**Documento de Arquitectura de Software**

**Proyecto EzCommerce**

**Versión 1.0**

**Control de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 21/09/2024 | 1.0 | Creación del documento. | Espinoza Peralta |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ÍNDICE**

[1. Introducción 4](#_heading=h.f9lx5fz8p2zk)

[1.1. Propósito 4](#_heading=h.acj1o92okewf)

[1.2. Alcance 4](#_heading=h.jhczf1mjb7kq)

[1.3. Definición, siglas y abreviaturas 4](#_heading=h.78lqgges3d8j)

[1.4. Vista global 5](#_heading=h.cfzjj3a5c434)

[2. Macro Arquitectura 5](#_heading=h.jhgu102miua1)

[2.1. Modelo multicapa 5](#_heading=h.f9lx5fz8p2zk)

[2.2. Metas y restricciones arquitectónicas 5](#_heading=h.f9lx5fz8p2zk)

[3. Vista Física 7](#_heading=h.rsx7tau9q47u)

[4. Vista Funcional o Lógica 8](#_heading=h.f9lx5fz8p2zk)

[5. Vista de despliegue 9](#_heading=h.f9lx5fz8p2zk)

[6. Vista de procesos 11](#_heading=h.fworok1v4vvb)

# Introducción

## Propósito

El propósito de este documento es definir y dar a conocer la Arquitectura de Software para el desarrollo del proyecto "EzCommerce”. Se utilizarán las vistas de proceso, lógica y física, para describir la estructura y funcionalidad del sistema, así como su distribución física, interacción y descripción general.

Este documento está dirigido al equipo de desarrollo, con el fin de proporcionar las bases arquitecturales y una visión global del diseño para llevar a cabo el proyecto.

## Alcance

Este documento describe la arquitectura del sistema "EzCommerce", el cual permitirá a los usuarios buscar productos y contactarse con los vendedores, asimismo también podrán publicar artículos y registrar su marca. El sistema se enfocará en mejorar los procesos internos y externos relacionados con la venta y compra de articulos. La aplicación será web y funcionará en dispositivos con navegación web estándar. Además, se especifican las restricciones, atributos de calidad, vistas arquitecturales y patrones de diseño a implementar por el equipo de desarrollo.

## Definición, siglas y abreviaturas

* DAS: Documento que describe la arquitectura de software.
* API: Interfaz que permite a diferentes aplicaciones comunicarse entre sí.
* HTTP: Protocolo utilizado para transferir datos en la web.
* IIS (Internet Information Services): Servidor web utilizado para alojar aplicaciones y sitios web.
* DBMS: Sistema que administra bases de datos.
* SQL (Structured Query Language): Lenguaje utilizado para interactuar con bases de datos.
* GUI (Graphical User Interface): Interfaz visual que permite interactuar con un programa.
* React: Biblioteca de JavaScript para crear single page applications.
* Django: Framework de Python para desarrollar API’s .
* Responsive Design: Diseño que se adapta a diferentes dispositivos y tamaños de pantalla.
* Singleton: Patrón de diseño que garantiza una única instancia de una clase.
* Modelo Multicapa (Modelo n-tier): Patrón de arquitectura de software que separa la lógica de negocio de la interfaz del usuario en capas.

## Vista global

Se presentará una macro arquitectura del sistema "EzCommerce", con especificaciones de las vistas utilizadas en el proyecto, metas y restricciones arquitectónicas, así como vistas con diagramas para una comprensión completa del sistema.

# Macro Arquitectura

## Modelo multicapa

El modelo multicapa, también conocido como 3-tier, se utiliza para separar la lógica de negocio de la interfaz del usuario en distintos componentes, ofreciendo:

* Capa de Presentación/UI: Interfaz de usuario, componentes Front-End.
* Capa de Lógica de Negocio: Gestión de procesos, reglas de negocio, controladores, middlewares y enrutador.
* Capa de Acceso a Datos: Gestión de datos, Interacción con la base de datos, ORM y APIs.

