



**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: 27 de junio de 2023**

<b>Implementación de Servicios Web en la aplicación “Mundos de Papel” .....</b>	<b>3</b>
<b>Estructura del repositorio:.....</b>	<b>3</b>
<b>Introducción.....</b>	<b>3</b>
<b>Marco teorico.....</b>	<b>4</b>
MongoDB Atlas.....	4
JavaScript.....	4
Node.js.....	4
Google Cloud.....	5
Postman.....	5
<b>Objetivo General.....</b>	<b>5</b>
<b>Entrevista al Cliente.....</b>	<b>6</b>
Pregunta 1.....	6
Pregunta 2.....	6
Pregunta 3.....	6
Pregunta 4.....	6
Pregunta 5.....	7
Pregunta 6.....	7
Pregunta 7.....	7
Pregunta 8.....	7
<b>Marco de Trabajo.....</b>	<b>8</b>
<b>Historias de Usuario (Requerimientos).....</b>	<b>8</b>
<b>Diagrama de caso de uso.....</b>	<b>9</b>
<b>Diagrama de clase.....</b>	<b>10</b>
<b>Diseño de arquitectura.....</b>	<b>10</b>
<b>Link del Video de la Funcionalidad de la Aplicación.....</b>	<b>11</b>
<b>Conclusiones.....</b>	<b>11</b>
<b>Recomendaciones.....</b>	<b>12</b>
<b>Referencias.....</b>	<b>13</b>

## Implementación de Servicios Web en la aplicación “Mundos de Papel”.

### Estructura del repositorio:

[https://github.com/Camilo2600/Team\\_2\\_Web\\_A1](https://github.com/Camilo2600/Team_2_Web_A1)



### Introducción

La aplicación de inventario para la librería "Mundo de Papel" se encuentra en desarrollo con el objetivo de optimizar la gestión y consulta de productos, precios y análisis financiero. En este proceso, se ha llevado a cabo la implementación de 9 servicios API, aprovechando herramientas como MongoDB Atlas, JavaScript, Node.js, Google Cloud y Postman. Estas API están diseñadas para mejorar la funcionalidad y eficiencia de la aplicación, permitiendo consultas más rápidas y precisas, así como un análisis financiero más completo.

Las API (Application Programming Interfaces) son interfaces de software que permiten a diferentes aplicaciones comunicarse y compartir datos entre sí de manera estandarizada. Sirven como puentes de conexión que facilitan la interacción entre diferentes sistemas y componentes de software. Las API permiten a los desarrolladores aprovechar la funcionalidad existente de otros sistemas, integrar servicios externos, crear aplicaciones más rápidamente y fomentar la colaboración entre plataformas.

### **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.

#### **Node.js**

Node.js es un entorno de ejecución de JavaScript basado en el motor de JavaScript V8 de Chrome. A diferencia del JavaScript tradicional que se ejecuta en el navegador, Node.js permite ejecutar código JavaScript en el servidor. Node.js es conocido por su enfoque en la escalabilidad y la eficiencia en la gestión de aplicaciones en tiempo real y de alto rendimiento.

## **Google Cloud**

Google Cloud es una plataforma de servicios en la nube ofrecida por Google que permite a los desarrolladores implementar, escalar y administrar aplicaciones y servicios en línea. Proporciona una amplia gama de servicios, como almacenamiento, bases de datos, procesamiento en lotes, aprendizaje automático y más, para ayudar a las empresas a ejecutar sus aplicaciones y servicios de manera confiable y eficiente.

## **Postman**

Postman es una herramienta de desarrollo de API que permite a los desarrolladores probar, documentar y compartir API de manera eficiente. Actuando como un cliente de API, Postman permite enviar solicitudes HTTP a diferentes puntos finales y recibir respuestas, lo que facilita la verificación del comportamiento y el funcionamiento de las API.

## **Objetivo General**

El principal objetivo de las API desarrolladas es facilitar la gestión de productos en la librería "Mundo de Papel". Esto incluye la capacidad de agregar nuevos productos al inventario, actualizar información relevante como son descuentos, y realizar búsquedas eficientes de productos específicos.

