

SENA

ANÁLISIS Y DESARROLLO DE SOFTWARE

EVIDENCIA

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

APRENDIZ

Duván Felipe Méndez Ayala

BOGOTA D.C.

SENA CENTRO DE DISEÑO Y METROLOGÍA

octubre de 2024

# Informe técnico

Este documento presenta el manual técnico del sistema de información para la tienda Stunt Rider, desarrollado con el objetivo de optimizar el proceso de ventas físicas y virtuales, gestionar el inventario de productos y brindar una experiencia de usuario eficiente y segura. El manual integra el análisis del problema, los objetivos, los requerimientos funcionales y no funcionales, la propuesta técnica, los diagramas de diseño y el plan de pruebas e implantación.

# Technical report

This document presents the technical manual of the information system for Stunt Riders Store, developed to optimize physical and virtual sales processes, manage product inventory, and provide an efficient and secure user experience. The manual integrates problem analysis, objectives, functional and non-functional requirements, technical proposal, design diagrams, and testing and deployment plan.

# Introducción

El tema del proyecto es el desarrollo de un sistema de información para venta de accesorios de moto para la tienda Stunt Raiders ubicada en la ciudad de Bogotá, de acuerdo con la información recolectada mediante la aplicación de técnicas como la entrevista y las encuestas, donde se deduce que hay la necesidad de desarrollar este software que permite a los usuarios explorar un catálogo actualizado, realizar compras y estar al tanto de las últimas promociones y productos nuevos. En el desarrollo del sistema de información, realizamos un análisis exhaustivo de los requerimientos funcionales que debe cumplir este software, dividiéndolos en los siguientes módulos: usuario, inventario, compra y pago. Posteriormente, iniciamos la fase de diseño y programación.

*Ilustración 1 Tienda física*



# Planteamiento del problema

## 5.1 Descripción del problema

Las actividades de levantamiento de información realizada en la empresa, dio como resultado la identificación del problema de la venta de productos virtualmente ya que bien en la tienda están físicos, en el área virtual hay una desorganización por parte de la tienda, en la cual no tienen una comunicación asertiva con los clientes la que ha llevado como consecuencia malestar, disminución en la confianza y pérdida de ingresos.

Para solucionar la problemática descrita se propone crear un e-commerce en APP para la compra de accesorios en la cual contara con un catálogo actualizado de cada producto disponible, seccionado en repuesto, accesorio, ropa de la marca etc.…

¿Cómo un sistema de información mejora la organización en el área de venta virtual?

## 5.2 Objetivo general

Construir un sistema de información para la tienda Stunt Raiders, ubicada en la ciudad de

Bogotá, localidad Santafé orientado a la web.

## 5.3 Objetivos específicos

* Realizar el levantamiento de información para encontrar el problema e identificar los requerimientos de la empresa.
* Analizar la información obtenida y proponer soluciones creativas al problema identificado.
* Diseñar el modelo de acuerdo con el criterio de la empresa.
* Implementar una base de datos funcional para el sistema de información.
* Implantar la solución informática de acuerdo con la configuración negociada con el cliente.

## 5.4 Justificación

Haciendo una revisión de las páginas web y APP en páginas como mundimotos, motorepuestos y motopartes que son páginas de e-commerce de repuestos y productos para el mantenimiento funcional de la moto mas no de accesorios como el que vamos a realizar. Dada la búsqueda de un software similar al que buscamos para poder enriquecer nuestros conocimientos en este mundo; No existe por el momento una página web de accesorios de motos especialmente para el deporte de Stunt. Basándonos en estos vamos a desarrollar un e-commerce único con un estilo Stunt que sea atractivo para nuestros stakeholders. Será fácil de usar, permitiendo a los usuarios ver productos y sus características de manera clara. Además, implementaremos un método de pago sencillo que incluirá las nuevas tecnologías, como aplicaciones de pago (pse).

Por lo anterior, la solución planteada permite mejorar el proceso y por lo tanto este trabajo se justifica a partir de los siguientes factores:

* El producto de este trabajo es conveniente para el empresario desde el punto de vista de mejoramiento las operaciones virtuales haciéndolo más eficaz y eficiente en las prácticas diarias del mismo.
* Desde el punto de vista social el proyecto aporta a los empleados, a los clientes y a los dueños una herramienta que les facilita sus tareas diarias.
* El rendimiento en el desarrollo de las actividades permite a los implicados proponer nuevas prácticas y atender otros asuntos para la obtención de mejores resultados.
* El conocimiento, a fondo de los procedimientos e implicados en los procesos permite fortalecer el análisis.
* El análisis adecuado de los datos recolectados contribuye en la propuesta de mejores prácticas y mayor rendimiento en el desarrollo de los procesos.

## 5.5 Alcance del Proyecto

El proyecto se llevará a cabo en la tienda Stunt Raides ubicada en Bogotá en el área de ventas. El desarrollo del proyecto tendrá una espera de 12 meses, desde el momento de la aceptación del proyecto.

El software no realizara actividades adicionales que no estén definidas dentro del dominio del problema identificado.

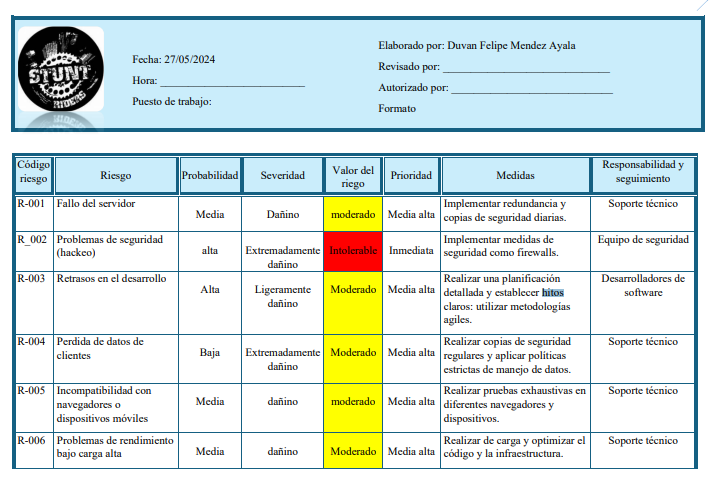
## 5.6 Matriz de riesgo

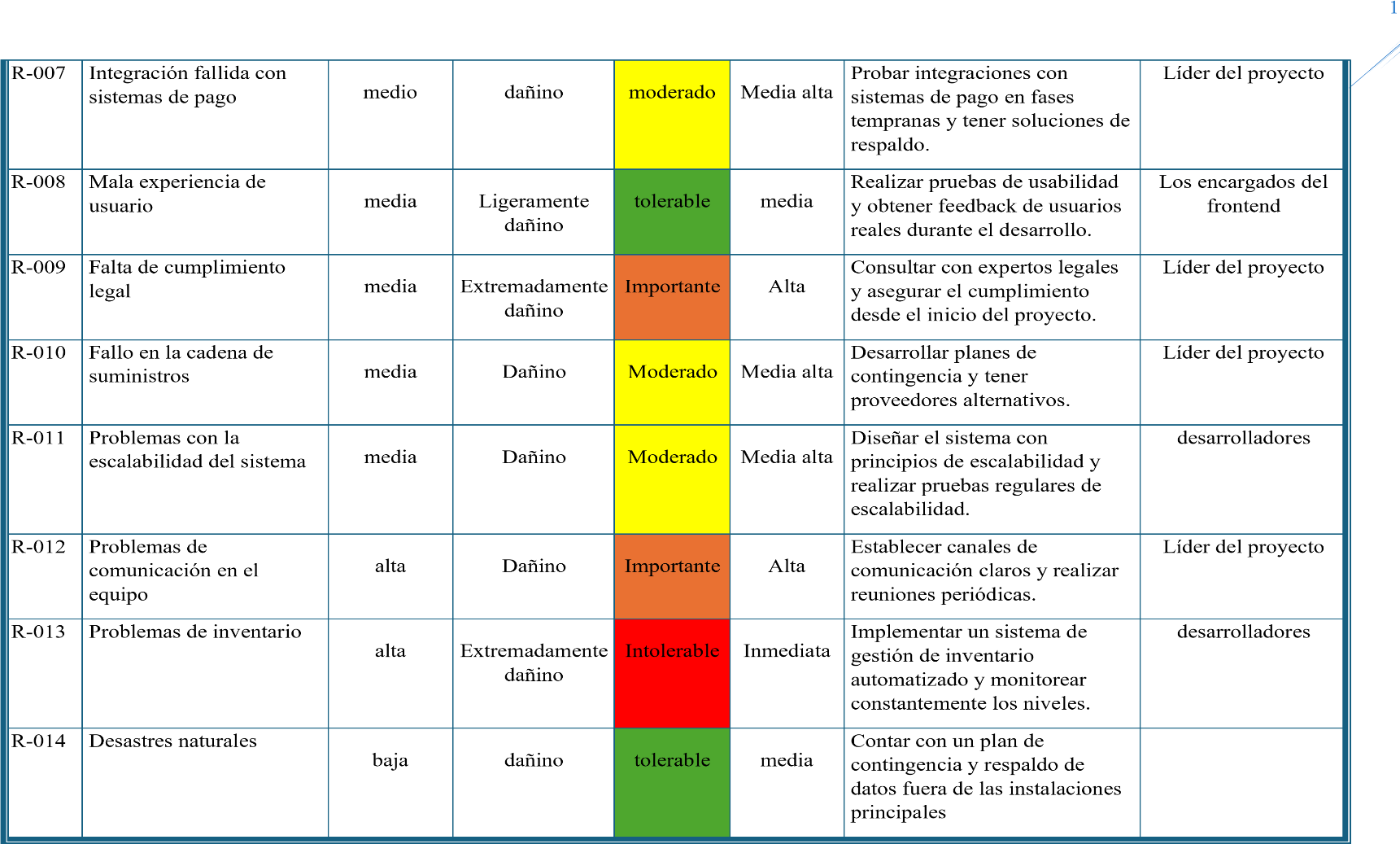
|  |  |
| --- | --- |
| Valoración de la probabilidad | |
| Probabilidad | Descripción |
| Baja | El daño ocurrirá raras veces |
| Media | El daño ocurrirá en algunas ocasiones |
| Alta | El daño ocurrirá siempre o casi siempre |

|  |  |
| --- | --- |
| Valoración de la severidad | |
| Severidad | Descripción |
| Ligeramente dañino | Eventos o situaciones que tiene un impacto mínimo en el proyecto, que no impiden el funcionamiento del software como: errores tipográficos en comentarios del código, problemas menores en la interfaz de usuario, pero no afecta la funcionalidad y pequeñas discrepancias en la documentación. |
| Dañino | Eventos o situaciones que, si afectan al proyecto, pero no de manera critica, no impide el uso del uso del software ni actividades esenciales; como, por ejemplo: errores en funcionalidades secundarias del software, retrasos en el cronograma y problemas de rendimiento en condiciones específicas de uso. |
| Extremadamente dañino | Entos o situaciones que tiene un impacto severo en el proyecto, puede impedir el uso del servicio o afectar actividades criticas del software tales como: fallos en la seguridad que podrían llevar a la exposición de datos privados, fallos en los requerimientos funcionales del software y incumplimientos con los contratos. |

|  |  |
| --- | --- |
| Valor del riesgo | Prioridad |
| Trivial | Baja |
| Tolerable | Media |
| Moderado | Media alta |
| Importante | Alta |
| Intolerable | inmediata |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Valoración de severidad | |
| Valoración de probabilidad | Ligeramente dañino | Dañino | Extremadamente dañino |
| Baja | Trivial | Tolerable | Moderado |
| Media | Tolerable | Moderado | importante |
| Alta | Moderado | Importante | intolerable |





# E licitación de requisitos

## 6.1 Identificación de procesos

* Usuarios (login, registro, roles, perfil).
* Inventario (entradas, salidas, stock).
* Ventas (carrito, factura, pago, gestión).
* Soporte (PQR, calificación, historial, favoritos).