El principal beneficio de la arquitectura de tres niveles es su separación lógica y física de la funcionalidad. Cada nivel puede ejecutarse en un sistema operativo y una plataforma de servidor independientes (por ejemplo, servidor web, servidor de aplicaciones, servidor de base de datos) que mejor se adapte a sus requisitos funcionales. Además, cada nivel se ejecuta en al menos un hardware de servidor o un servidor virtual dedicado, por lo que los servicios de cada nivel pueden personalizarse y optimizarse sin afectar los demás niveles. Lo que mejora la mantenibilidad, escalabilidad y flexibilidad del sistema, ya que cada capa cumple una función específica, facilitando la incorporación de nuevas funcionalidades y adaptaciones futuras.

## Metas y restricciones arquitectónicas

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributos de calidad “Observables”** | | | |
| Atributo | Descripción | Tácticas /Patrón de  Arquitectura | Aplicación |
| Seguridad | Garantizar que el acceso a los recursos y funcionalidades esté restringido según los roles asignados a los usuarios autorizados. | Control de Acceso Basado en Roles: Asignación de permisos concretos al usuario para poder comprar o vender productos. | Aplicación al momento de autenticación de usuarios y accesos a funcionalidades específicas. |
| Disponibilidad | Capacidad del sistema para estar operativo y accesible para los usuarios en todo momento. | Balanceo de Carga: Distribución equitativa de la carga de trabajo entre varios servidores para evitar sobrecargas. | Implementación en la infraestructura de servidores de EzCommerce. |
| Usabilidad | Facilidad de uso del sistema para los usuarios finales. | Diseño Centrado en el Usuario (UCD): Integración de procesos de diseño y pruebas centrados en la experiencia del usuario. | Desarrollo de interfaces intuitivas, para el manejo de operaciones de compra y venta. |

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributos de calidad “No observables”** | | | |
| Atributo | Descripción | Tácticas /Patrón de  Arquitectura | Aplicación |
| Modificabilidad | Facilidad para realizar cambios en el sistema, como agregar nuevas funcionalidades o adaptarse a requisitos cambiantes. | Arquitectura Multicapa: Descomposición del sistema en servicios independientes y autocontenidos. | División de la lógica de negocios, gestión de recursos y acceso a datos en microservicios. |
| Mantenibilidad | Capacidad de realizar mantenimiento y mejoras al sistema de manera eficiente y económica. | Separación de capas: División clara de las capas de presentación, lógica de negocios y acceso a datos para facilitar las actualizaciones. | Implementación de patrones de diseño y estándares de codificación que favorezcan la legibilidad, cohesión y reutilización del código. |
| Portabilidad | Adaptabilidad del sistema para funcionar en diferentes entornos. | Diseño Responsivo: Desarrollo de interfaces de usuario que se ajustan automáticamente. | Implementación de técnicas de diseño web responsivo para una experiencia de usuario consistente y accesible. |

# Vista Física

Diagrama informal de la vista física de EzCommerce:

El Diagrama informal de la vista física contempla el uso del aplicativo por parte del usuario, quien ingresa al navegador desde diversos dispositivos, como computadoras de escritorio, portátiles y dispositivos móviles. El usuario accede a la página web de EzCommerce, la cual está alojada en un servidor de aplicaciones.

Elementos físicos:

* Usuarios: Personas que acceden al sistema desde diferentes dispositivos.
* Dispositivos electrónicos: Computadoras de escritorio, portátiles y dispositivos móviles con conexión a internet y que sean capaces de ejecutar un navegador.
* Internet: La red global que permite la comunicación entre todos los dispositivos y servidores.
* Servidor: Recursos informáticos alojados y administrados por terceros dando garantía de la integridad y la disponibilidad de la aplicación.
* Base de datos: Almacena la información del sistema, como datos de libros, usuarios y reservas. Utiliza PostgreSQL como gestor de bases de datos.

