< Universidad Tecnólogica De Bólivar>

<App sistema de gestión documental>



PROPIETARIOS DEL CONTENIDO: <Lucas Pedrozo Pertuz, Michael Andres Casadiegos Berrio, Jose Miguel Mendoza>

PROFESOR: <Edwin Alexander Puerdas Del Castillo>

Contenido

[1 Introducción. 4](#_Toc120646508)

[1.1 Resumen caso de estudio 4](#_Toc120646509)

[1.2 Objetivo 4](#_Toc120646510)

[2 Tipos de usuarios 5](#_Toc120646511)

[3 Stakeholders del sistema 6](#_Toc120646512)

[4 Responsabilidades del entorno de desarrollo 7](#_Toc120646513)

[5 Cobertura de requisitos 8](#_Toc120646514)

[5.1 Requisitos funcionales 8](#_Toc120646515)

[5.2 Requisitos no funcionales 9](#_Toc120646516)

[6 Diagramas 10](#_Toc120646517)

[6.1 Diagrama de proceso de negocio 10](#_Toc120646518)

[6.2 Diagrama de componentes: servicio de inventario 10](#_Toc120646519)

[6.3 Diagrama de secuencias: servicio de inventario 11](#_Toc120646520)

[6.4 Diagrama de componentes: servicio de carrito 11](#_Toc120646521)

[6.5 Diagrama de secuencias: servicio de carrito 12](#_Toc120646522)

[6.6 Diagrama de despliegue de la arquitectura del sistema 12](#_Toc120646523)

[7 Diagramas lógicos del sistema 13](#_Toc120646524)

[7.1 Diagrama casos de uso 13](#_Toc120646525)

[7.2 Diagrama BPMN 13](#_Toc120646526)

[7.3 Diagrama de actividades 14](#_Toc120646527)

[8 Rutas de servicios 15](#_Toc120646528)

[8.1 Ruta de servicio: Inventario 16](#_Toc120646529)

[8.2 Ruta de servicio: Inventario id 16](#_Toc120646530)

[8.3 Ruta de servicio: Create inventory 16](#_Toc120646531)

[8.4 Ruta de servicio: Update inventory 16](#_Toc120646532)

[8.5 Ruta de servicio: Delete inventory 17](#_Toc120646533)

[8.6 Ruta de servicio: Carrito 17](#_Toc120646534)

[8.7 Ruta de servicio: Carrito id 17](#_Toc120646535)

[8.8 Ruta de servicio: Update carrito 17](#_Toc120646536)

[8.9 Ruta de servicio: Login 18](#_Toc120646537)

[8.10 Ruta de servicio: Sigin 18](#_Toc120646538)

[8.11 Ruta de servicio: Crear orden 18](#_Toc120646539)

[8.12 Ruta de servicio: Feedback 19](#_Toc120646540)

[9 Puntos de vistas 20](#_Toc120646541)

[9.1 Mockup inventario 20](#_Toc120646542)

[9.2 Mockup catalogo 20](#_Toc120646543)

[9.3 Mockup carrito 21](#_Toc120646544)

[9.4 Mockup dashboard 21](#_Toc120646545)

[9.5 Mockup inicio de sesión 22](#_Toc120646546)

[9.6 Mockup registro de usuarios 22](#_Toc120646547)

[10 Descripción de la Arquitectura de hardware a utilizar en el sistema 23](#_Toc120646548)

[11 Compatibilidad 24](#_Toc120646549)

[12 Informes JIRA 25](#_Toc120646550)

[12.1 Informes de trabajo completado 25](#_Toc120646551)

[12.2 Informe de velocidad 28](#_Toc120646552)

[12.3 Diagrama de flujo acumulado 28](#_Toc120646553)

[12.4 Flow Time 29](#_Toc120646554)

[12.5 Created vs Resolved 29](#_Toc120646555)

[13 Recursos 30](#_Toc120646556)

[13.1 Enlace repositorio 30](#_Toc120646557)

[14 Agradecimientos 31](#_Toc120646558)

# Introducción.

## Resumen caso de estudio

|  |
| --- |
| Nuestra aplicación del sistema de gestión documental, llamada “Docs.”, es realizada con un enfoque en la arquitectura de servicios. Esta aplicación permitirá a los usuarios interactuar con los diferentes documentos que se ofrecen en ella, teniendo la posibilidad de comprar los documentos que están a la venta, como lo son libros.  Implementamos diferentes vistas que nos brindará una mejor experiencia, como son: iniciar sesión para el registro de nuestros usuarios, catálogo de nuestros documentos y a continuación también tenemos algunos servicios esenciales como son: Servicio stock o inventario y el servicio de carrito.  Al momento que los usuarios hagan su elección de los documentos, el carrito recoge toda la información de ellos para así terminar el proceso de compra y el inventario decremente. Al momento de realizar estos servicios finalizaremos el sistema de una manera exitosa.  Esta aplicación su código fue documentado en relación con html, css y JavaScript, para el inventario de todos nuestros documentos, donde podemos almacenar los datos, relacionados y estructurados, para poder consultarlos rápidamente, necesitamos de una base de datos, que está trabaja directamente con MySQL y es desplegada mediante el servicio en la nube llamado Heroku. |

## Objetivo

|  |
| --- |
| El objetivo de la aplicación del sistema de gestión documental es brindar a todos nuestros usuarios diferentes documentos, como puedes ser libros, artículos, revistas, audios, podcast, entre otros más. Con el fin, de que ellos puedan comprar cualquiera de esos documentos que necesite. |

# Tipos de usuarios

|  |
| --- |
| En nuestro sitio web encontramos dos tipos de roles:   * Administrador: encargado de estar al tanto del correcto funcionamiento y del contenido de la página web. * Usuario: Rol de las personas registradas de nuestro sitio web para realizar el proceso de compra de los documentos. |

# Stakeholders del sistema

|  |
| --- |
| Stakeholder internos:   * Administrador: encargado de estar al tanto del correcto funcionamiento y del contenido de la página web. * Empleado: Prestación de servicios. * Programador: Se aseguran de que la información del sitio web es correcta, segura y está actualizad   Stakeholder externos:   * Proveedores: Encargados de suministrar a la página web, los distintos documentos que serán vendidos y que incrementarán el inventario. * Cliente: Rol de las personas registradas de nuestro sitio web para realizar el proceso de compra de los documentos. * Distribuidores: Colocar los documentos a disposición que los clientes adquirieron. |

# Responsabilidades del entorno de desarrollo

|  |
| --- |
| * HTML: Utilizamos este lenguaje para poder estructurar y desplegar nuestra aplicación “Docs.” como página web y sus contenidos, como lo son los textos, imágenes de nuestros documentos, etc. * Css: Implementado para poder darle un mejor aspecto/estilo a nuestra página web, que sea mucho más atractiva. * JavaScript: Su responsabilidad en el entorno de desarrollo es añadir características en nuestro sitio web para que sea interactiva, como por ejemplo, las animaciones. * NodeJS: Todas las API se desarrollaron en ella. * MySQL: Es un sistema de gestión de base de datos, por ende, su implementación en nuestra aplicación en almacenar la cantidad de datos para poder llevar un inventario de nuestros documentos y poder consultarlos fácilmente. Además, también se implementaron variables de entorno que permiten la conexión a la base de datos, sim que los datos se vean comprometidos. |