## **Objetivos específicos**

- Implementar las API en la aplicación de la librería para permitir consultas rápidas y precisas de precios de productos. Esto garantizará que los empleados de la librería puedan proporcionar información actualizada y precisa a los clientes, mejorando la experiencia de compra.
- Aprender y desarrollar las API, se ha utilizado Google Cloud como plataforma para alojar y ejecutar las aplicaciones. Esto permite una integración fluida con otros

servicios y herramientas de Google Cloud, aprovechando su escalabilidad, seguridad y confiabilidad.

- 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](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 un enfoque ágil de gestión de proyectos que se utiliza para desarrollar productos y software de manera iterativa e incremental. En Scrum, las historias de usuario representan los requisitos y funcionalidades del producto y se recopilan en el Product Backlog, que es una lista priorizada de estas historias, en nuestro caso hemos utilizado un formato Excel.

ITEM	PROBLEMA	QUE (NECESIDAD)	PARA QUE (SOLUCIÓN)	PARA QUIEN (USUARIO)	COMO (DESCRIPCIÓN DE TAREAS)	HECHO POR (PROG. RESP.)	CUÁNTO TIEMPO (ESTIMADO EN HRS)	FECHA DE ENTREGA	PRIORIDAD	STATUS	PRUEBA (COMO SE VERIFICA)	COMENTARIOS	NOMBRE DE HISTORIA
REQ005	Permitir calcular impuesto de un producto	Calcular Impuesto	Conocer el impuesto de un producto	Administrador / Personal	Primero se calcula el precio neto del libro dividiendo el precio normal para 1.12. Una vez calculado el precio neto se multiplica por 0.12 para conseguir el valor del iva	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Mostrando en pantalla del impuesto calculado		Calcular Impuesto
REQ006	Permitir buscar libro según el nombre registrado	Buscar Libro	Agilizar la búsqueda de un libro según su nombre	Administrador / Personal	Para buscar un libro de debe escribir el nombre del libro para filtrar las opciones de la base de datos	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	la aplicación debe responder solo con el libro que se esta buscando		Buscar Libro
REQ007	Permitir calcular ganancia de un producto registrado	Calcular Ganancia	Conocer cual es la ganancia de cada producto	Administrador / Personal	en esta aplicación tenemos establecido como predeterminado la ganancia del 70%	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	se muestra la ganacia del los libros de la base de datos		Calcular Ganancia
REQ008	Permitir calcular descuento aplicado a un producto registrado	Calcular Descuento	Conocer cual es el descuento que aplica a cada producto	Administrador / Personal	El descuento predeterminado de la aplicacion es del 1% para todos los libros	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Muestra el valor de libro aplicando el descuento		Calcular Descuento
REQ009	Permitir buscar precio de un producto registrado	Buscar Precio	Buscar el producto según el precio registrado para filtrar solo con el precio asignado	Administrador / Personal	al colocar el precio de algun libro de la base de datos se filtran los resultados que conciden	Camilo Acosta, Mauricio Benavides	5	En proceso	Alta	En proceso	Aparecen los libros que se estan buscando conforme al precio		Buscar Precio

Link: [https://github.com/Camilo2600/Team\\_2\\_Web\\_A1\\_y\\_A2/tree/main/02-Requerimientos](https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/02-Requerimientos)