Atributos de calidad:

**Escalabilidad**: El sistema está diseñado para crecer con la carga de trabajo, utilizando un enfoque de diseño multicapa que separa la lógica de negocio y los datos de la interfaz de usuario. Esto permite que nuevas funcionalidades se implementen fácilmente manipulando las capas del sistema.

**Portabilidad:** EzCommerce implementa el patrón de diseño Responsive Design, lo que le permite adaptarse a diferentes dispositivos con diferentes tamaños de pantalla, como computadoras de escritorio, portátiles y dispositivos móviles.

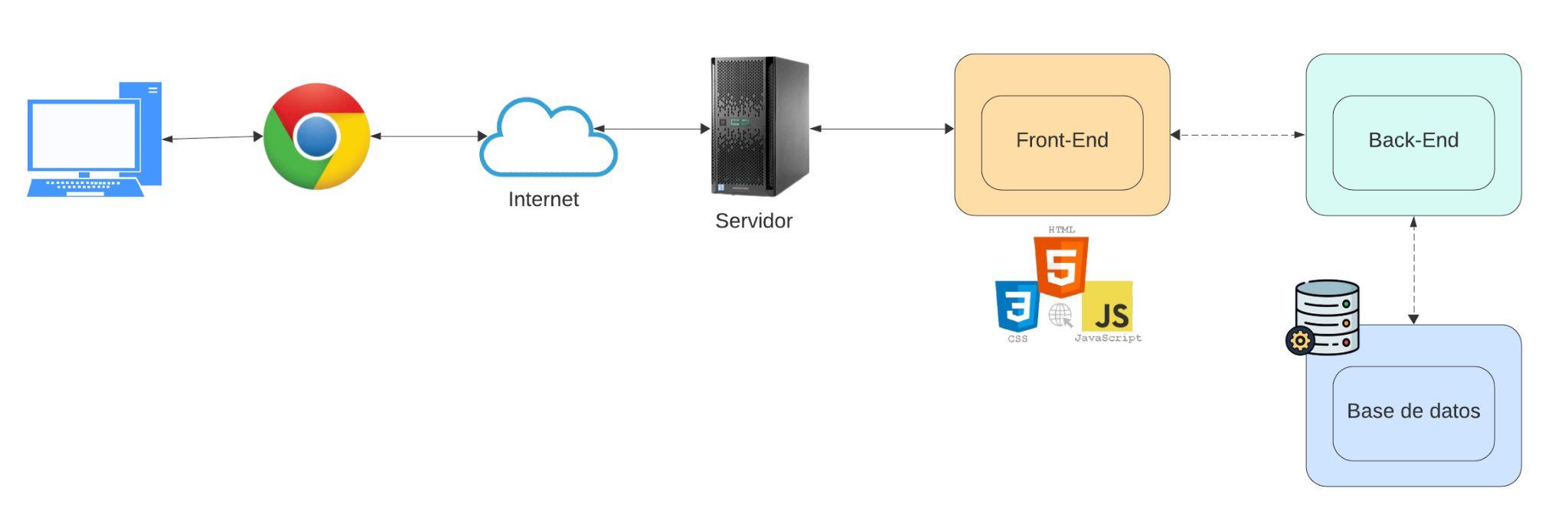


Figura 01. Diagrama informal físico - EzCommerce

# Vista Funcional o Lógica

Diagrama de Componentes de EzCommerce:

El siguiente diagrama representa el marco de trabajo de la relación entre los componentes que se encuentran dentro de esta arquitectura para la funcionalidad del aplicativo web. En este diagrama se ve reflejado que el sistema es multicapa representado por la capa de presentación, capa de lógica del negocio y capa de acceso a datos, lo que permite una óptima y eficiente sostenibilidad ,y mayor funcionalidad del sistema.

Componentes:

* **Capa de Presentación:**

Descripción: Responsable de la interfaz de usuario del aplicativo web.

* **Capa de Lógica de Negocio:**

Descripción: Gestiona la lógica de procesos y funcionalidades del sistema.