## 6.2 Recolección de la información del software a construir de acuerdo con las necesidades del cliente

## 6.2.1 Elección de la técnica de recolección de la información

Entrevista: Se aplicó una entrevista al dueño y gerente de la tienda, con el fin de comprender de primera mano las necesidades de la empresa, los problemas actuales en la gestión de ventas virtuales y las expectativas frente al nuevo sistema.

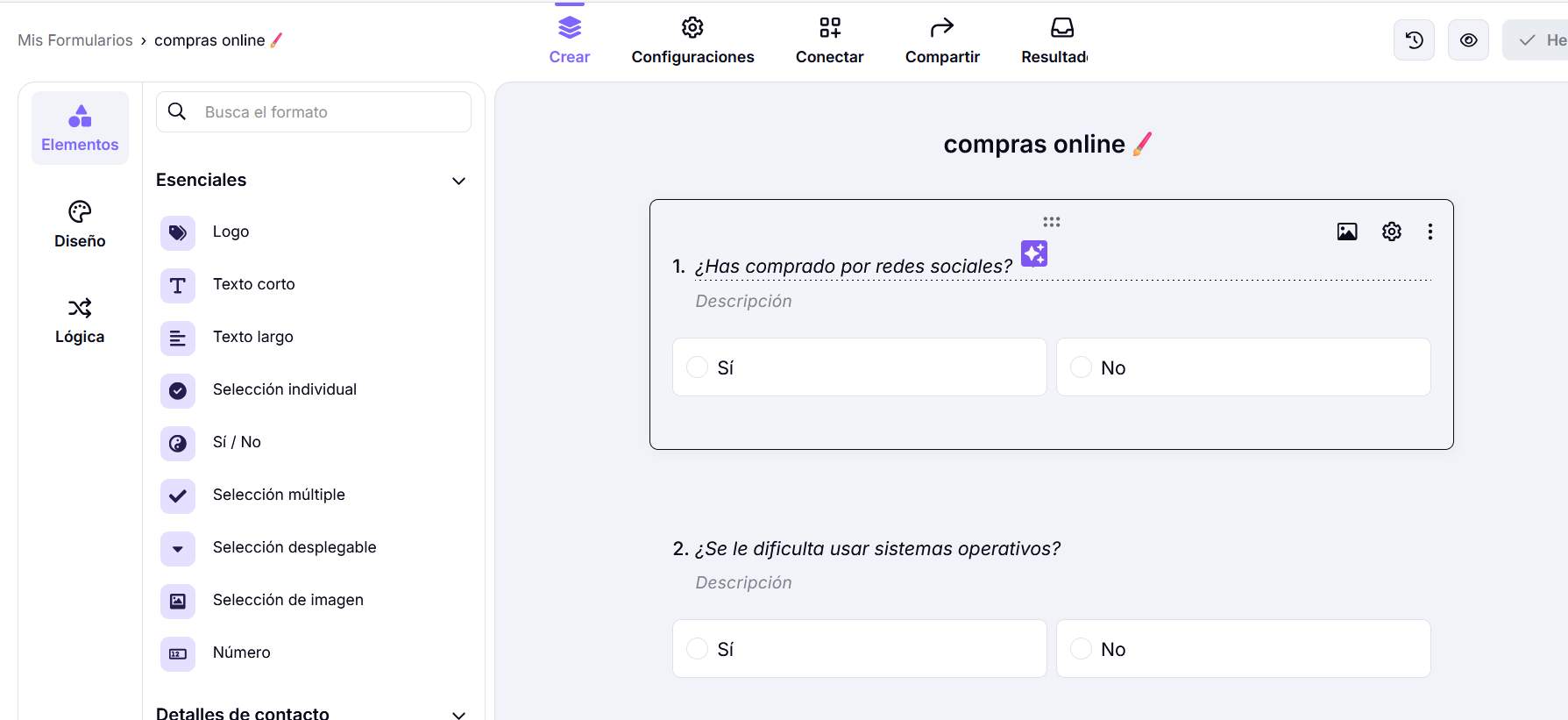
Encuesta: Se aplicaron encuestas a los clientes de la tienda con el fin de conocer sus hábitos de compra, confianza en plataformas digitales y preferencia en canales (aplicación vs redes sociales).

## 6.2.2 Diseño de los formatos, según la técnica o técnicas de recolección de la información seleccionada

Entrevista: preguntas:

* *¿Qué problemas debe solucionar el Sistema?*
* *¿Cómo se resuelven ahora?*
* *¿Cómo los resolvería?*
* *¿Quiénes serán los usuarios del Sistema?*
* *¿Cuál será su nivel de control de PC, dispositivos?*
* *Tienen los usuarios experiencia en este tipo de aplicaciones*
* *Posee software de inventario*
* *Cuáles son las expectativas de usabilidad del Producto*
* *¿Cuáles son las expectativas de formación a usuarios para el uso del Sistema?*
* *¿Qué tipo de ayuda requerirá el usuario (ayuda online, etc…)?*
* *¿Quién en la organización necesita la aplicación?*
* *¿Cuántos tipos de usuarios usarán la aplicación?*
* *¿Cómo valoraría que la solución ha sido un éxito?*
* *¿Cuáles son tus expectativas en cuanto a fiabilidad del sistema?*
* *¿Cuáles son tus expectativas en cuanto a rendimiento del sistema?*
* *¿Cuáles son los requisitos de seguridad?*
* *¿Cuáles son los requisitos de instalación y configuración?*
* *¿Existen requisitos legales que deban soportarse?*
* *¿Existen requisitos de entorno?*
* *¿Existen otro tipo de requisitos?*

Encuesta: para la encuesta nos decidimos por utilizar Google form y con preguntas mas orientadas a los clientes



## 6.2.3 Aplicación de la técnica de recolección de la información

Entrevista:



WhatsApp Audio 2024-06-21 at 2.23.05 PM (1).aac

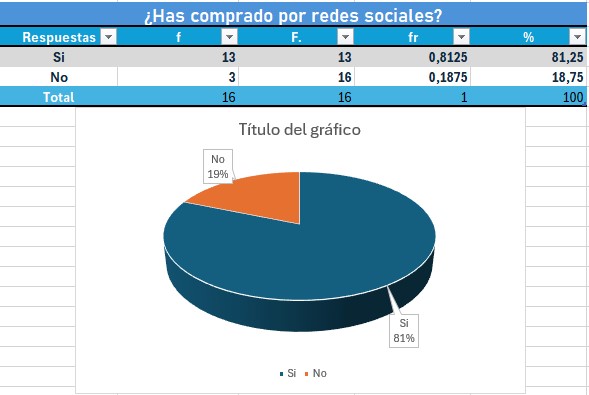
Encuesta:

|  |
| --- |
|  |

## 6.2.4 Organización de la información recolectada

Encuesta: la información obtenida la pasamos un Excel para poder hacer las graficas y tener un mejor orden de la información.

*¿Has comprado por redes sociales?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 81% de los encuestados si compra por redes sociales.

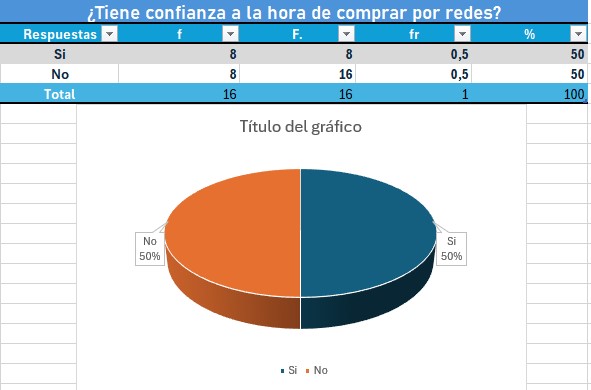
*¿Se le dificulta usar sistemas operativos?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 81% de los encuestados si saben manejar sistemas operativos o aplicaciones.

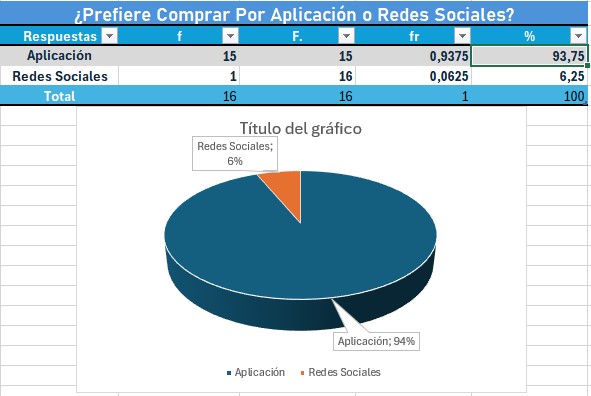
*¿Tiene confianza a la hora de comprar por redes?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que es un 50/50 la mitad si tiene confianza y la otra mitad no tiene confianza a la hora de comprar por redes.

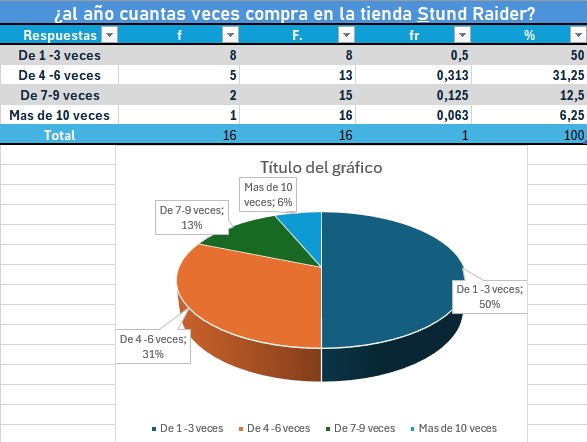
*¿Prefiere comprar por redes sociales o por aplicación?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 94% de los encuestados prefieren comprar por aplicación.