# Cobertura de requisitos

## Requisitos funcionales

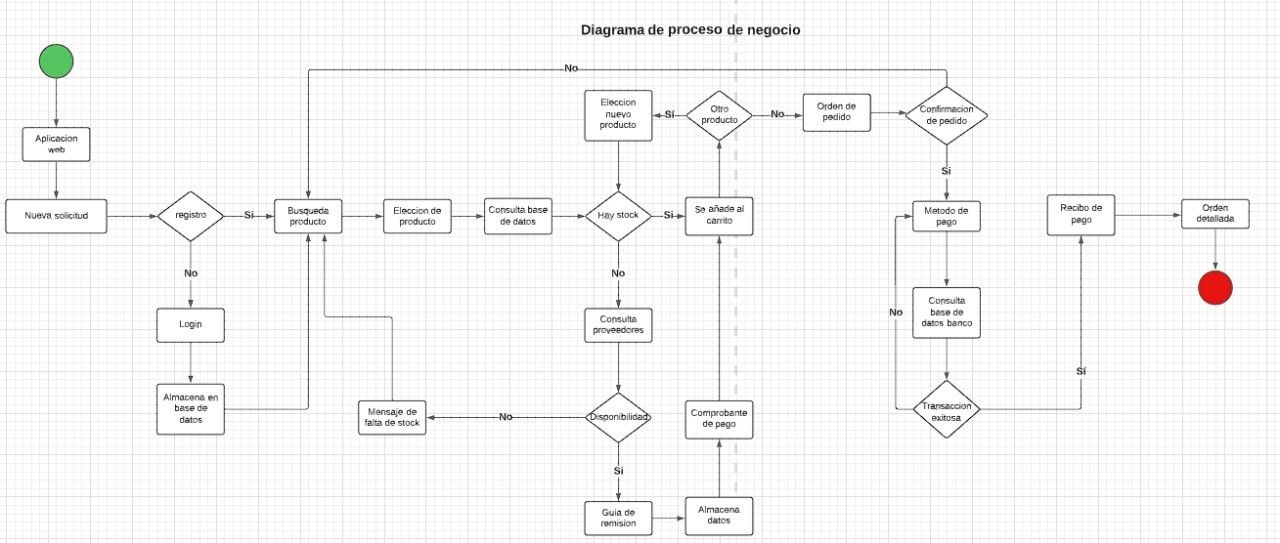
|  |
| --- |
| * Registrar datos del cliente * Consultar los documentos * Consultar los detalles del documento (ya sea libros, artículos, revistas, audios, podcast, entre otros más) * Añadir los documentos al carrito * Modificar los documentos del carrito, como eliminar, agregar o modificar la cantidad * Seleccionar la forma de pago * Elegir la forma de envío * Elegir si deseamos comprar, canjear o alquilar el documento que elegimos * Confirmar pedido |

## Requisitos no funcionales

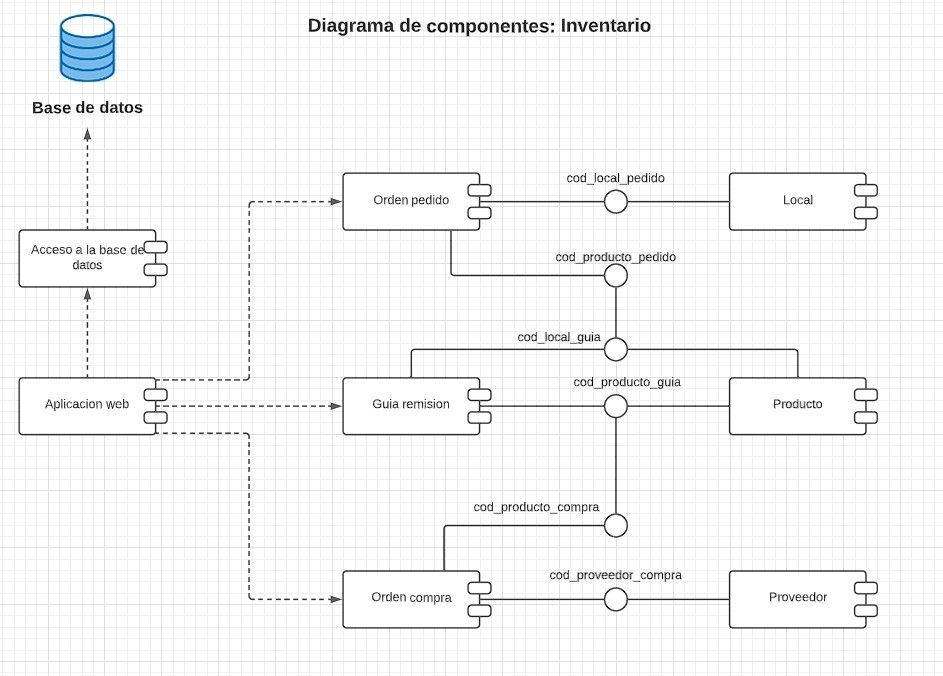
|  |
| --- |
| * Rendimiento en el sistema * El sistema debe ser capaz de modificar la base de datos, dependiendo el incremento o decremento del inventario * El sistema es desarrollado en java, javascrip, la base de datos en MySQL y es desplegado en Heroku |

