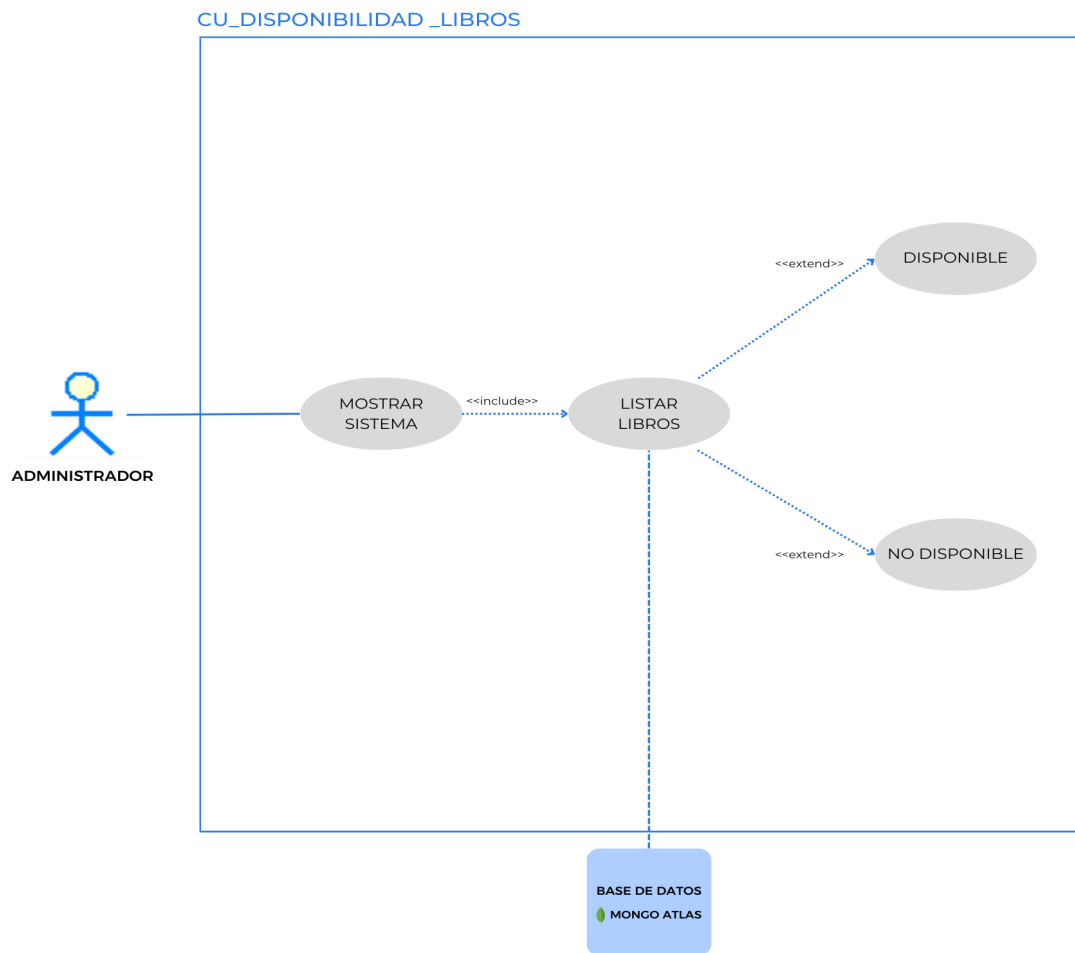


Diagrama de clase

El diagrama de clases ayuda a comprender la arquitectura y el diseño del sistema, identificando las clases clave, sus características y las relaciones que existen entre ellas.