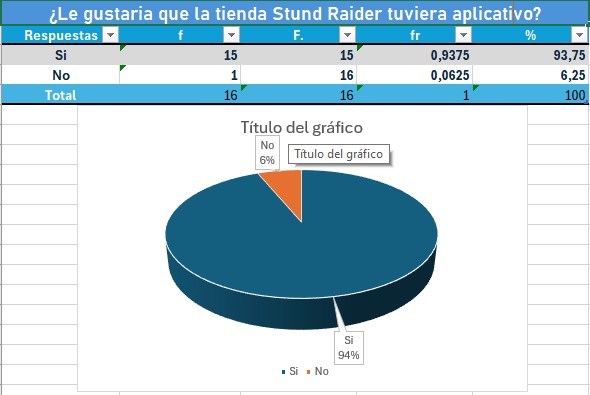
*¿Al año cuantas veces compra en la tienda Stunt Raiders?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 50% de la gente encuestada compra de 1-3 veces, que el 31% de 4-6 veces, el 13% de 7-9 veces y el 6% más de 10 veces.

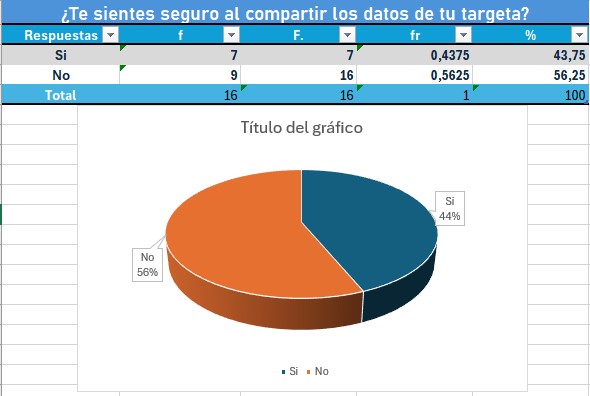
*¿Le gustaría que la tienda Stunt Raiders tuviera aplicativo?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 94% de los encuestados prefieren que la tienda si tenga un aplicativo.

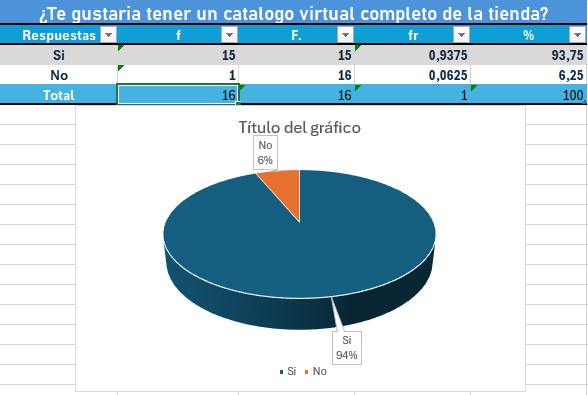
*¿Te sientes seguro al compartir los datos de tu tarjeta?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 56% de los encuestados no se sienten seguros al compartir los datos de su tarjeta.

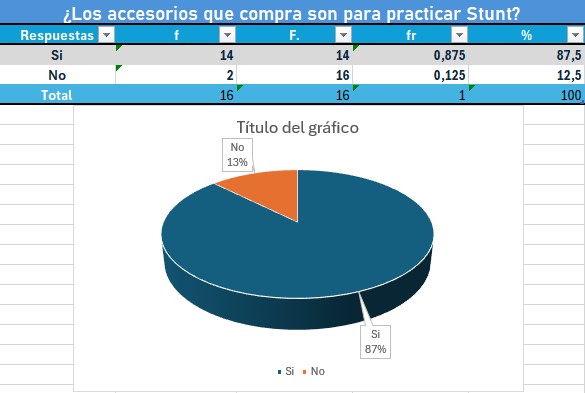
*¿Te gustaría tener un catálogo virtual completo de la tienda?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 94% de los encuestados si les gustaría un catálogo virtual completo de la tienda.

*¿Los accesorios que compra son para practicar stunt?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta de que el 87% de los encuestados si compran accesorios que son para practicar stunt.

*¿Ya nos sigues en redes sociales?*



Análisis de la pregunta:

En esta grafico de la pregunta hecha nos damos cuenta que el 56% de los encuestados si nos siguen redes sociales.

Entrevista:

Identificando perfil de usuario/cliente

Nombre del entrevistado: Daniel torres

Cargo en organización: dueño y gerente tienda Stunt Raiders

Responsabilidades del entrevistado: Gerencia

*Evaluación del problema*

*¿Qué problemas debe solucionar el Sistema?*

La empresa experimenta la necesidad de expandir su presencia en el mercado y sistematizar la toma de pedidos.

*¿Cómo se resuelven ahora?*

Actualmente, contamos con un procedimiento mediante el cual por WhatsApp creo un catálogo de forma de cascada de todos los productos de la tienda el cual no dice si se encuentran disponible o no, así que los clientes tienen que preguntar la disponibilidad e información del producto por mensaje el cual es muy demorado.

*¿Cómo los resolvería?*

A través de un software, he visto que es una herramienta muy útil para llevar el control de mi inventario a de más la app me ayudaría bastante con el tema de las ventas

*¿Quiénes serán los usuarios del Sistema?*

Cualquier persona interesada en el producto, pero en su mayoría gente que practica el deporte.

*¿Cuál será su nivel de control de PC, dispositivos?*

En el local contamos con un portátil sencillo Core i3, y ya en mi casa cuento con un pc muy bueno con cual edito los videos para las redes sociales.

*Tienen los usuarios experiencia en este tipo de aplicaciones*

Hoy en día es muy común ver que muchas personas poseen conocimiento sobre este tipo de software, pero podríamos llevar un control o apoyo para que así puedan ir conociendo la aplicación

*Posee software de inventario*

No actualmente llevo un control con tablas de Excel.

*Cuáles son las expectativas de usabilidad del Producto*

Reducir los tiempos de espera y efectividad a la hora de realizar y despachar los pedidos.

*¿Cuáles son las expectativas de formación a usuarios para el uso del Sistema?*

Creando un video tutorial para enseñarles a los usuarios a usar la aplicación

*¿Qué tipo de ayuda requerirá el usuario (ayuda online, etc…)?*

Primeramente, seria con acompañamiento video y un manual

*¿Quién en la organización necesita la aplicación?*

La empresa como los clientes.

*¿Cuántos tipos de usuarios usarán la aplicación?*

Usuarios interesados em los productos o personas queriendo adaptar sus motos para el deporte.

*¿Cómo valoraría que la solución ha sido un éxito?*

Si las ventas aumentan ya que con la aplicación más personas tendrán conocimiento de la empresa y mejorar los tiempos de entrega.

*¿Cuáles son tus expectativas en cuanto a fiabilidad del sistema?*

Tenemos que garantizar la transparencia en las transacciones y la privacidad de su perfil

*¿Cuáles son tus expectativas en cuanto a rendimiento del sistema?*

Que sea funcional las 24 horas y garantizar una buena base de datos estable y rápida

*¿Cuáles son los requisitos de seguridad?*

Usuario con contraseña para garantizar la privacidad y estarlo monitoreando constantemente.

*¿Cuáles son los requisitos de instalación y configuración?* Android, IOS solo se va a poder por APP

Otros requisitos

*¿Existen requisitos legales que deban soportarse?*

No

*¿Existen requisitos de entorno?*

No

*¿Existen otro tipo de requisitos?*

No

# 6.3. Especificación de requerimientos, de acuerdo con la información recolectada, aplicando estándares de requerimientos.