# Diagramas

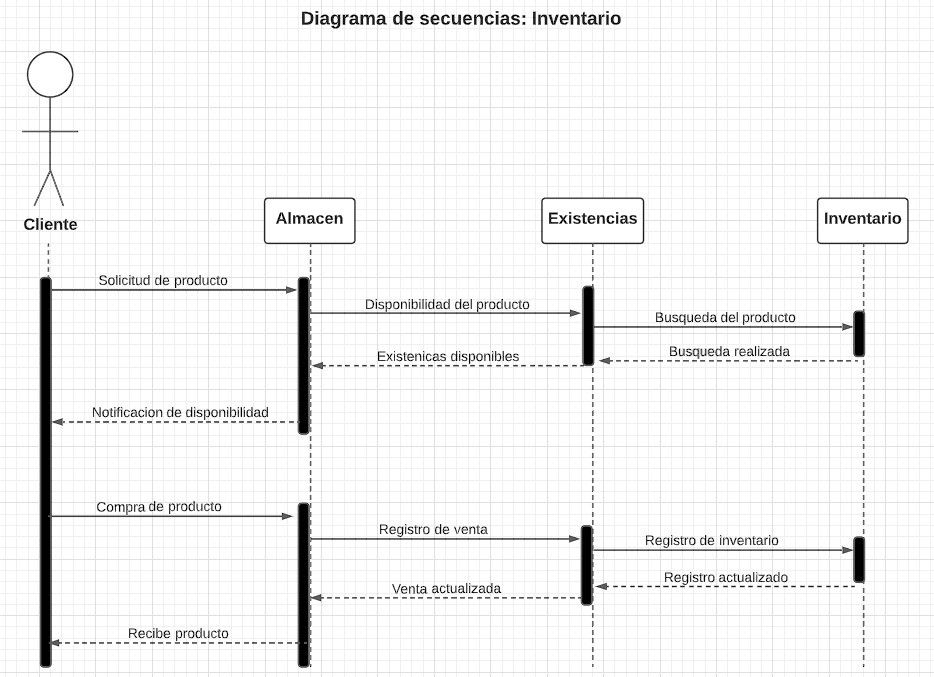
## Diagrama de proceso de negocio



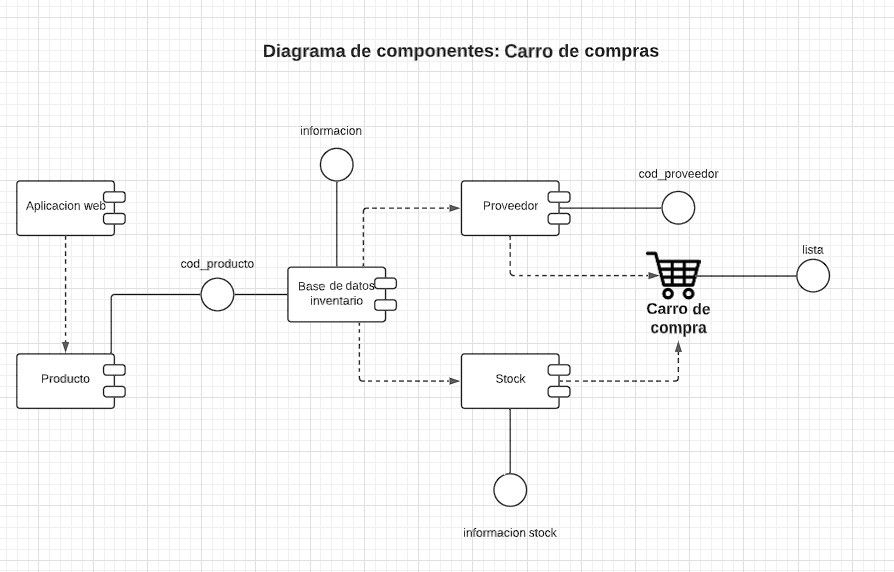
## Diagrama de componentes: servicio de inventario



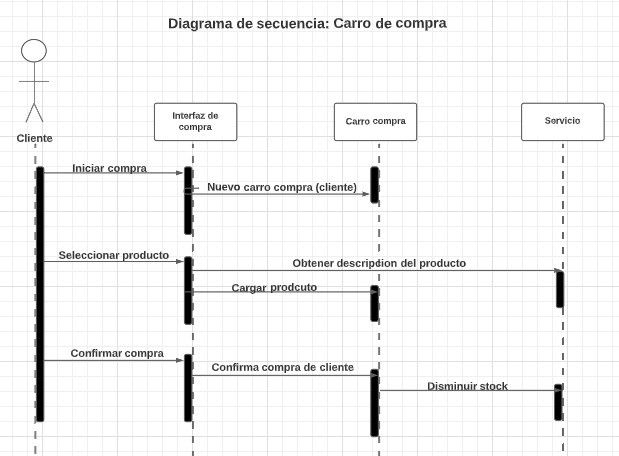
## Diagrama de secuencias: servicio de inventario



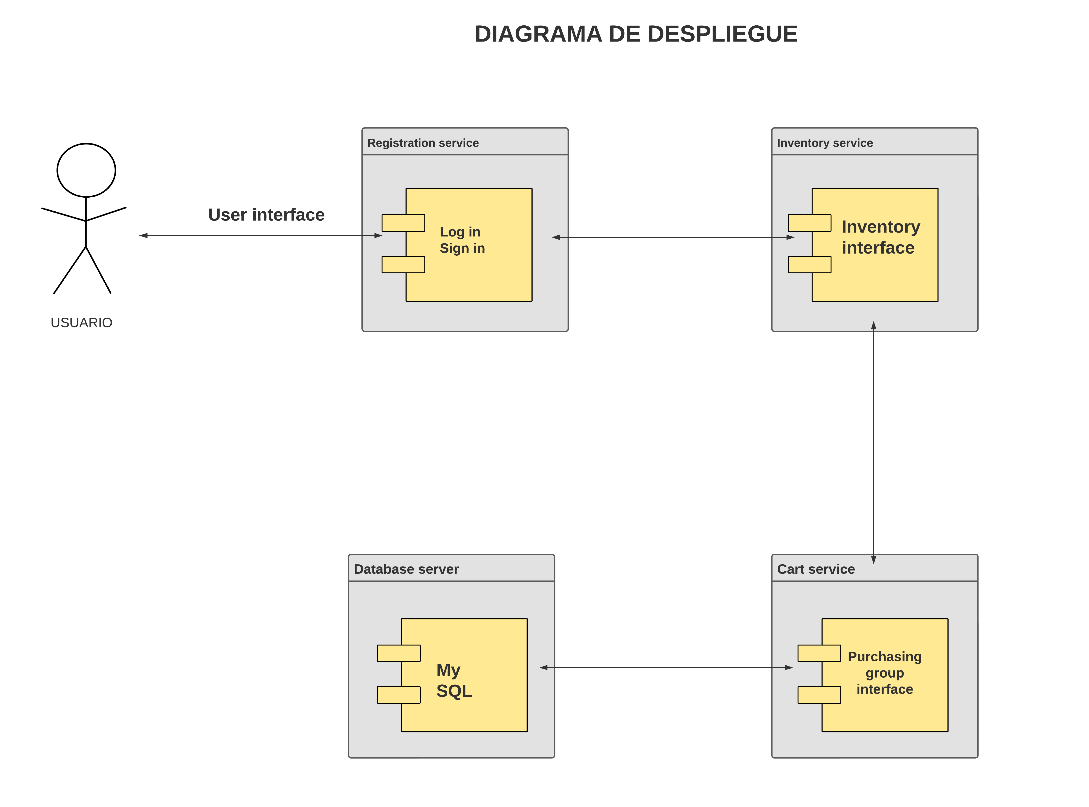
## Diagrama de componentes: servicio de carrito