Enrutador, Middlewares y Controladores: Utiliza un enrutador para redireccionar las solicitudes mientras que los controladores sirven para responder adecuadamente a las solicitudes, los cuales ejecutan funciones específicas de acuerdo con la ruta invocada. En adición los middlewares nos ayudan a establecer reglas que deben cumplir las solicitudes para poder ejecutarse Incluye funciones propias del marco de trabajo y lógica de procesos del sistema.

* **Capa de Acceso a Datos:**

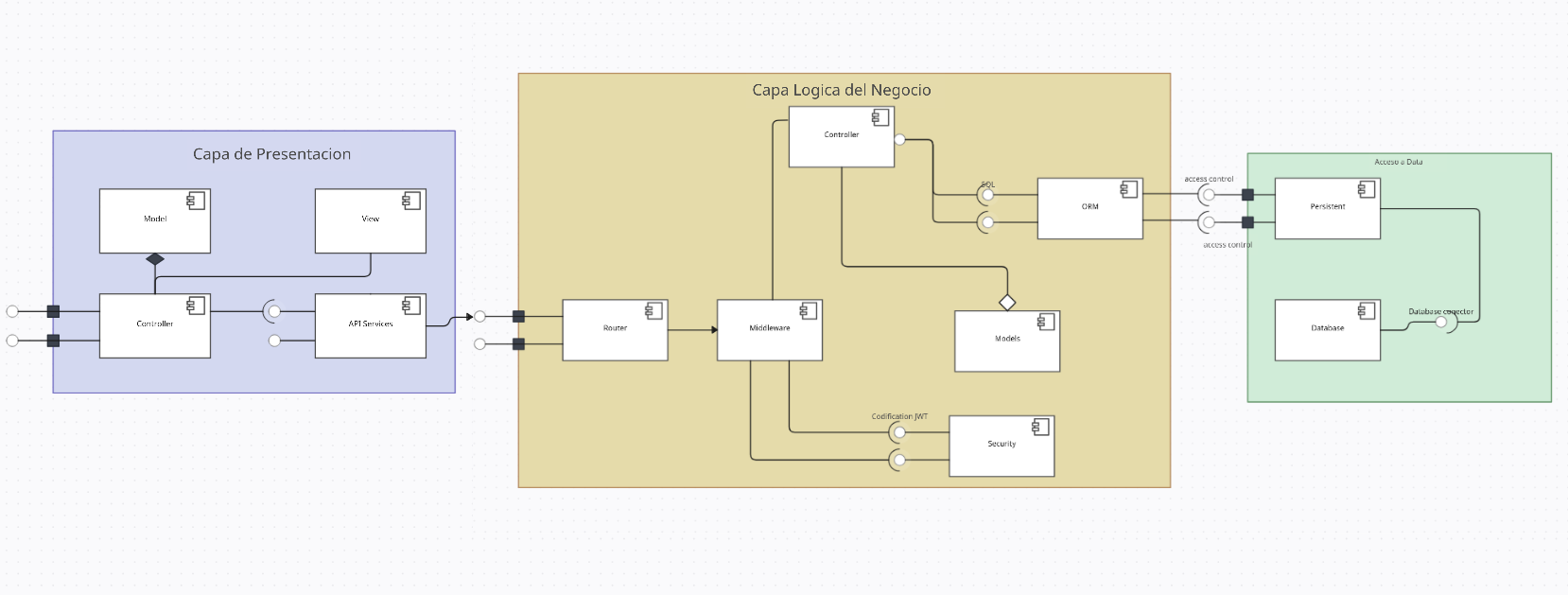
Descripción: Encargada del acceso directo a la base de datos y la gestión de la capa de persitencia.

Atributos de Calidad:

**Mantenibilidad**: Se garantiza mediante la alta cohesión de componentes, la separación de tareas y la aplicación del patrón multicapa. Esto permite que el sistema sea dinámico y adaptable a las necesidades de expansión, facilitando la modificación y el mantenimiento del aplicativo web.