## 6.3.1 Requerimientos funcionales

Modulo login

|  |  |
| --- | --- |
| Identificación del requerimiento | RFO1 |
| Nombre del requerimiento | Registro de usuario. |
| Características | Los usuarios deberán llenar datos personales. |
| Descripción del requerimiento | El sistema exige tener datos básicos del usuario para llevar un control; en esta sección tendrá que llenar lo requerido por el sistema. |
| Entradas | Nombre completo, correo electrónico, numero móvil, contraseña, fecha de nacimiento, sexo y aceptar términos y condiciones. |
| Condiciones | El sistema confirmara su registro. |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RFO2 |
| Nombre del requerimiento | Iniciar sesión |
| Características | Los usuarios deberán identificarse para acceder a cualquier parte del sistema. |
| Descripción del requerimiento | El usuario después de que se haya registrado procederá al sistema con su cedula, nombre y contraseña. |
| Entradas | Nombre completo, cedula y contraseña. |
| Condiciones | Revisar que esté registrado. |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RFO3 |
| Nombre del requerimiento | Gestión del usuario. |
| Características | Eliminar, actualizar, modificar y consultar |
| Descripción del requerimiento | El administrador podrá eliminar usuarios, modificar información del usuario, actualizar datos. |
| Entradas | Código de usuario. |
| Condiciones | Tiene que estar registrado el usuario y el administrador será el único que podrá gestionar los usuarios. |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RFO4 |
| Nombre del requerimiento | Asignar rol. |
| Características | El administrador asignara al nuevo usuario su rol en el sistema. |
| Descripción del requerimiento | El administrador podrá asignar el rol del usuario el cual dependiendo el perfil tendrá acceso a más funciones del sistema. |
| Entradas | Código del rol y el nombre de rol |
| Condiciones | Solo el administrador puede asignar rol. |
| Prioridad del requerimiento: | Media |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF05 |
| Nombre del requerimiento | Registro de entradas. |
| Características | Productos que ingresan al sistema. |
| Descripción del requerimiento | Registro de los productos con sus características y cantidades. |
| Entradas | Código del producto y fecha. |
| Condiciones | Perfil del administrador y vendedor. |
| Prioridad del requerimiento: | Media |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF06 |
| Nombre del requerimiento | Registro de salidas. |
| Características | Productos que salen del sistema. |
| Descripción del requerimiento | Registro de salida de productos con cantidades y código del producto. |
| Entradas | Código del producto y fecha |
| Condiciones | Perfil del administrador y vendedor |
| Prioridad del requerimiento: | Media |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF07 |
| Nombre del requerimiento | Gestión de productos. |
| Características | Mover, eliminar, actualizar, modificar. |
| Descripción del requerimiento | Mover, eliminar, actualizar y modificar, orientado a objetos serán las funciones que realizará este módulo. |
| Entradas | Código producto, características del producto, nombre del producto y fecha de entrada. |
| Condiciones | El producto tiene que estar registrado con todos los datos requeridos. |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF08 |
| Nombre del requerimiento | Stock |
| Características | Cantidades disponibles |
| Descripción del requerimiento | El sistema, al registrar la salida de los productos lo restará con los productos de entrada para mostrar cantidades disponibles; igualmente al haber pocas cantidades dar una alerta. |
| Entradas | Código producto y cantidades. |
| Condiciones | Cantidades disponibles. |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF09 |
| Nombre del requerimiento | Quejas y reclamos. |
| Características | Verificación de reclamos e inconformidades. |
| Descripción del requerimiento | El usuario tendrá acceso desde la página principal a una sección en el cual podrá comentar sus inquietudes y reclamos sobre la página o algún producto. |
| Entradas | Inicio de sección, ingreso sección de quejas y reclamos. |
| Condiciones | A ver interactuado con la página web. |
| Prioridad del requerimiento: | Baja |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF10 |
| Nombre del requerimiento | Calificación del cliente |
| Características | Calificación de la página por medio de estrellas. |
| Descripción del requerimiento | El usuario podrá representar su experiencia con la página web con un sistema de 5 estrellas donde tendrán un valor de 1(inconforme) hasta 5(excepcional). |
| Entradas | Inicio de sesión, ingreso sección de quejas y reclamos. |
| Condiciones | A ver interactuado con la página web. |
| Prioridad del requerimiento: | Baja |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF11 |
| Nombre del requerimiento | Verificación de quejas y peticiones. |
| Características | Verificación del administrador. |
| Descripción del requerimiento | El sistema le mostrará al gerente los mensajes, quejas y peticiones de los usuarios. |
| Entradas | Inicio de sesión, ingreso sesión de quejas y reclamos, nombre del usuario, fecha y hora de la queja o petición. |
| Condiciones | El administrador será el único que lo podrá ver. |
| Prioridad del requerimiento: | Baja |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF12 |
| Nombre del requerimiento | Perfil personal |
| Características | Datos personales del usuario |
| Descripción del requerimiento | El usuario tendrá que estar registrado, luego podrá ingresar a su perfil desde la página principal que lo enviará a una sección donde encontrará los datos personales que diligenció en el formulario de registro, allí también podrá editarlos por si necesita actualizarlos. |
| Entradas | Documento de identidad, foto, edad, sexo, descripción |
| Condiciones | A ver echo el proceso de registro con la información pertinente. |
| Prioridad del requerimiento: | Media |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF13 |
| Nombre del requerimiento | Historial del cliente |
| Características | Información completa de movimientos y productos favoritos del usuario. |
| Descripción del requerimiento | En este apartado se mostrará una lista de todas las compras que se han realizado mostrando nombre, código, descripción y el total que pagó por ese producto, productos favoritos aparecerán con un símbolo de corazón que se puede seleccionar de cada producto, productos que están en pedido y la fecha de compra. |
| Entradas | Números de facturas, fechas y horas, código de los productos, descripción y precio. |
| Condiciones | El usuario tendría que haber realizado una compra para poder ver el historial. |
| Prioridad del requerimiento: | Media |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF14 |
| Nombre del requerimiento | Catálogo de productos |
| Características | Exhibición de productos |
| Descripción del requerimiento | Menú principal donde se mostrarán todos los productos cargados al sistema ordenados por categorías y donde se mostrará el nombre y la descripción de cada producto. |
| Entradas | Código del producto y descripción del producto |
| Condiciones | Estar registrado |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF15 |
| Nombre del requerimiento | Registrar venta (carrito de compras). |
| Características | Registro de producto que desea comprar. |
| Descripción del requerimiento | Registro de cada producto con sus cantidades el cual se podrá modificar antes de la confirmación. |
| Entradas | Código del producto y cantidades. |
| Condiciones | Cantidades suficientes (stock) |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF16 |
| Nombre del requerimiento | Generar factura |
| Características | El sistema genera una factura de pago. |
| Descripción del requerimiento | El sistema genera una factura con el precio total de los productos seleccionados, un numero de factura, fecha de la compra y datos del comprar. |
| Entradas | Nombre de usuario, código productos, cantidades y fecha |
| Condiciones | Cantidades de productos disponibles |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF17 |
| Nombre del requerimiento | Comprobante de pago |
| Características | Comprobante de pago (Baucher) de transacción exitosa. |
| Descripción del requerimiento | El sistema validará la transacción del dinero y posteriormente emitirá un baucher como comprobante del pago. |
| Entradas | Número de factura, precio total, transferencia de dinero |
| Condiciones | Transacción con las plataformas autorizadas |
| Prioridad del requerimiento: | Alta |

|  |  |
| --- | --- |
| Identificación del requerimiento | RF18 |
| Nombre del requerimiento | Gestionar ventas. |
| Características | agregar, eliminar, actualizar, modificar. |
| Descripción del requerimiento | agregar, eliminar, actualizar y modificar los movimientos realizados en las ventas. |
| Entradas | Nombre, fecha, código producto, numero de factura, comprobante de pago |
| Condiciones | Perfil administrador, el sistema generado y almacenado el informe diario. |
| Prioridad del requerimiento: | Alta |

## 6.3.2 Requerimientos no funcionales

falta

## 6.3.3 Requerimientos normativos:

Si tu software e-commerce va a operar en Colombia, los requerimientos normativos se rigen por un conjunto de leyes nacionales que protegen al consumidor, regulan los datos personales y garantizan la validez de las transacciones.

Un e-commerce en Colombia debe cumplir con Ley 1581 (datos personales), Ley 1480 (consumidor) y ley licenciamiento de software 603 de 2000.

Ley 1581(habeas data):

Son reglas que toda empresa debe cumplir al tratar datos personales:

* Legalidad: el tratamiento de datos debe hacerse conforme a la ley.
* Finalidad: solo se pueden usar los datos para los fines específicos informados al titular.
* Libertad: se requiere autorización previa, expresa e informada del titular.
* Veracidad o calidad: los datos deben ser veraces, completos y actualizados.
* Transparencia: el titular debe poder acceder a su información en cualquier momento.
* Acceso y circulación restringida: solo personas autorizadas pueden acceder a los datos.
* Seguridad: deben protegerse contra pérdida, robo, acceso no autorizado o fraude.
* Confidencialidad: los datos deben ser tratados de forma privada.

Un e-commerce en Colombia debe:

* Solicitar consentimiento expreso antes de recolectar datos (ej. casilla de “Acepto la política de privacidad”).
* Crear y publicar una Política de Tratamiento de Datos Personales accesible en la web.
* Registrar sus bases de datos ante el Registro Nacional de Bases de Datos (RNBD) administrado por la SIC.
* Garantizar mecanismos de seguridad (encriptación, HTTPS, contraseñas seguras).
* Atender consultas y reclamos de usuarios sobre sus datos en un plazo máximo de 10 días hábiles.
* Conservar la evidencia de la autorización otorgada por los titulares.

Ley 480 (consumidor):

Un cliente de un e-commerce en Colombia tiene derecho a:

* Información clara: precios, impuestos, costos de envío, condiciones del producto.
* Seguridad y calidad: productos en buen estado, sin riesgos indebidos.
* Protección contra publicidad engañosa.
* Garantía mínima legal: según el tipo de producto o servicio.
* Derecho al retracto: devolver el producto en máximo 5 días hábiles desde la entrega (aplica para compras online).
* Atención y respuesta a quejas, reclamos y solicitudes.

Los proveedores (tiendas online) deben:

* Publicar precios totales (incluyendo IVA y costos adicionales).
* Tener políticas de devolución y retracto claras y visibles en la web.
* Responder en un máximo de 15 días hábiles a reclamos.
* Garantizar la entrega del producto en el tiempo ofrecido.
* Respetar las promociones y descuentos publicados.
* Informar si un producto es usado, reacondicionado o de segunda mano.

Ley 603 del 2000 (licencias de Software, propiedad intelectual y derechos de autor):

La Ley 603 de 2000 en Colombia no establece un marco legal general para el software, sino que modifica el artículo 47 de la Ley 222 de 1995 para exigir que en el informe de gestión de las sociedades comerciales se incluya el estado de cumplimiento de las normas de propiedad intelectual y derechos de autor, incluyendo el uso de software legal. Esta ley faculta a las autoridades tributarias, como la DIAN, para verificar este cumplimiento y evitar la evasión de tributos.

Aspectos clave de la Ley 603 de 2000:

Obligación de informar:

Las empresas deben incluir en sus informes de gestión información sobre el cumplimiento de las normas de derechos de autor y propiedad intelectual.

Verificación por autoridades:

Las autoridades tributarias (como la DIAN) pueden verificar que las empresas cumplan con las licencias de software para asegurar el cumplimiento de los derechos de autor y la no evasión de impuestos.

Consecuencias del incumplimiento:

Usar software sin licencia puede acarrear:

* Sanciones administrativas de hasta 200 salarios mínimos legales mensuales vigentes (SMMLV) para los administradores de la empresa.
* Delitos como la defraudación de derechos patrimoniales de autor o la violación a mecanismos de protección de dichos derechos, que implican penas de prisión y multas considerables.
* ¿Qué se entiende por software legal?
* El software es un bien intelectual protegido por derechos de autor desde su creación.
* El uso de software sin licencia, o la violación de los términos de las licencias de uso, constituye una práctica ilegal.
* Para que el software sea legal, las empresas deben contar con las licencias de uso correspondientes y respetar los límites de estas.

## 6.3.4 Propuesta técnica:

Propuesta Técnica

1. Tecnologías a utilizar

El sistema se implementará en PHP, aprovechando su flexibilidad y amplia adopción en el desarrollo de aplicaciones web. Se trabajará bajo un framework propio desarrollado manualmente, diseñado de acuerdo con las necesidades específicas del proyecto, lo que permite mayor control sobre las funcionalidades y la estructura del código.  
La base de datos será gestionada con MySQL, por su rendimiento, estabilidad y compatibilidad con PHP.  
Para el control de versiones se empleará Git, con repositorios en GitHub y gestión local a través de Git Desktop, lo que garantiza trazabilidad, trabajo colaborativo y respaldo del código.

2. Arquitectura

La aplicación se desarrollará bajo el patrón arquitectónico MVC (Modelo – Vista – Controlador), exigido por el SENA y considerado como uno de los más utilizados en la programación orientada a objetos.  
Este patrón permitirá separar la lógica de negocio, la interfaz de usuario y el control de flujo, lo que facilita la mantenibilidad, escalabilidad y reutilización del código.

3. Herramientas para el desarrollo y despliegue

Para el desarrollo se utilizará el editor de código Visual Studio Code, por su ligereza, flexibilidad y soporte a múltiples extensiones que mejoran la productividad del programador.  
En el entorno local, se implementará XAMPP como servidor web para la gestión de Apache, MySQL y PHP, simulando el ambiente de producción.  
En cuanto al despliegue, se utilizará el servicio de hosting Infinity Free, el cual permite alojar aplicaciones web PHP con soporte para bases de datos MySQL, garantizando disponibilidad y acceso desde internet.