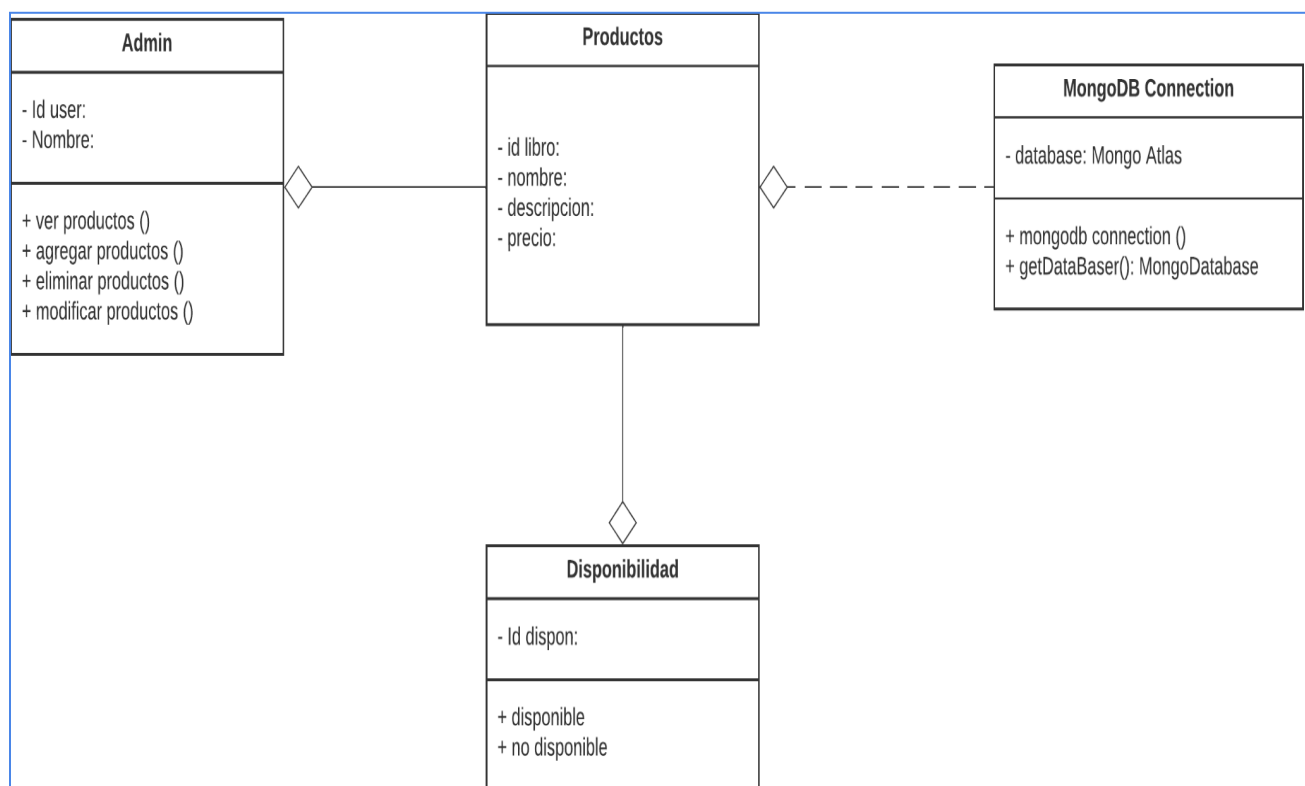
Proporciona una visión general de las entidades y sus interacciones en el sistema.

Algunos elementos importantes que se pueden encontrar en un diagrama de clases son:

- **Classes:** Representan las entidades principales del sistema y contienen atributos y métodos que definen su comportamiento.
- **Atributos:** Características o propiedades de una clase que describen su estado.
- **Métodos:** Comportamientos o acciones que una clase puede realizar.

- Relaciones: Conexiones entre las clases, como asociaciones, agregaciones, composiciones, herencias y dependencias.
- Multiplicidad: Indica que obtienen instancias de una clase están asociados con instancias de otra clase en una relación.
- Herencia: Representa la relación de especialización y generalización entre las clases, donde una clase hereda los atributos y métodos de una clase base.