**Modificabilidad**: Se logra a través de la separación de tareas y la estandarización de componentes, lo que permite realizar cambios en una parte de la aplicación sin afectar los demás componentes. Esto se logra gracias al diseño multicapa y la modularización del sistema.

**Funcionalidad**: Se garantiza mediante la estandarización de componentes y la interoperabilidad del aplicativo. El uso del patrón multicapa y la separación de responsabilidades facilitan la integración de nuevos componentes y funcionalidades, mejorando así la funcionalidad global del sistema.

Figura 02. Diagrama de componentes - EzCommerce

# Vista de despliegue

Diagrama de Despliegue de EzCommerce:

Este diagrama describe las configuraciones de redes físicas sobre las cuales el software será desarrollado. Se indican los nodos físicos que interactúan con la aplicación web, tales como un nodo para el dispositivo donde se accede a la aplicación web en un PC con acceso a Internet y un navegador. Este envía las peticiones por HTTP para mayor seguridad de la información accediendo al nodo del Web Server donde aloja el Internet Information Service (IIS) HTTP Server y El DBMS con el componente de base de datos PostgreSQL.

Componentes:

* **Nodo de Acceso a la Aplicación Web:**

Descripción: Se refiere a cualquier dispositivo que pueda ejecutar un navegador y conectarse a internet.

Protocolo de Comunicación: HTTP/2

Interacción: Envía peticiones de tipo HTTP al servidor web para interactuar con la aplicación.

* **Nodo del Servidor Web:**

Descripción: Aloja el Internet Information Service (IIS) HTTP Server.

Funcionalidad de la Aplicación: Divide la funcionalidad de la aplicación en capas de presentación, lógica del negocio lo que permite una mejor sostenibilidad a través del tiempo.

* **Nodo del Servidor de la Base de Datos:**

Descripción: Aloja toda los datos en un servidor propio y los gestiona a traves de un DBMS con el componente de base de datos PostgreSQL.

Funcionalidad de la Aplicación: Administra unicamente la información y se conecta al servidor web a través del protocolo SSH.

* **Lógica de Negocios y Backend:**

Descripción: Integra las interfaces de APIs e intermedia la interaccion del usuario con la data.

Interacción: Gestiona la lógica de negocio y la comunicación con la capa de acceso a datos para realizar operaciones en la base de datos.

Atributos de Calidad:

**Modificabilidad**: Se aplica mediante la separación de tareas y la modularización del sistema, lo que permite realizar cambios en una parte de la aplicación sin afectar los demás componentes. El diseño multicapa facilita la modificación y mantenimiento del sistema, lo que mejora su adaptabilidad y escalabilidad a medida que evoluciona el proyecto.

Seguridad: Separar la capa de base de datos en otro servidor nos permite proteger nuestra data de posibles atacantes u otras vulnerabilidades.