4. Buenas prácticas

El proyecto aplicará buenas prácticas de desarrollo para asegurar la calidad y seguridad del sistema:

Implementación de HTTPS en la capa de comunicación para garantizar la confidencialidad de los datos transmitidos.

Encriptación de contraseñas mediante la función SHA-256, con el fin de proteger la información sensible de los usuarios.

Documentación del código durante el proceso de desarrollo, facilitando el entendimiento, mantenibilidad y futuras mejoras del software.

5. Plan técnico de implementación

Se adoptará la metodología Scrum, que promueve el desarrollo ágil y colaborativo del software.  
El proyecto se organizará en sprints semanales, donde se realizarán revisiones y entregas parciales para evaluar el avance del desarrollo, identificar posibles mejoras y asegurar el cumplimiento de los objetivos establecidos.  
Este enfoque permitirá adaptarse a cambios en los requerimientos y garantizar un producto final alineado con las expectativas del cliente.

# Reglas del negocio

1. RN-01: Todo usuario debe registrarse con datos válidos (nombre completo, correo, número móvil, Cedula de ciudadanía) para interactuar con el sistema.

2. RN-02: El correo electrónico registrado debe ser único en la base de datos.

3. RN-03: Solo el administrador podrá modificar, actualizar o eliminar cuentas de usuario.

4.RN-04: Los roles del sistema serán: Cliente, Administrador y Vendedor, con permisos diferenciados.

5. RN-05: Solo los usuarios con rol de administrador pueden asignar o modificar roles.

6. RN-06: Todo producto debe estar registrado con un código único, nombre, descripción, precio y cantidad disponible.

7. RN-07: No se permitirá registrar una venta si el producto no cuenta con stock suficiente.

8. RN-08: Las entradas y salidas de inventario deben registrarse con fecha, cantidad y responsable de la operación.

9. RN-9: Un pedido sólo se confirmará si el sistema valida stock disponible y método de pago autorizado.

10. RN-10: El cliente podrá modificar su carrito antes de la confirmación final del pedido.

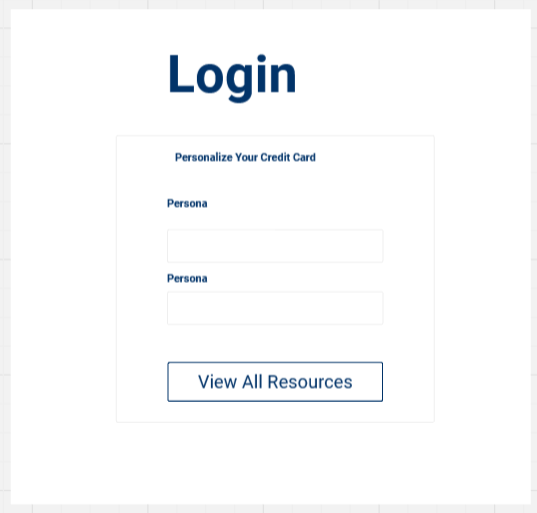
11. RN-11: El sistema generará automáticamente la factura con fecha, código de pedido, datos del cliente y valor total.

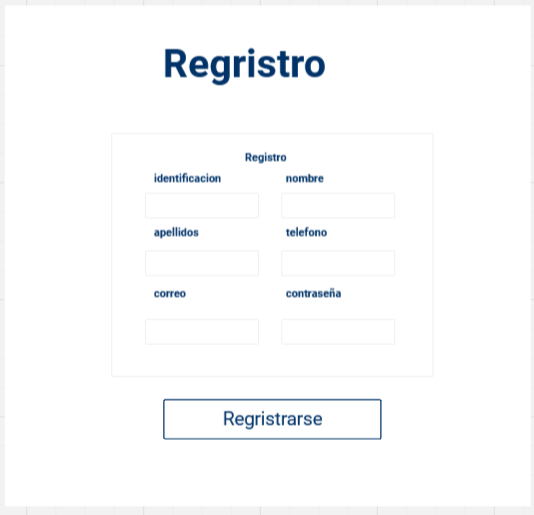
12. RN-12: Las transacciones deben confirmarse mediante plataformas autorizadas como Nequi, Daviplata o PayPal.

13. RN-13: Solo los usuarios registrados podrán generar solicitudes de PQR.

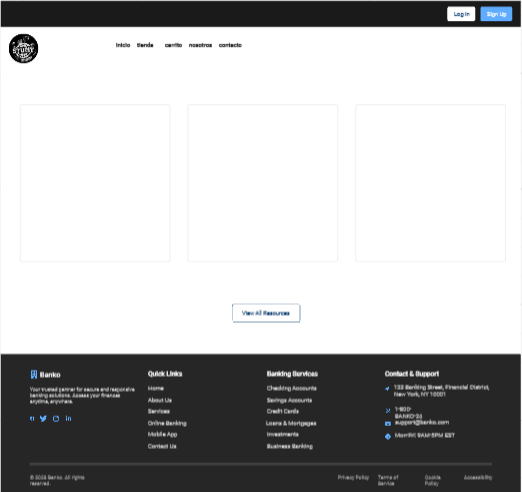
# Análisis de la especificación de requisitos del software