## Diagrama de secuencias: servicio de carrito

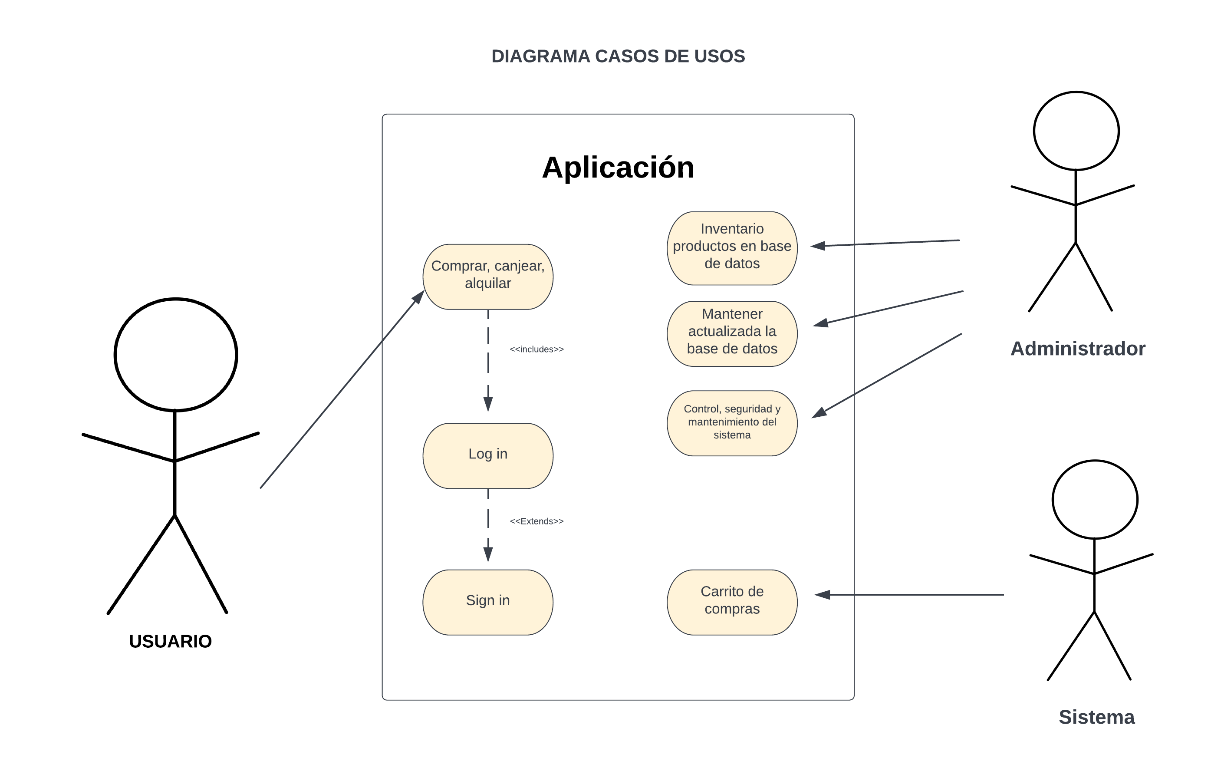


## Diagrama de despliegue de la arquitectura del sistema

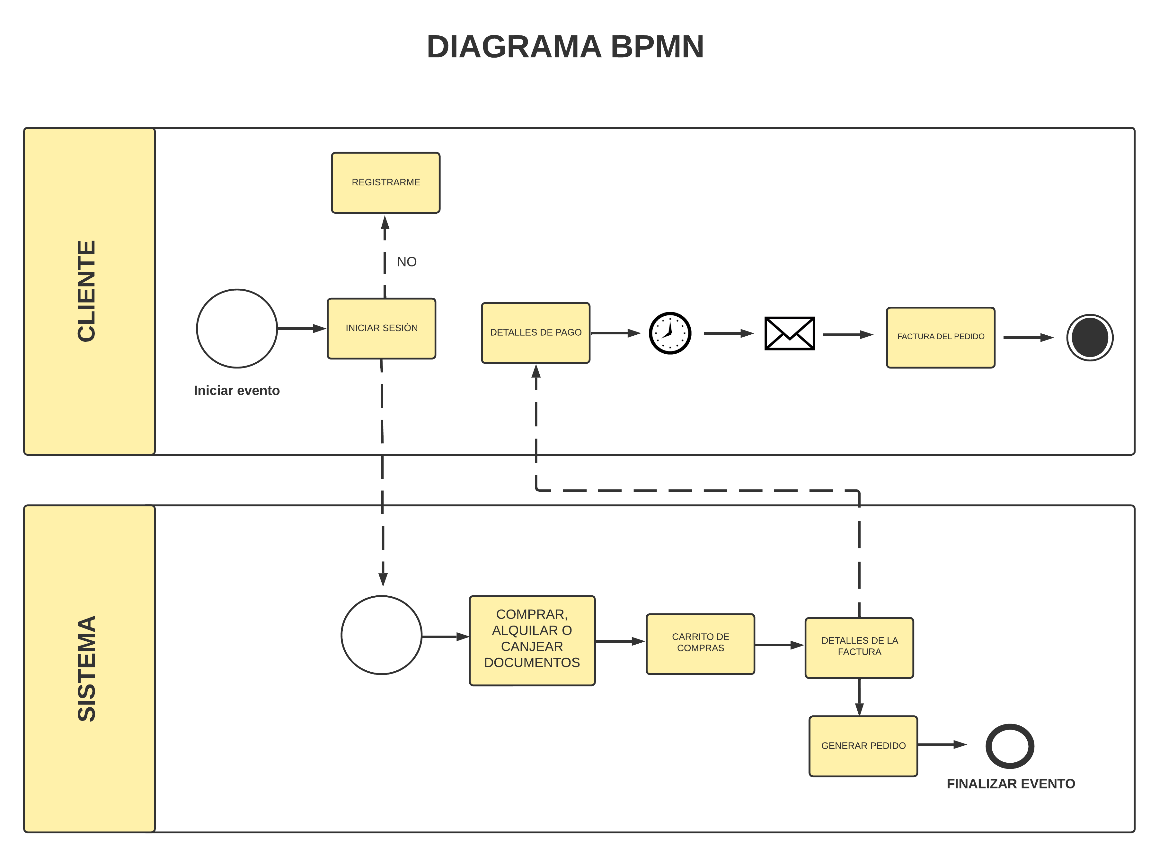


# Diagramas lógicos del sistema

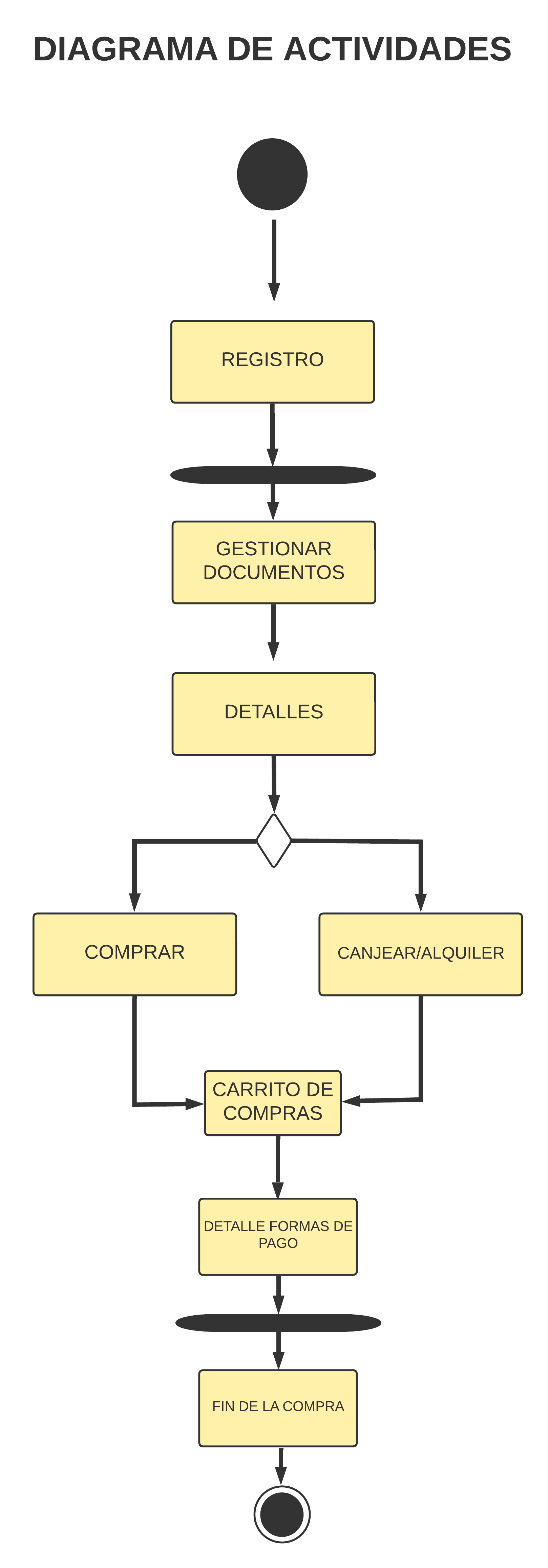
## Diagrama casos de uso



## Diagrama BPMN



## Diagrama de actividades



# Rutas de servicios

*Rutas definidas en el Endpoint.*

|  |
| --- |
| En base a que se trabajó con MYSQL para la manipulación de la base de datos, únicamente se utilizaron los métodos HTPP, GET y POST, los cuales nos permitirán obtener y enviar información a la base de datos. Rutas implementadas. |

## Ruta de servicio: Inventario

|  |
| --- |
| ***/inventario,*** este lo que nos permitirá será obtener el ID, titulo, autor, categoría, detalle, cantidad, precio de compra y de venta, y la URL de la imagen asociada a cada libro, de todos los libros almacenados en la base de datos. |

## Ruta de servicio: Inventario id

|  |
| --- |
| ***/inventario/:id,*** por medio de esta ruta se obtendrán los mismos datos mencionados en la ruta anterior, pero mostrando únicamente la información del libro relacionada al id que se le pase por parámetro. |

## Ruta de servicio: Create inventory

|  |
| --- |
| ***/create\_inventory,*** como su nombre lo indica, esta ruta permite la creación de un nuevo libro, teniendo en cuenta los datos mencionados anteriormente en la ruta ***/inventario.*** |

## Ruta de servicio: Update inventory

|  |
| --- |
| **/update\_inventory**, esta ruta permite la actualización del inventario, para esta sección se tiene en cuenta el ID para realizar la actualización, los campos permitidos para la actualización son cantidad, en caso de que se compren más stock, o precio de compra o de venta, si se requiere actualizar en algún momento. |

## Ruta de servicio: Delete inventory

|  |
| --- |
| ***/delete\_inventory,*** por medio de la presente ruta se eliminará todo el registro de datos que exista de un libro, en esta sección solo es necesario el ID del libro que se desee borrar. |