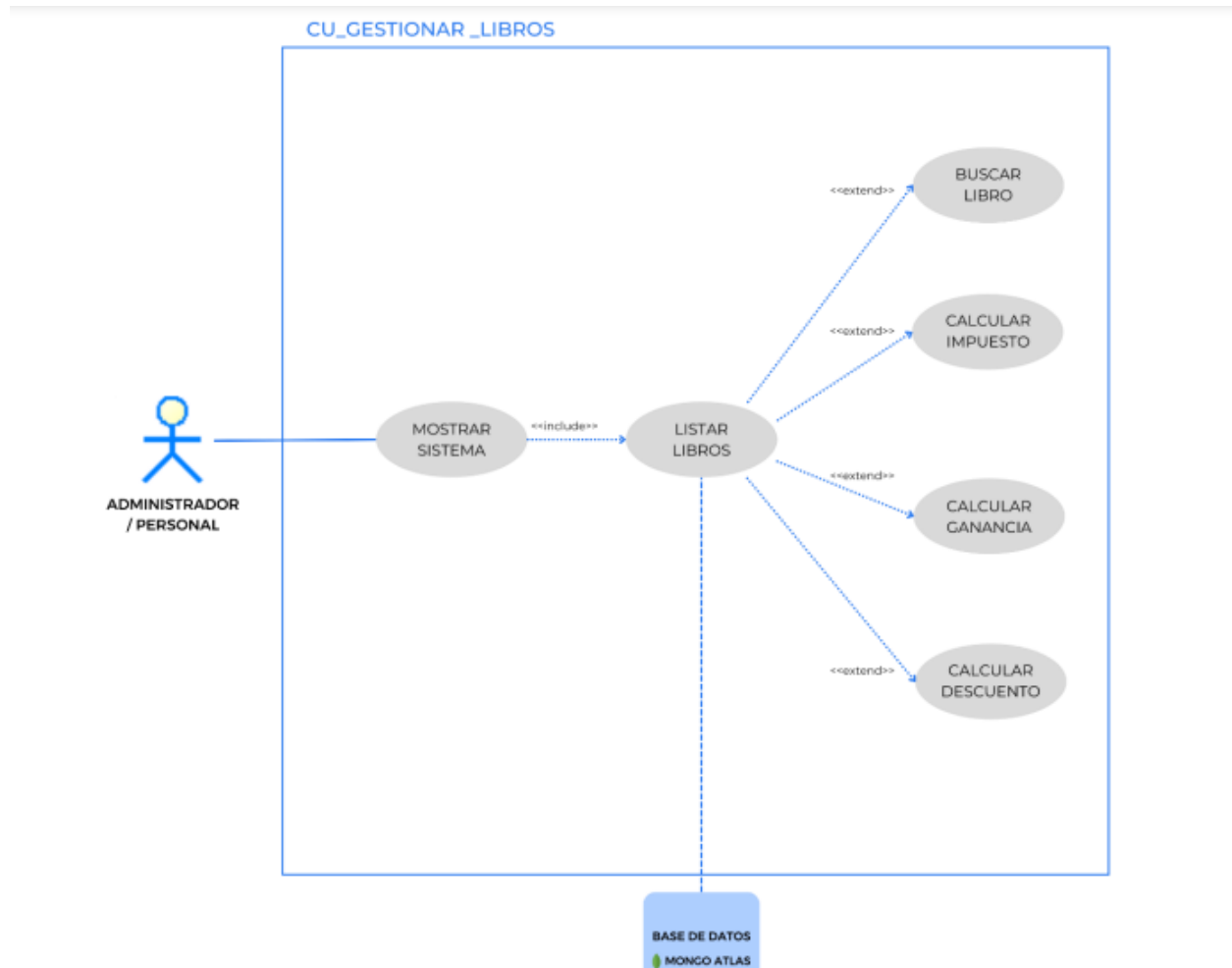
## Historias de Usuario (Requerimientos)

HISTORIA DE USUARIO (HU)			
ITEM	USUARIO	STATUS	
REQ006	Administrador/ Personal	En proceso	
TIEMPO	PRIORIDAD	PROG. RESP	
5	Alta	Camilo Acosta, Mauricio Benavides	
QUE	Buscar Libro	PARA QUE	COMO
		Agilizar la búsqueda de un libro según su nombre	Para buscar un libro de debe escribir el nombre del libro para filtrar las opciones de la base de datos
NOMBRE HISTORIA	Buscar Libro		
PRUEBA	la aplicación debe responder solo con el libro que se esta buscando		COMENTARIOS
			0