## 8.1 Alternativas de solución (prototipo o prototipos del sistema, mockups)

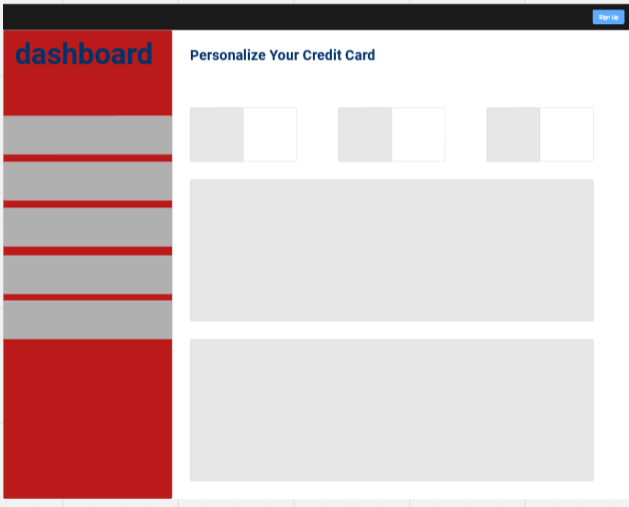




Página principal



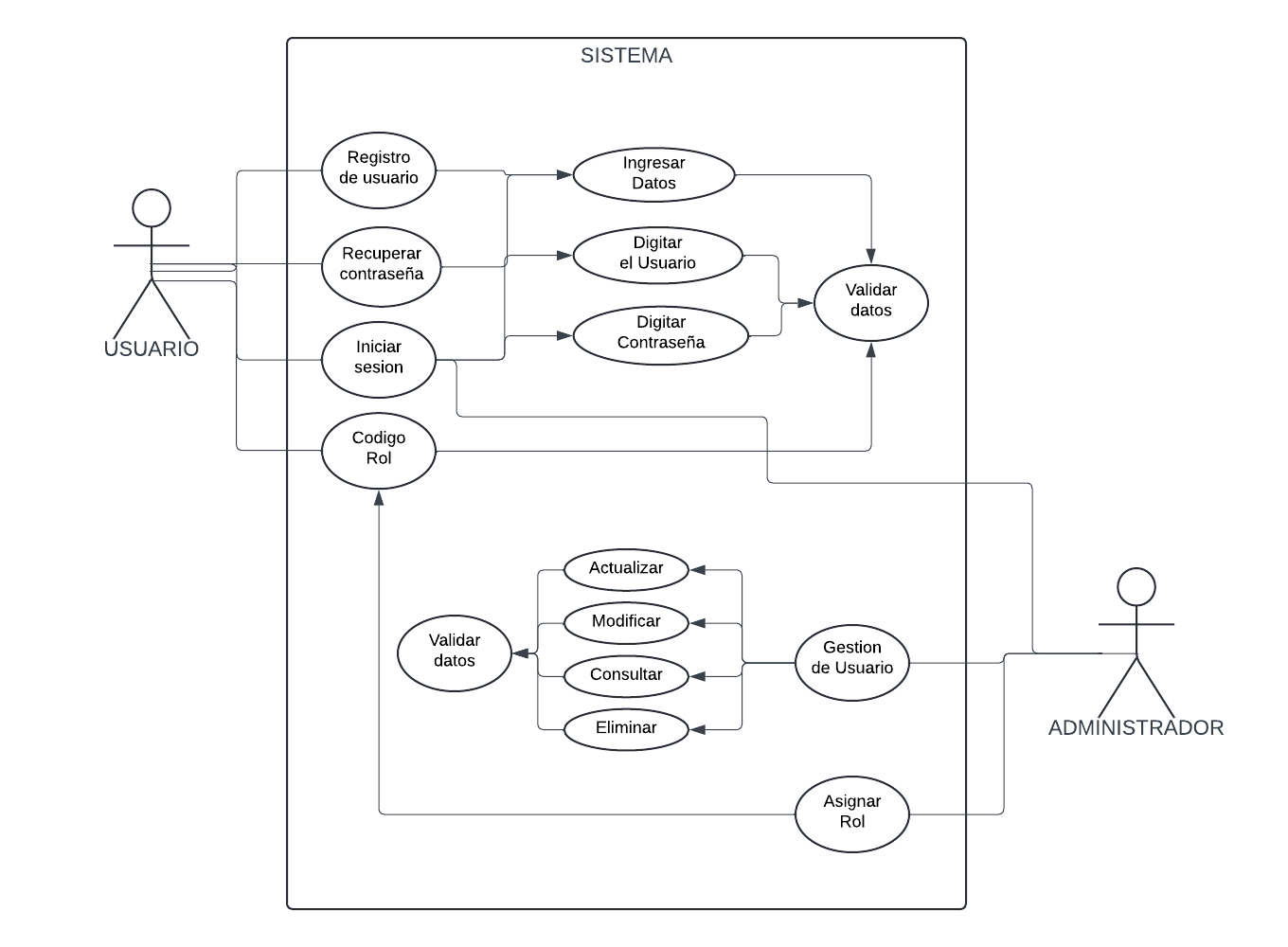
Dashboard

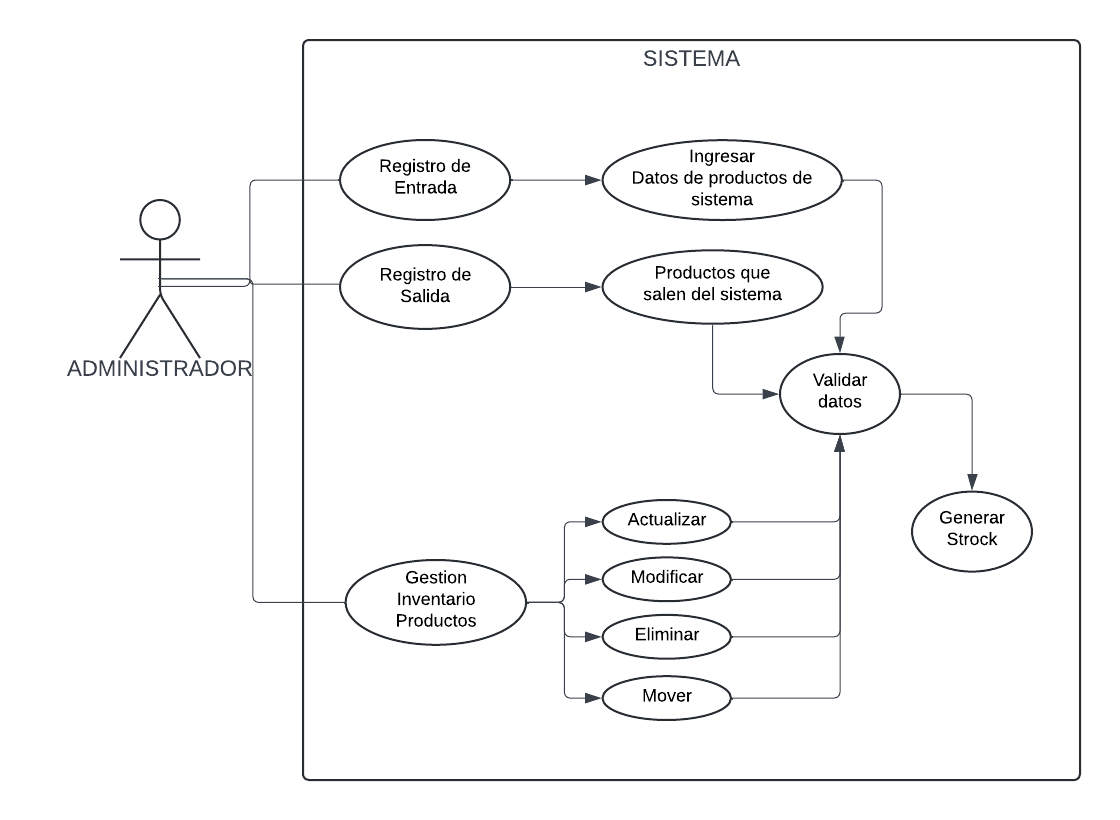


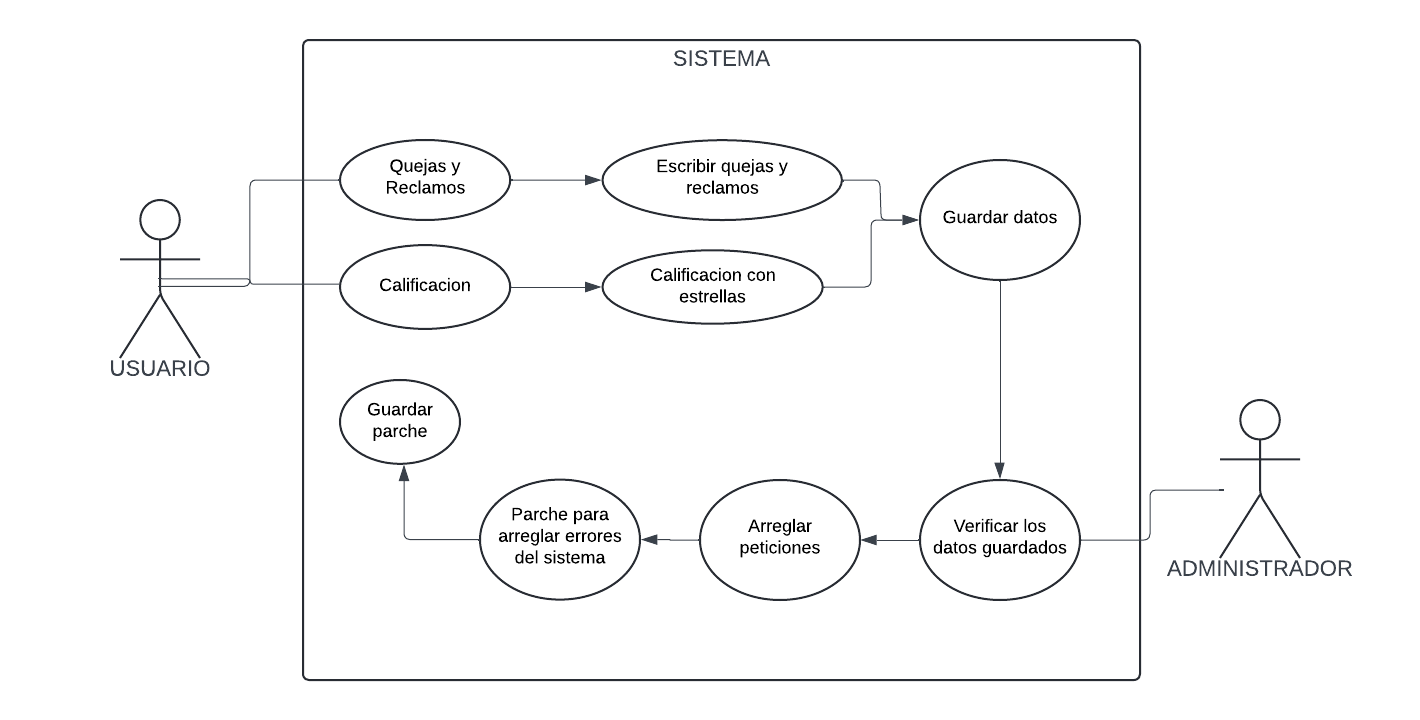
Plantilla estándar para data tables

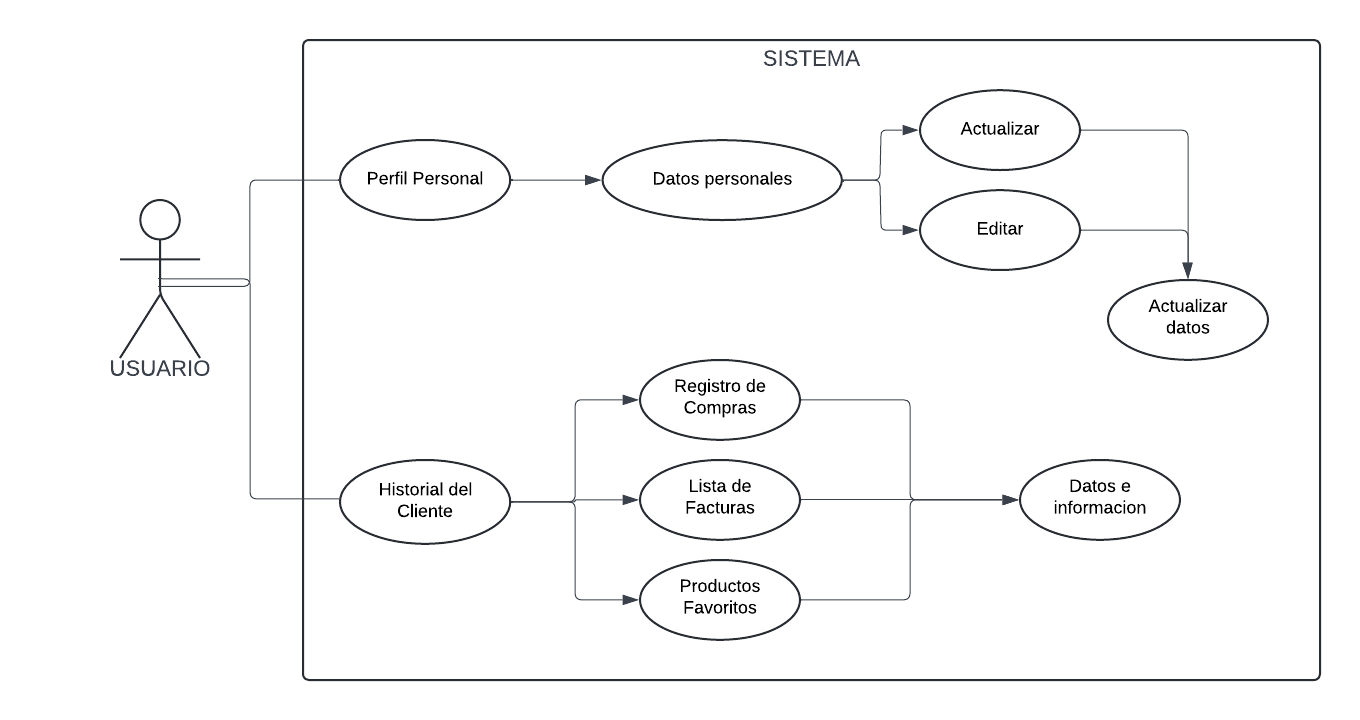


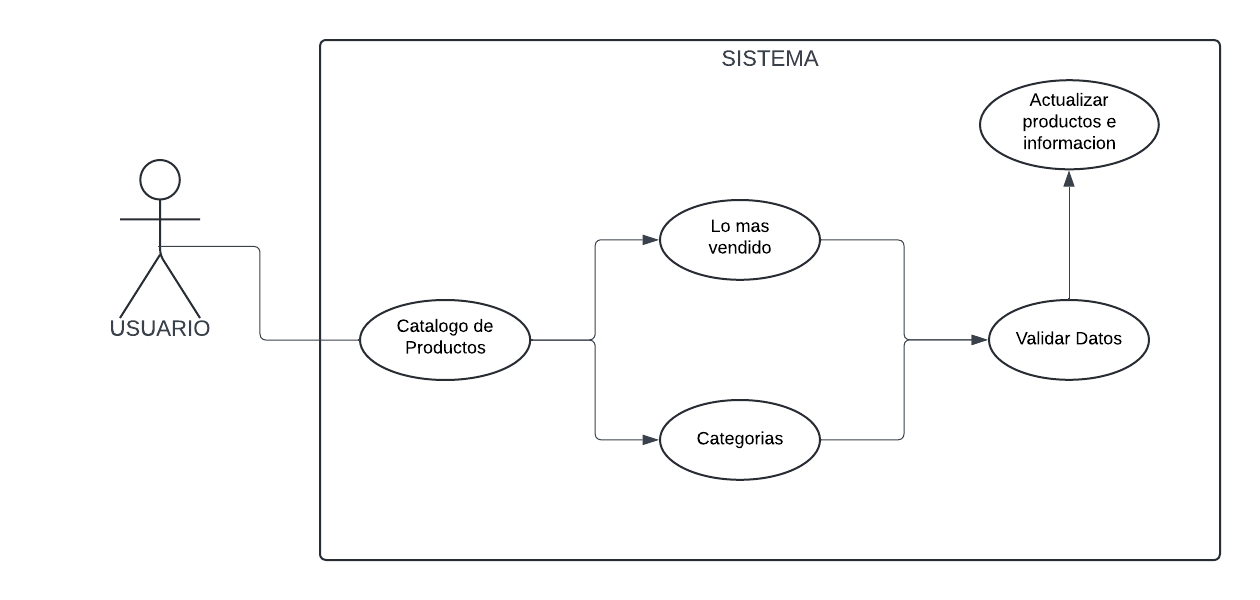
## 8.2 Diagrama de casos de uso y extensibilidad de estos