## Ruta de servicio: Carrito

|  |
| --- |
| ***/carrito,*** este lo que nos permitirá será obtener el ID, titulo, cantidad, y precio de venta de los libros almacenados en la base de datos. |

## Ruta de servicio: Carrito id

|  |
| --- |
| ***/carrito/:id,*** por medio de esta ruta se obtendrán los mismos datos mencionados en la ruta anterior, pero mostrando únicamente la información del libro relacionada al id que se le pase por parámetro. |

## Ruta de servicio: Update carrito

|  |
| --- |
| ***/update\_carrito,*** esta ruta permite la actualización del carrito, para esta sección se tiene en cuenta el ID para realizar la actualización, el stock de productos disponibles se disminuirá de acuerdo con la cantidad que el cliente compre. |

## Ruta de servicio: Login

|  |
| --- |
| ***/login****,* esta ruta es la encargada de permitir a los usuarios el ingreso al sistema, se tiene en cuenta el correo y la contraseña, inicialmente se realiza una comparación de correo para validar que el usuario esté registrado en el sistema, si es así, se realiza la verificación de contraseña donde de manera cifrada se hará la comprobación de la clave almacenada en la base de datos con la que el usuario está digitando. |

## Ruta de servicio: Sigin

|  |
| --- |
| ***/sigin,*** la presente ruta permite la creación de nuevos usuarios al sistema, siempre y cuando el correo y el ID no se encuentren registrados con anterioridad, se requiere el ID, nombre, apellido, correo y contraseña, dicha contraseña estará cifrada por motivos de seguridad. |

## Ruta de servicio: Crear orden

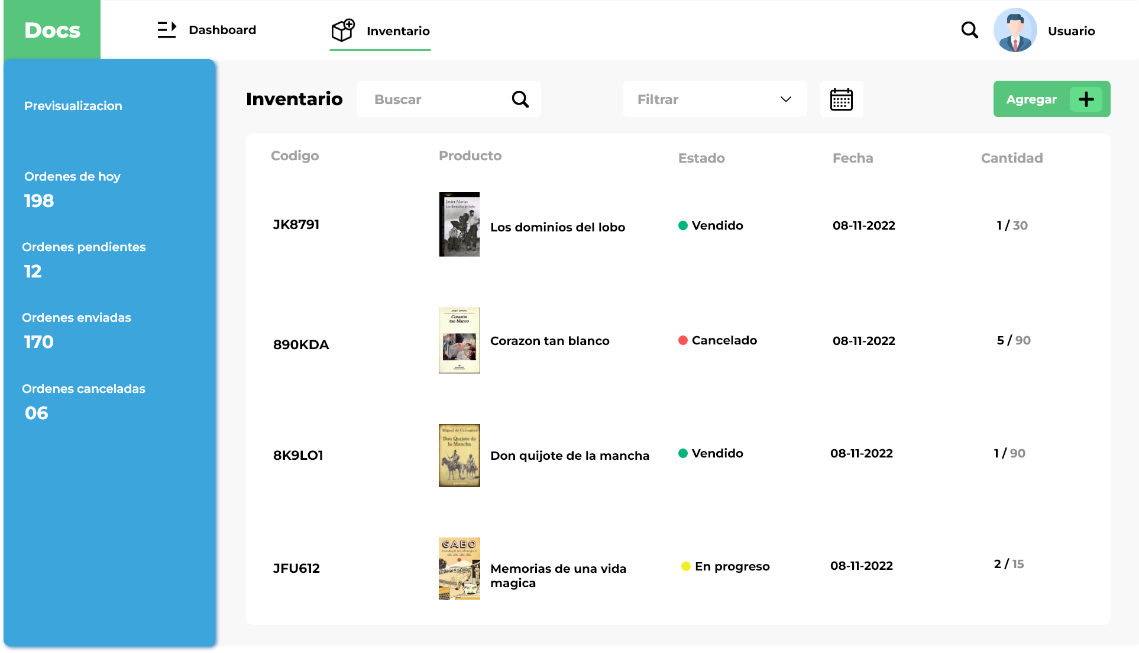
|  |
| --- |
| ***/crear-orden,*** por medio de esta ruta se crear las URL que permitirán realizar y procesar los pagos por medio de Mercado Pago, se espera obtener del body, el título, cantidad y precio unitario de los artículos, donde se hace obligatorio la cantidad y precio de cada artículo para que se pueda crear la preferencia que contendrá todos los ítems seleccionados por el usuario y pueda realizar el pago. |