En resumen, un diagrama de clases es una herramienta fundamental en la programación orientada a objetos para modelar la estructura estática de un sistema y comprender las relaciones entre las clases. Ayuda a comunicar y documentar el diseño del sistema de manera visual y clara.



Link: https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/04-Diagramas

Diseño de arquitectura

En el diseño de la arquitectura que menciona, el frontend se desarrolla utilizando Handlebars. Handlebars es un motor de plantillas que te permite generar HTML dinámico en el lado del cliente. Proporciona una sintaxis sencilla y expresiva para crear plantillas reutilizables y renderizar datos en el frontend.

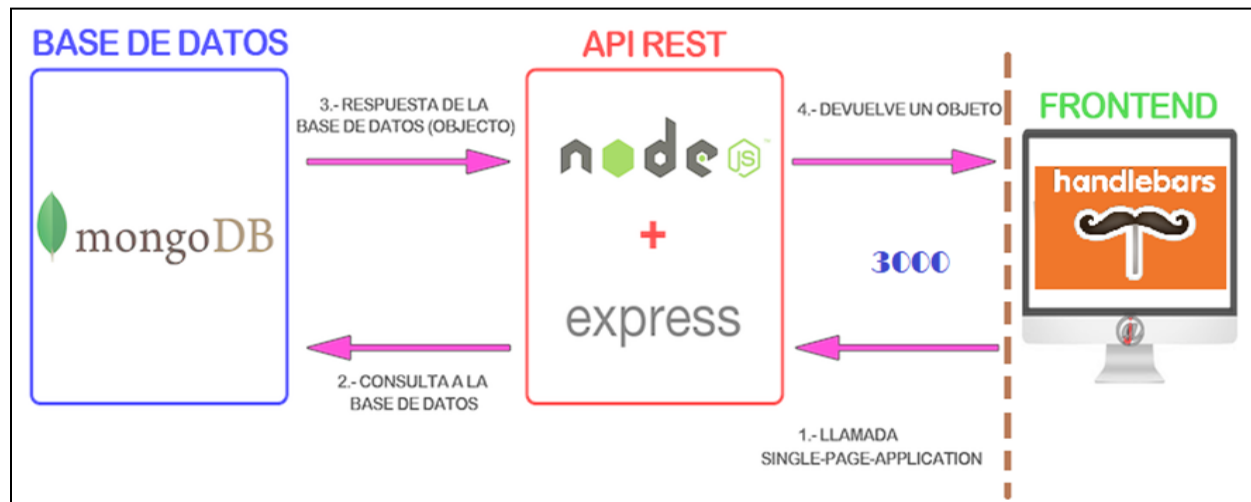
La arquitectura sigue el patrón Cliente-Servidor, donde el frontend se encarga de presentar la interfaz de usuario y enviar solicitudes al servidor a través de la API REST. El servidor, implementado con Node.js y Express, procesa las solicitudes entrantes, realiza las operaciones correspondientes en la base de datos de MongoDB Atlas y devuelve los datos necesarios al frontend.

Este diseño ofrece varias ventajas. Por un lado, MongoDB Atlas proporciona una base de datos flexible y escalable en la nube, lo que permite almacenar y gestionar eficientemente los datos de la aplicación. Por otro lado, el uso de una API REST con Node.js y Express facilita la creación de servicios web eficientes y de alto rendimiento.

El frontend desarrollado con Handlebars permite generar dinámicamente las vistas en función de los datos recibidos de la API REST, lo que brinda una experiencia de usuario más interactiva y personalizada.

En resumen, este diseño de arquitectura combina MongoDB Atlas como base de datos en la nube, una API REST implementada con Node.js y Express para el backend, y Handlebars como motor de plantillas en el frontend, brindando una solución completa y escalable para desarrollar aplicaciones web.