## Diagrama de caso de uso

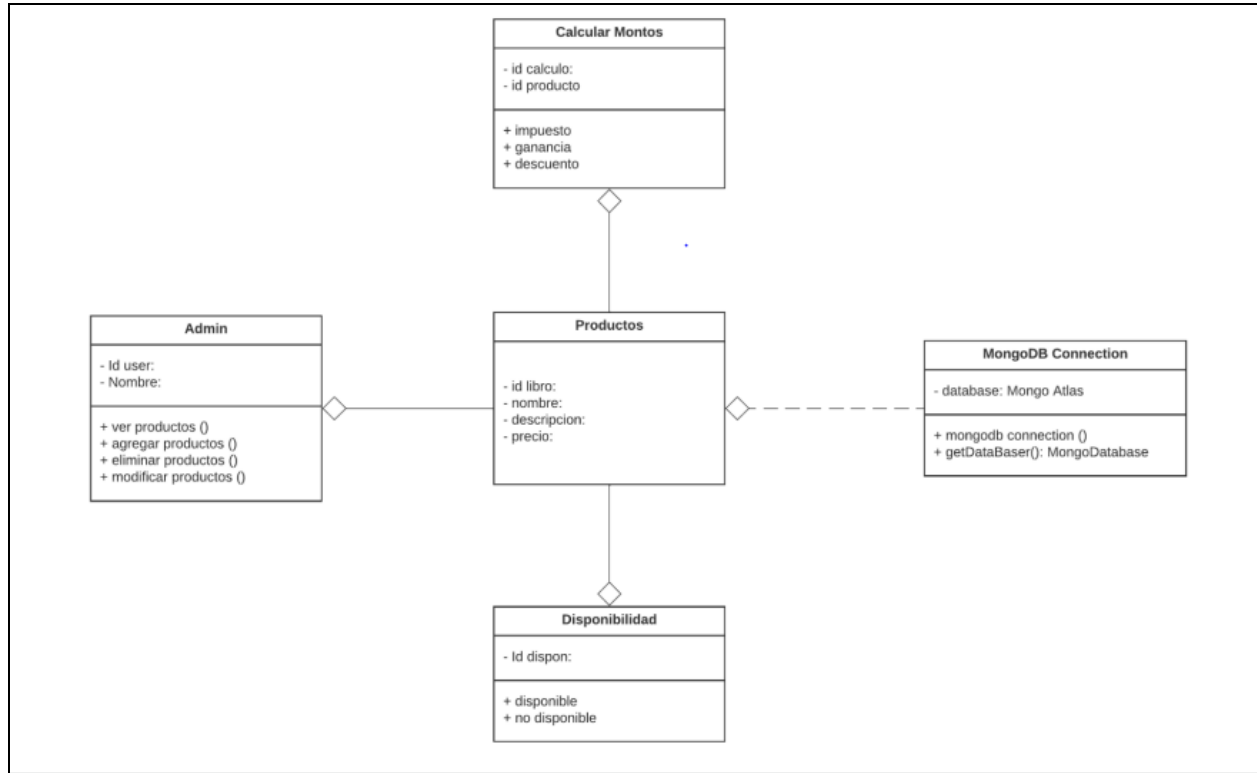
Los diagramas de caso de uso son una herramienta utilizada en ingeniería de software para visualizar y comprender las interacciones entre los actores y el sistema en desarrollo. En el contexto de una aplicación de API, los diagramas de caso de uso son útiles para representar cómo los diferentes actores interactúan con las API y cómo se utilizan para lograr ciertos objetivos.

Link: [https://github.com/Camilo2600/Team\\_2\\_Web\\_A1\\_y\\_A2/tree/main/04-Diagramas](https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/04-Diagramas)



### Diagrama de clase

Los diagramas de clase son una herramienta fundamental en la utilización de API para la aplicación del inventario de una librería. Ayudan en el modelado, el diseño, la organización del código, la comunicación y la documentación del sistema, lo que contribuye a un desarrollo eficiente y una comprensión clara de la aplicación..