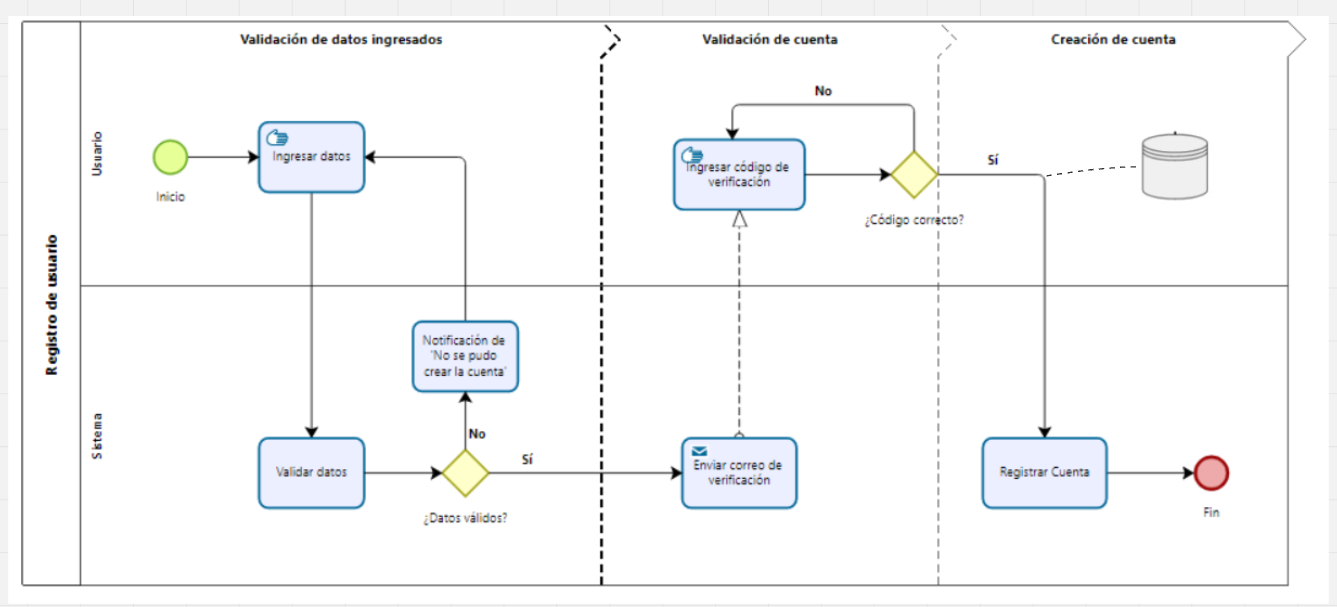
HTTP

SSH

Figura 03. Diagrama de despliegue - EzCommerce

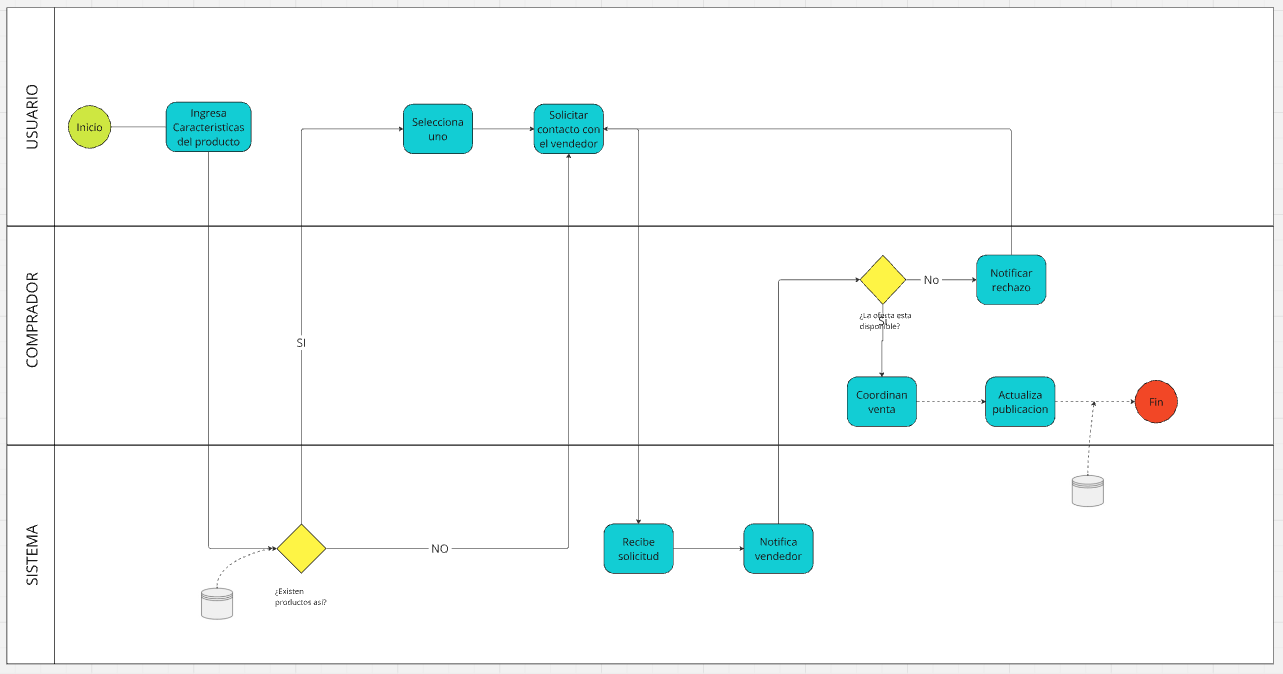
# Vista de procesos

Diagrama de actividades: Registro de Usuarios



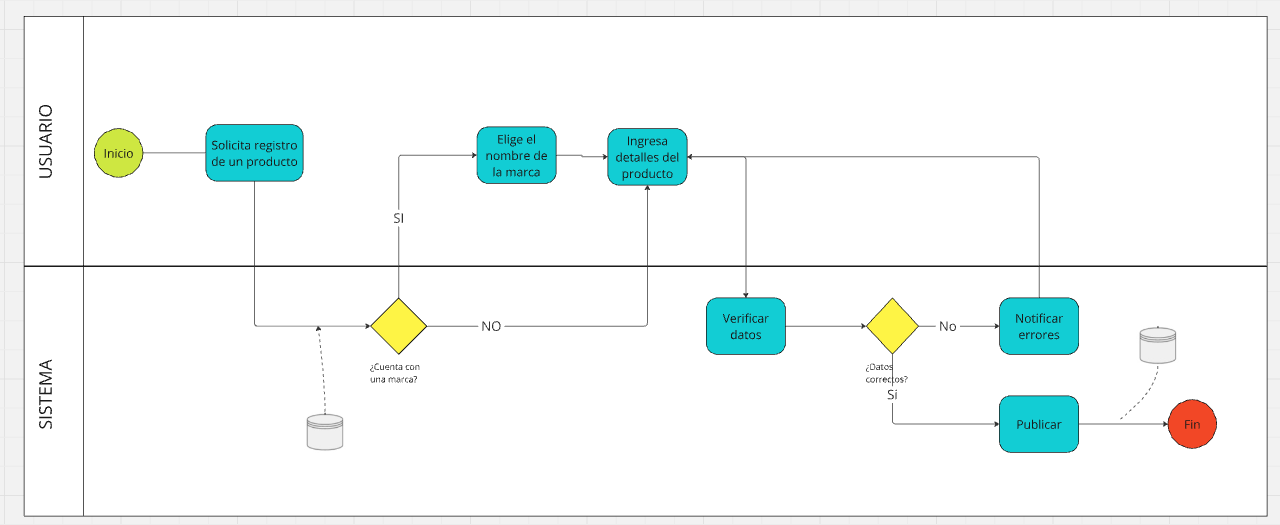
El usuario se registra con sus datos, estos datos en primera instancia son validados, posterior a esto se le enviara un correo al email registrado con la finalidad de validar el correo, en caso se obtenga la confirmacion el usuario sera creado y registrado en la base de datos.

Diagrama de Actividades: Proceso de compradores



El usuario busca un articulo de interes dentro del catalogo de la aplicación, una vez que lo encuentra lo enfoca y solicita contacto con el vendedor, el sistema lo verifica y envia la solicitud al vendedor, si la oferta esta disponible el vendedor acepta la solicitud y se comunican para coordinar la venta, posteriormente en caso se concrete la venta se actualizara el inventario del vendedor.

Diagrama de Actividades: Publicacion de Producto



El usuario desea publicar un producto para lo cual primero solicita un formulario a lo que el sistema verifica si tiene una marca en caso de que si, el producto se publicara bajo la marca, en caso contraria sera registrado sin marca, postumamente se verificaran los datos del producto, lo cual sera verificado por el sistema y si es el caso el producto sera publicado.