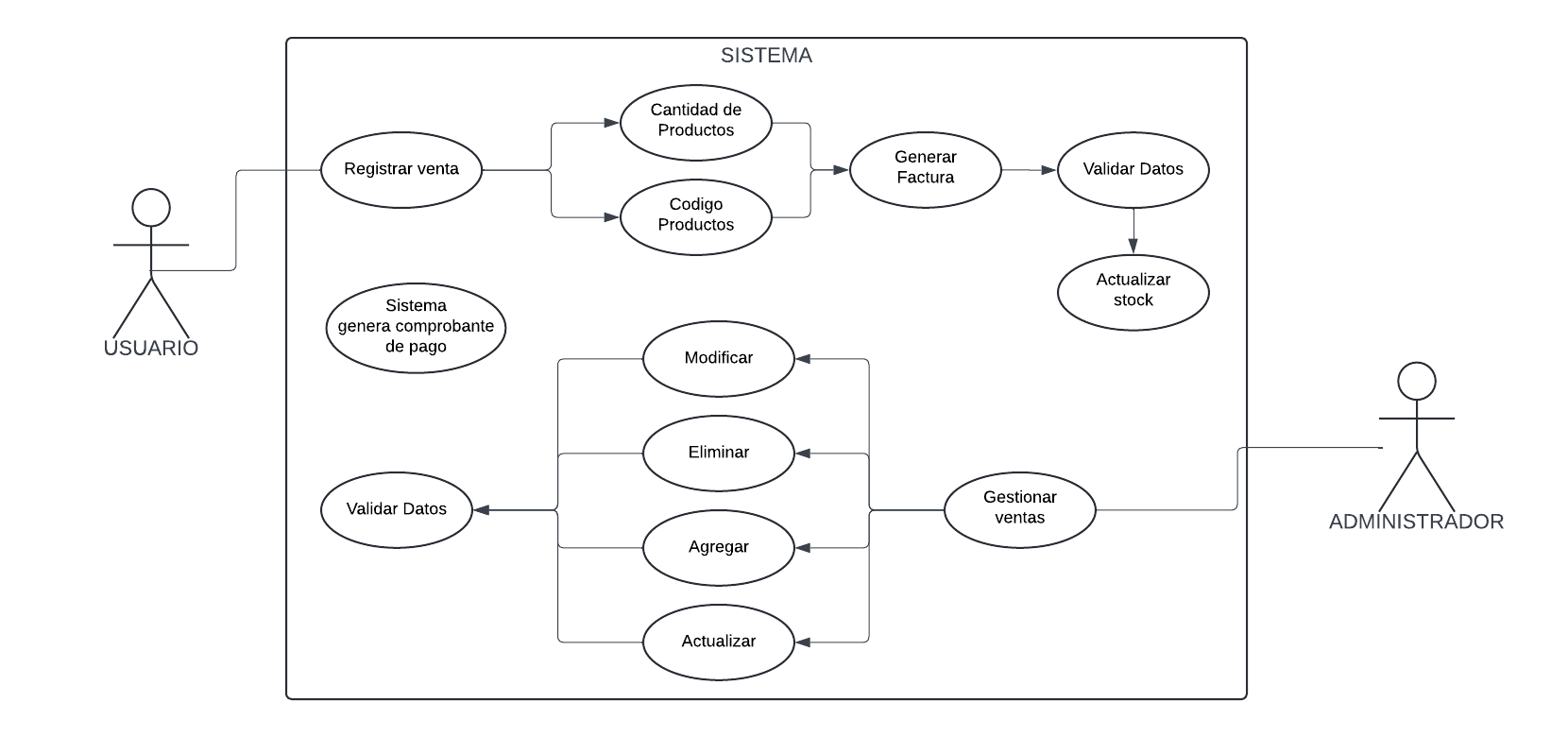












## 8.3 Diagramas de actividades y secuencias

## 8.4 Construir el modelo de dominio del sistema (diagrama de clases)

## 8.5 Elaborar el modelo entidad relación

# Diseño de la solución del software de acuerdo con los procedimientos y requisitos técnicos

## 9.1 Arquitectura el software (Diagrama de componentes) y patrones de diseño de software

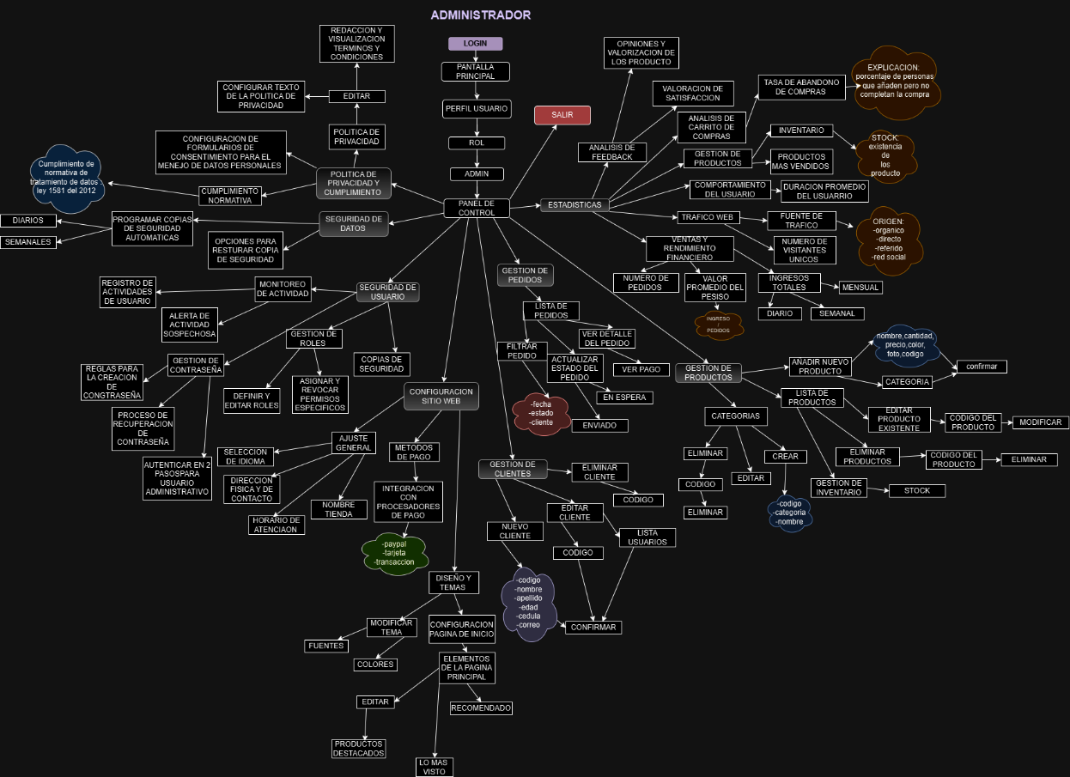
## 9.2 Diagrama de despliegue

## 9.3 Diseño front-end – interfaces graficas de usuario (html, css y jv)

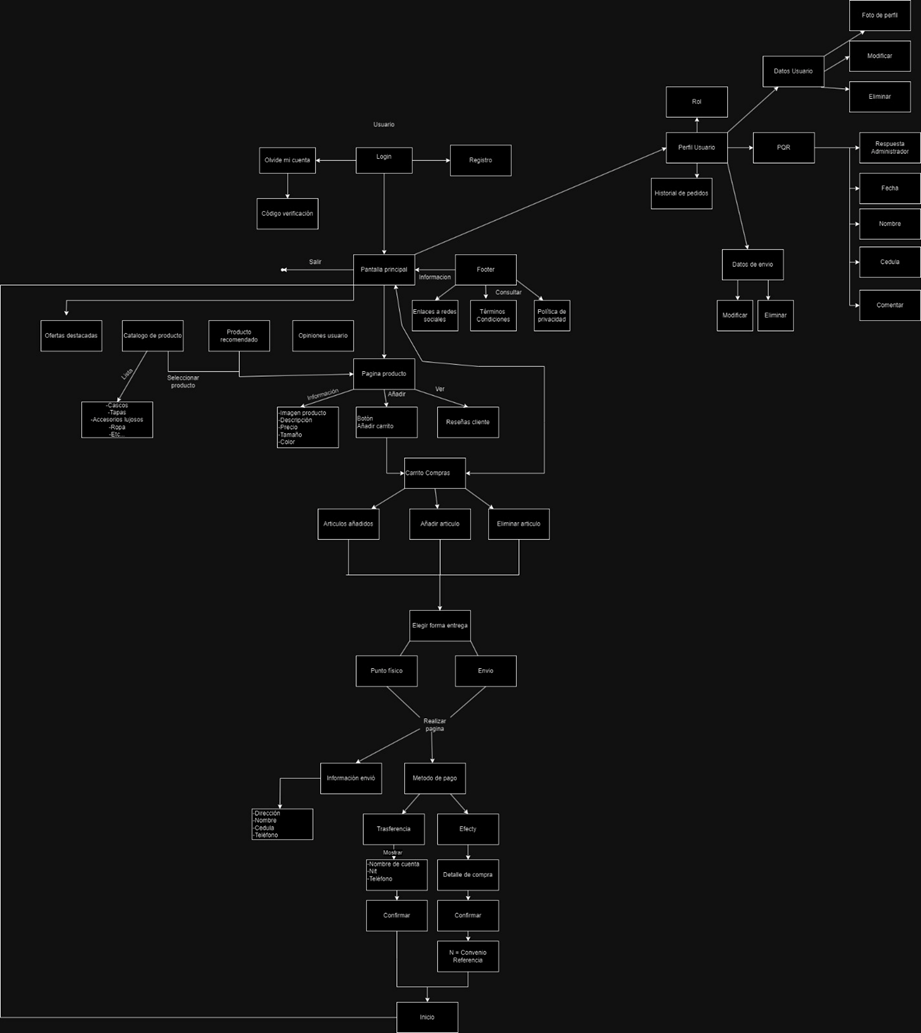
## 9.4 interfaces graficas de usuario móviles

## 9.5 Mapa de navegación de la aplicación

Administrador



Usuario



## 9.6 Determinar tipos de bases de datos.

1. Bases de Datos Relacionales (SQL)

Son aquellas que organizan la información en tablas (filas y columnas), y establecen relaciones entre ellas mediante claves primarias y foráneas.