Link: https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/08%20Arquitectura

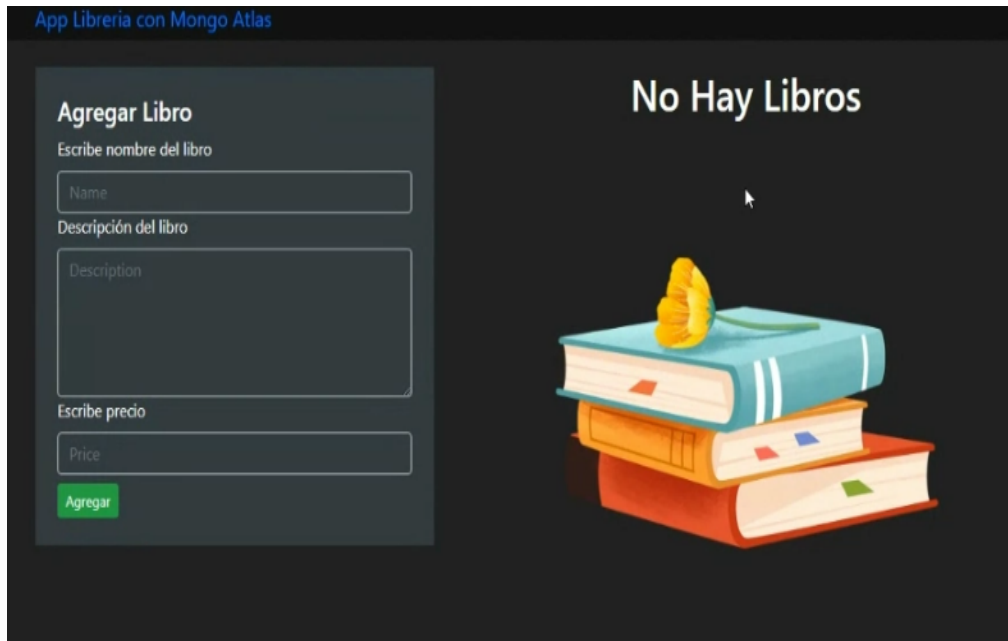


Diseño de interfaz UI

Al interactuar con la interfaz de usuario, los usuarios pueden realizar diversas acciones, como ingresar un nuevo libro requerirá su nombre, descripción, precio y estado de disponibilidad. Estos datos se envían a través de solicitudes HTTP al backend, donde se validan y procesan. Si la validación es exitosa, el backend utiliza los modelos de Mongoose para crear un nuevo documento en la colección de libros de la base de datos.

Del mismo modo, para modificar, eliminar o actualizar un libro, se envían solicitudes HTTP al backend con los datos correspondientes. El backend realiza las operaciones adecuadas en la base de datos utilizando los métodos adecuados por Mongoose.

En resumen, el diseño de la interfaz de usuario para la aplicación de la librería implica la implementación de formularios, vistas y acciones que permiten a los usuarios interactuar con la aplicación y realizar operaciones de ingreso, modificación, listado, eliminación y actualización de libros. Estas acciones se realizan a través de solicitudes HTTP entre el frontend (implementado con Handlebars) y el backend (implementado con Express y Mongoose), que se conecta a la base de datos MongoDB Atlas.



Link del Video de la Funcionalidad de la Aplicación

<https://youtu.be/2YTja1uhRvk>

Conclusiones

En conclusión, la creación de la aplicación de inventario para la librería "Mundo de Papel" es un proyecto viable y mejora para mejorar la gestión y control de los libros en la librería. Al implementar una interfaz de usuario intuitiva y funcional, se brinda a los usuarios la capacidad de realizar diversas operaciones, como ingresar nuevos libros, modificar su información, listar los libros disponibles, eliminar registros y actualizar detalles importantes como el nombre, descripción, precio y disponibilidad de los libros.

Al adoptar la metodología Scrum, se fomenta una gestión ágil del proyecto, permitiendo una mayor flexibilidad y adaptabilidad a medida que se desarrolla la

aplicación. El enfoque de desarrollo basado en historias de usuario y la planificación de sprints facilitan la entrega iterativa de funcionalidades y la retroalimentación continua del equipo y los usuarios.