## Ruta de servicio: Feedback

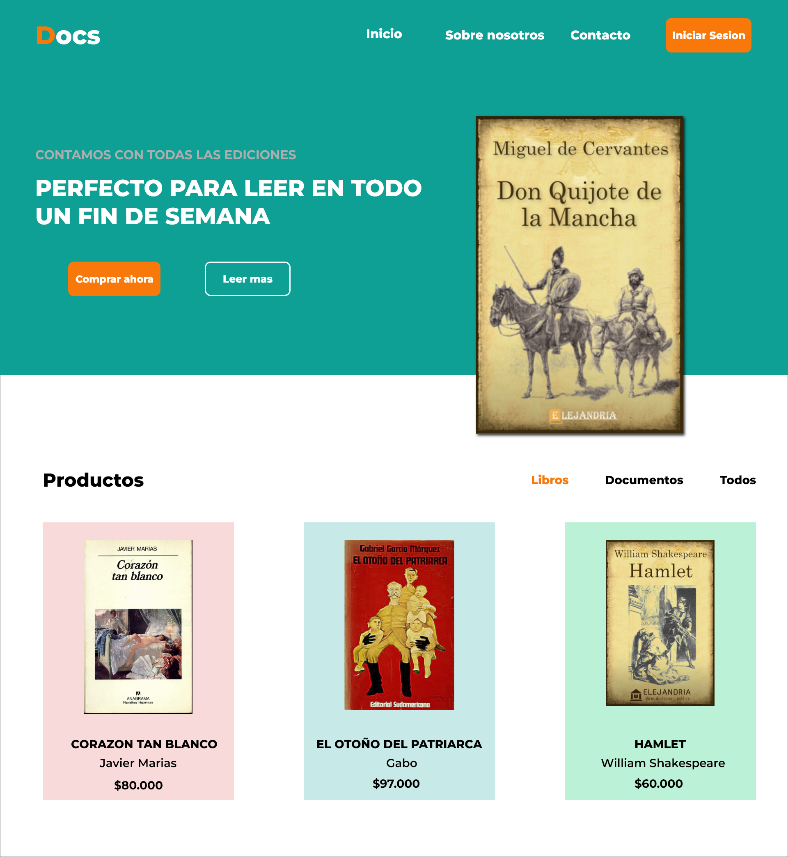
|  |
| --- |
| ***/feedback,*** esta ruta, sencillamente obtendrá los parámetros de ID, estado y número de orden por medio de URL, esto permitirá mostrarle al usuario el estado en que se encuentra su orden. |