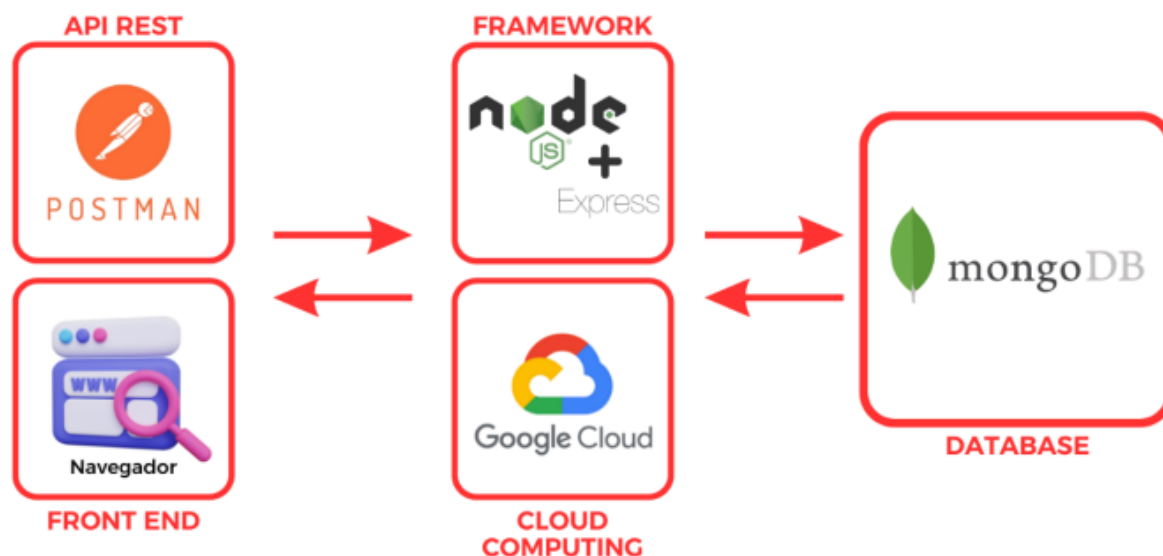


**Link:** [https://github.com/Camilo2600/Team\\_2\\_Web\\_A1\\_y\\_A2/tree/main/04-Diagramas](https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/04-Diagramas)

### Diseño de arquitectura

En la arquitectura utilizada, se combinan varias tecnologías y servicios para desarrollar y publicar las API en la nube. Postman se sacó para probar las API antes de desplegarlas, mientras que Node.js y Express permitieron construir las API y gestionar las solicitudes y respuestas. Google Cloud proporcionó una plataforma robusta para alojar las API, permitiendo su acceso y disponibilidad en línea. Por otro lado, MongoDB Atlas se encargó de almacenar y gestionar la base de datos en la nube, asegurando la persistencia y el acceso eficiente a los datos. En conjunto, esta arquitectura permitió desarrollar API eficientes y escalables, brindando una experiencia de usuario fluida al interactuar con el inventario de la librería a través de la nube.

## Modelación de Arquitectura del Sistema



**Link:** [https://github.com/Camilo2600/Team\\_2\\_Web\\_A1\\_y\\_A2/tree/main/08%20Arquitectura](https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/08%20Arquitectura)

## Link del Video de la Funcionalidad de la Aplicación

<https://youtu.be/dkNRweGdyNQ>

## Conclusiones

La implementación de las aplicaciones de API en el contexto del inventario de la librería "Mundo de Papel" ha brindado numerosos beneficios. Estas aplicaciones han mejorado la funcionalidad general del sistema al permitir la gestión eficiente de productos, actualización de impuestos, análisis de ganancias, aplicación de descuentos y búsqueda de precios. Esto ha facilitado las tareas diarias del personal de la librería y ha mejorado la experiencia del cliente al proporcionar respuestas rápidas y precisas a sus consultas relacionadas con los productos.

La adopción de tecnologías como Node.js, Express, Google Cloud y MongoDB Atlas ha sido fundamental en la implementación de estas aplicaciones de API. Node.js ha permitido la ejecución eficiente del código del servidor, mientras que Express ha simplificado la creación de rutas y el manejo de las solicitudes HTTP. El uso de Google Cloud ha proporcionado una plataforma confiable para alojar las API, lo que garantiza su disponibilidad en línea y su escalabilidad según las necesidades. Por su parte, MongoDB Atlas ha asegurado el almacenamiento y la gestión eficiente de la base de datos en la nube, brindando una solución robusta y confiable.

### **Recomendaciones**

Es importante mantener un código limpio y modular al desarrollar las aplicaciones de API. Esto facilitará su mantenimiento y permitirá futuras actualizaciones de manera más eficiente. Se recomienda seguir las mejores prácticas de programación, como el uso de comentarios claros, la organización adecuada del código en módulos y la implementación de patrones de diseño cuando sea necesario.

Además, es crucial realizar una configuración correcta del archivo `app.yaml`, que actúa como un intermediario entre la API y el servicio de Google Cloud. Es necesario asegurarse de que todas las configuraciones y variables estén correctamente especificadas para garantizar un uso exitoso de las API en la nube.

Finalmente, es recomendable realizar pruebas exhaustivas de las aplicaciones de API para verificar su funcionamiento correcto y su rendimiento. Esto puede incluir pruebas de

unidad, pruebas de integración y pruebas de carga para garantizar la estabilidad y la eficiencia del sistema en diferentes escenarios.

### 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](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>