La elección de tecnologías como MongoDB Atlas para la base de datos, Node.js con Express para la creación de la API REST y Handlebars para el frontend ofrece un conjunto sólido de herramientas para el desarrollo de la aplicación. Estas tecnologías son populares, muy utilizadas y cuentan con una buena documentación y soporte de la comunidad, lo que facilita el proceso de desarrollo y solución de problemas.

En cuanto a la interfaz de usuario, es importante diseñar una interfaz limpia, intuitiva y fácil de usar. Se deben considerar aspectos como la navegación clara y consistente, la presentación ordenada de la información y la inclusión de elementos visuales y controles que faciliten la interacción del usuario. Además, se debe prestar especial atención a la validación de los datos ingresados por el usuario para garantizar la integridad de la información.

Recomendación

Como recomendación, se sugiere realizar pruebas exhaustivas tanto en el frontend como en el backend para garantizar un correcto funcionamiento y evitar posibles errores o inconsistencias en los datos. Además, es importante considerar la seguridad de la aplicación, implementando como la autenticación y autorización de usuarios, así como la protección contra posibles ataques, como la inyección de código o la exposición de información sensible.

En resumen, la creación de la aplicación de inventario para la librería "Mundo de Papel" utilizando una interfaz de usuario bien diseñada, la metodología Scrum y las tecnologías adecuadas, puede brindar una solución eficiente y efectiva para mejorar la gestión y control de los libros en la librería, ofreciendo una experiencia positiva tanto para los usuarios como para el personal encargado de la administración de los libros.

Referencias

- Amador, D. P. (2015). Diseño e implementación de una aplicación web para una tienda virtual. <https://repositorio.upct.es/bitstream/10317/5199/1/tfg735.pdf>
- Candela, J., & Espinoza, J. (2021). Propuesta de un sistema de información sobre arquitectura Cloud para gestionar las ventas presenciales para una empresa comercial. <https://doi.org/10.19083/tesis/654013>
- Ramírez, A. P. G. (2022). Validación y verificación de requisitos de software en proyectos basados en desarrollo continuo usando técnicas de PLN. In Encuentro Internacional de Educación en Ingeniería. <https://doi.org/10.26507/paper.2657>
- Izaurrealde, M. P. (2013). Caracterización de Especificación de Requerimientos en entornos Ágiles: Historias de Usuario. Trabajo de especialidad, Febrero. https://www.institucional.frc.utn.edu.ar/sistemas/lidicalso/pub/file/Tesis/Anteproyecto_Requerimientos_en_Metodolog%C3%ADas_Agiles.pdf
- Otiniano, J., & Blas, L. P. (2013). Influencia de los servicios web en la reutilización de código en el desarrollo de software. In Universidad Nacional de Trujillo. <http://dspace.unitru.edu.pe/handle/UNITRU/8559>

- Da Silva, D. (2022b, August 5). Manejo de conflictos: estrategias prácticas de resolución. Zendesk MX.
<https://www.zendesk.com.mx/blog/manejo-de-conflictos-estrategias-practicas-de-resolucion/>
- Martínez, E. S. (2014). Propuesta de procedimiento para realizar pruebas de caja blanca a las aplicaciones que se desarrollan en lenguaje Python. 3 c TIC: cuadernos de desarrollo aplicados a las TIC, 3(2), 89-114.
- Gomez, C. (2023b, January 19). Técnicas de prueba de caja negra. Diario de QA.
<https://www.diariodeqa.com/post/tecnicas-de-prueba-de-caja-negra>
- López, D., & Maya, E. (2017). Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web.
- Vera, A., Zúñiga, N., & Bernal, Ñ. (1969). Herramienta en línea para la programación y depuración remota de funciones lógicas digitales. Ingeniería Y Competitividad, 15(1), 79–91.
<https://doi.org/10.25100/iyc.v15i1.2622>