# Puntos de vistas

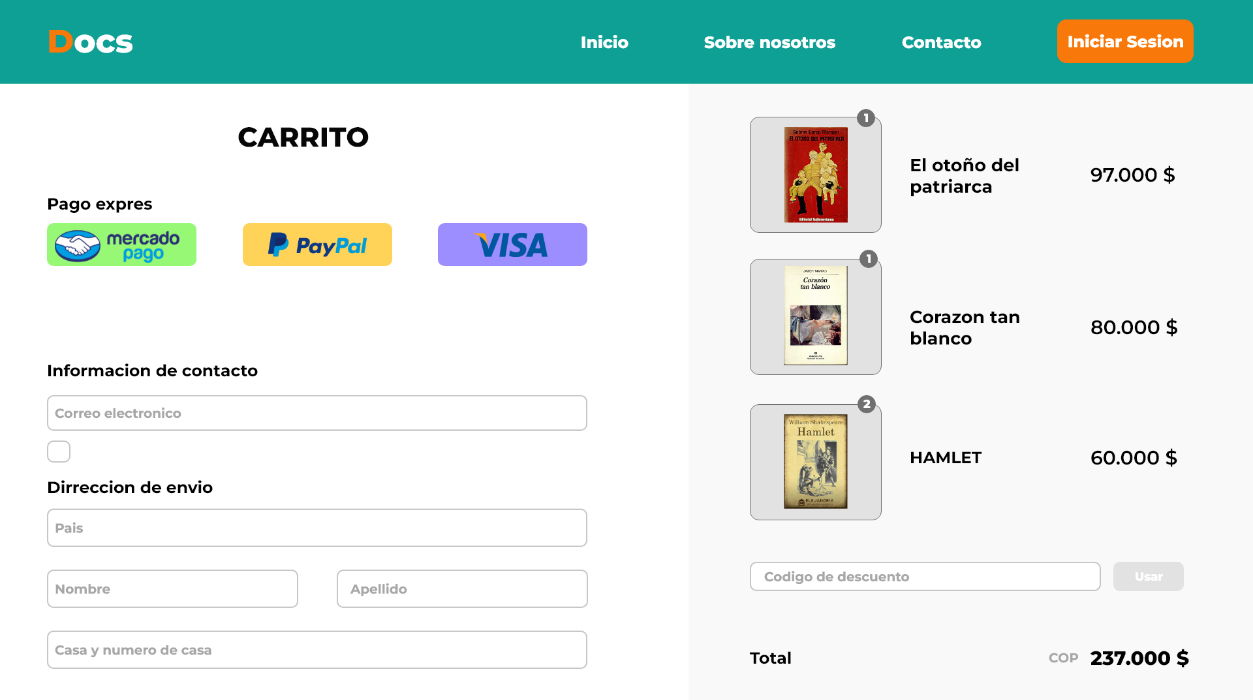
## Mockup inventario



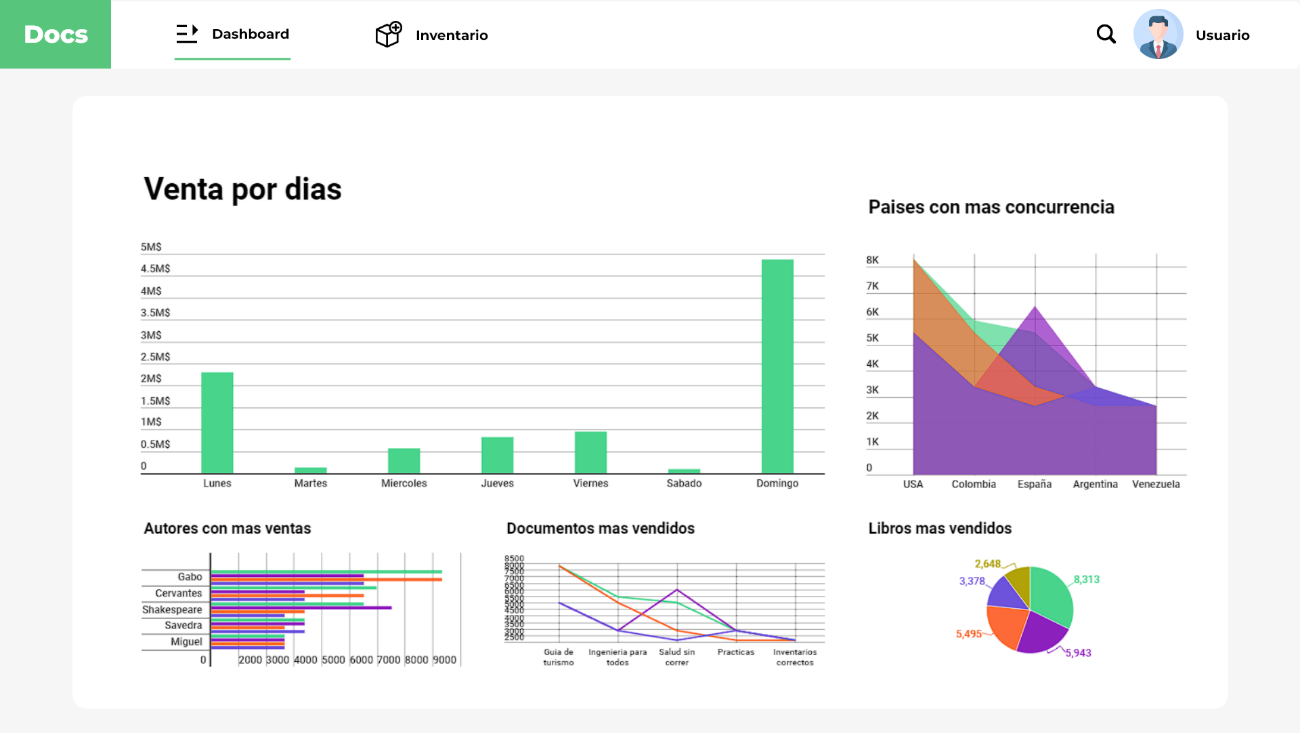
## Mockup catalogo



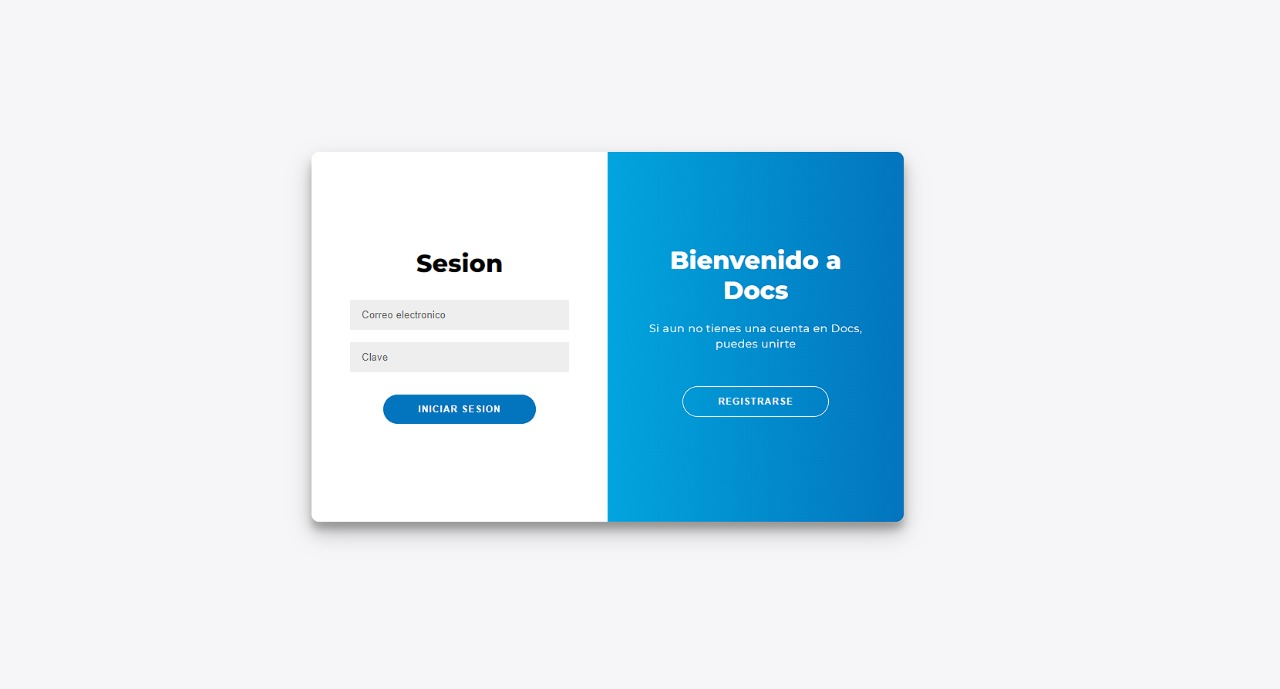
## Mockup carrito



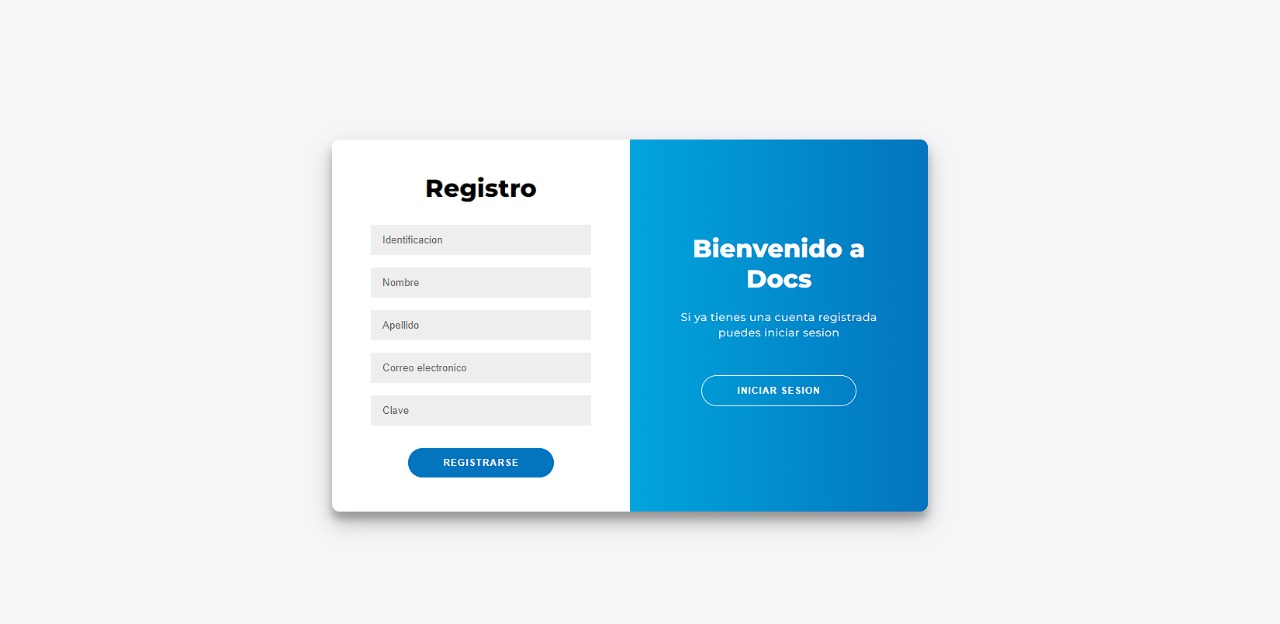
## Mockup dashboard



## Mockup inicio de sesión



## Mockup registro de usuarios



# Descripción de la Arquitectura de hardware a utilizar en el sistema

|  |
| --- |
| Dentro de nuestra aplicación de gestión de documentos trabajaremos con una arquitectura de software que consta de varios componentes, uno de ellos y el más esencial es Heroku, un servidor donde se almacena y procesa toda la información de nuestra aplicación, con esto, para poder trabajar con los demás componentes que son importantes dentro de cualquier empresa, como ordenadores de mesa, dispositivos portátiles, impresoras y escáneres para trabajar de manera profesional y brindarles una mejor atención y soporte a nuestros clientes. |

# Compatibilidad

|  |
| --- |
| En tema de compatibilidad e interpolación, nuestro sistema cuenta con la capacidad de que nuestros componentes puedan intercambiar cada información que recibe de una manera excelente y también cada una llevar a cada su función en específico. |

# Informes JIRA

## Informes de trabajo completado

Gráfico, Gráfico de líneas

Descripción generada automáticamenteTablero Sprint 1

Tablero Sprint 2

Gráfico

Descripción generada automáticamenteGráfico, Gráfico de líneas

Descripción generada automáticamenteTablero Sprint 3

Tablero Sprint 4

Gráfico, Gráfico de líneas

Descripción generada automáticamenteGráfico

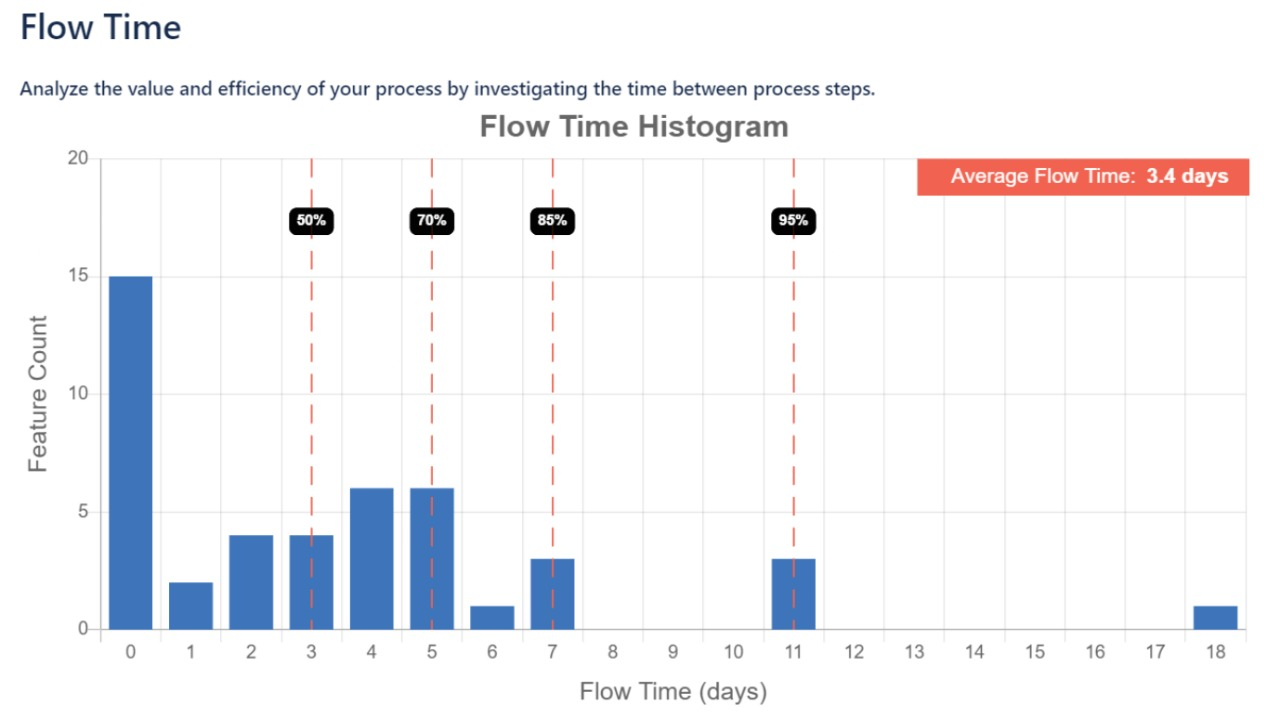
Descripción generada automáticamenteTablero Sprint 5

Tablero Sprint 6

## Gráfico, Gráfico de barras Descripción generada automáticamenteInforme de velocidad

## Gráfico, Histograma Descripción generada automáticamenteDiagrama de flujo acumulado

## Flow Time



## Created vs Resolved

# Recursos

## Enlace repositorio

|  |
| --- |
| Repositorio GitHub: <https://github.com/ISCODEVUTB/App_DMS_SOA> |

# Agradecimientos

|  |
| --- |
| Agradecer a nuestro profesor de Arquitectura de Software, Edwin Alexander Puertas del Castillo, por su dedicación y tiempo hacia nosotros en cada una de las reuniones de trabajo con el equipo de trabajo compuesto por: Lucas Mateo Pedrozo Pertuz, José Miguel Mendoza Sepúlveda y Michael Andres Casadiegos Berrio. Para así, poder realizar cada una de las incidencias y lograr llegar con éxito al trabajo final de la Aplicación de gestión documental. |