Modelo: Tablas estructuradas.

Lenguaje: SQL.

Estructura de datos: Muy definida y rígida, con esquemas fijos.

Ejemplos de SGBD: MySQL, PostgreSQL, Oracle, SQL Server.

Ventajas

Ideales cuando los datos tienen relaciones claras.

Integridad y consistencia de datos (ACID).

Fácil de usar para consultas complejas con JOINs.

Desventajas

Poco flexibles al cambiar la estructura (agregar columnas, modificar esquemas).

Menos eficientes cuando los datos son masivos y poco estructurados.

2. Bases de Datos No Relacionales (NoSQL)

Son más flexibles, no usan tablas rígidas sino otros modelos (documentos, grafos, pares c lave-valor, columnas).

Modelo: Dependiendo del tipo:

Documentos (JSON, BSON) → MongoDB.

Grafos → Neo4j.

Clave-valor → Redis.

Columnas → Cassandra.

Lenguaje: No usan SQL estándar, cada motor tiene su propia sintaxis o API.

Estructura de datos: Flexible y dinámica, sin necesidad de esquemas rígidos.

Ventajas

Alta escalabilidad (Big Data, aplicaciones distribuidas).

Flexibilidad en el almacenamiento (no necesitan esquemas fijos).

Rendimiento superior en consultas masivas o datos poco estructurados.

Desventajas

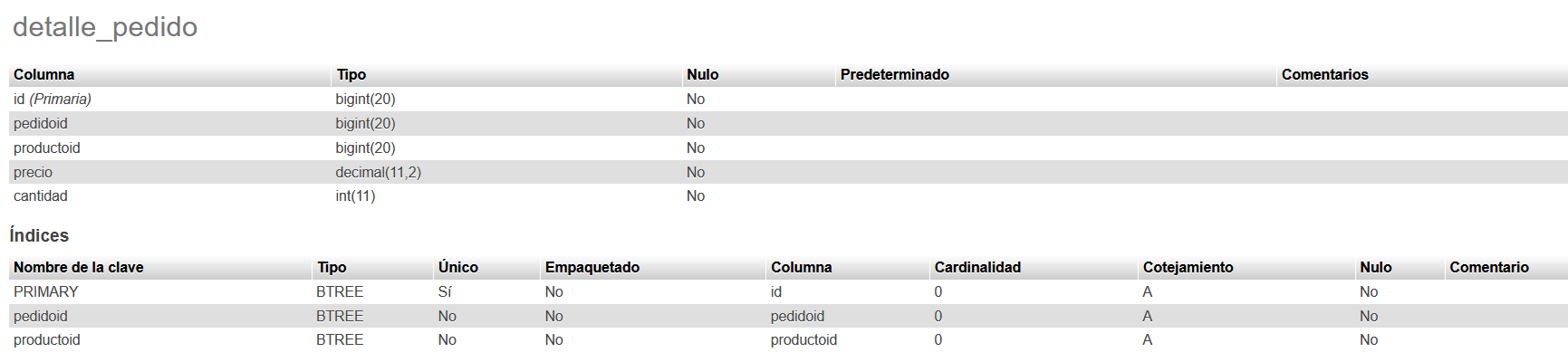
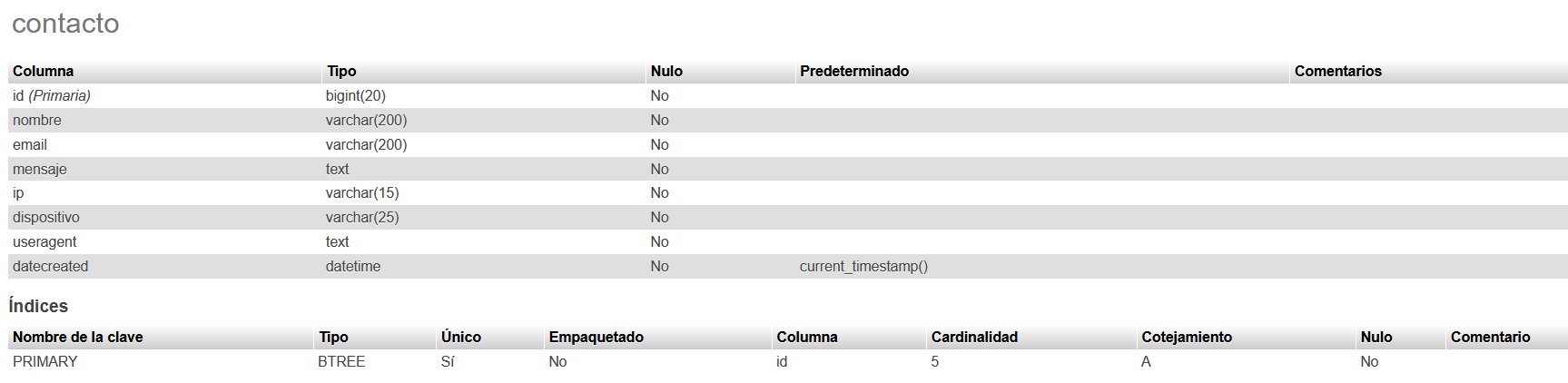
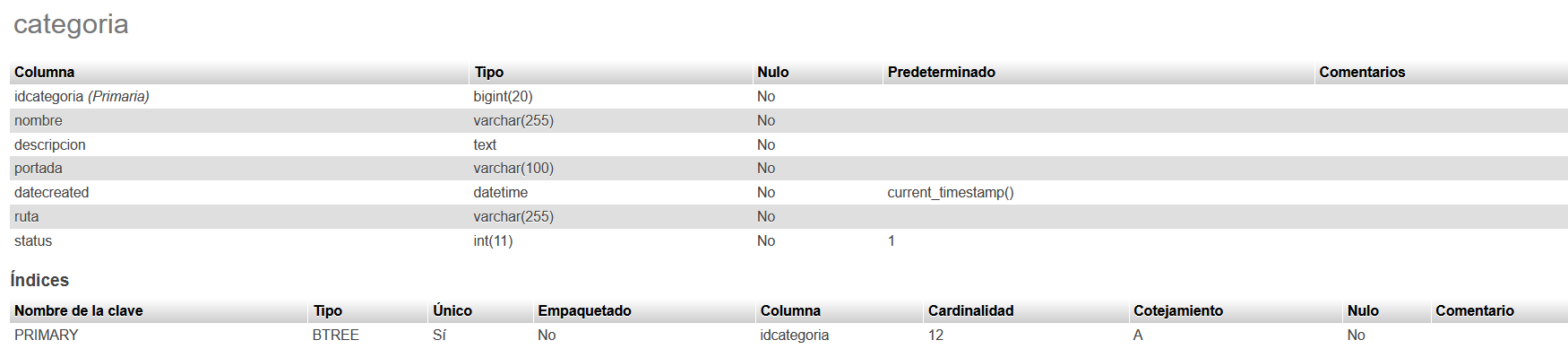
No siempre cumplen totalmente con las propiedades ACID (usan BASE).

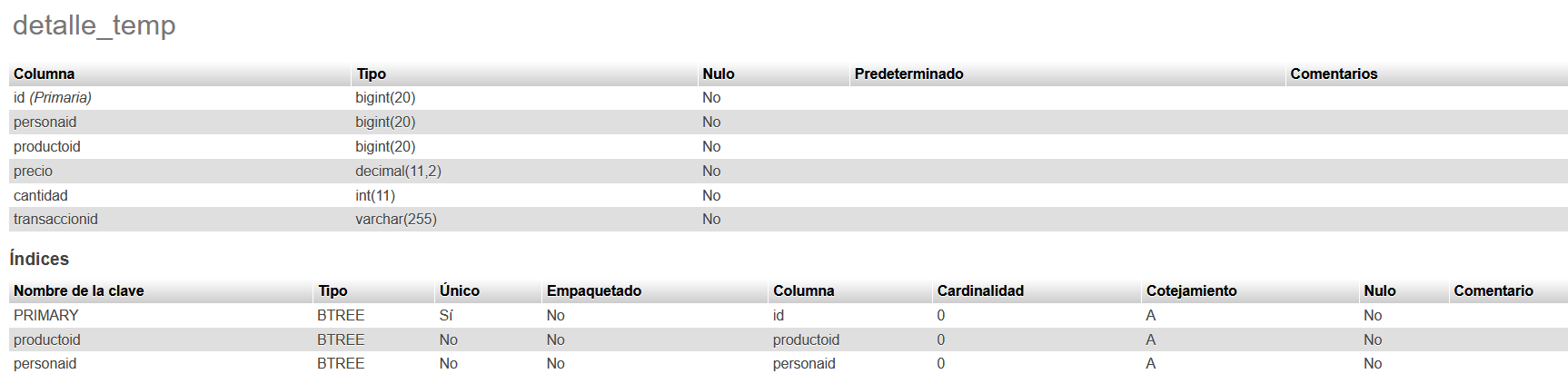
Menos adecuadas para relaciones complejas entre datos.

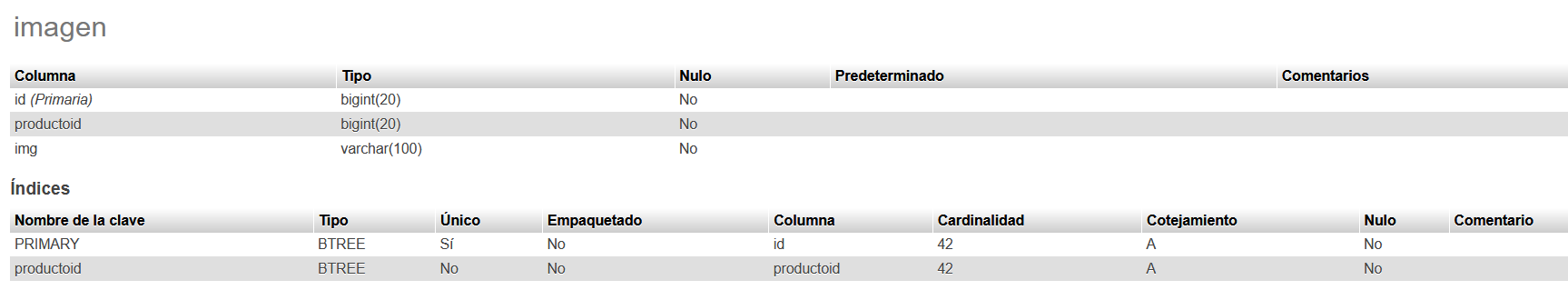
Lenguaje de consultas menos estandarizado.

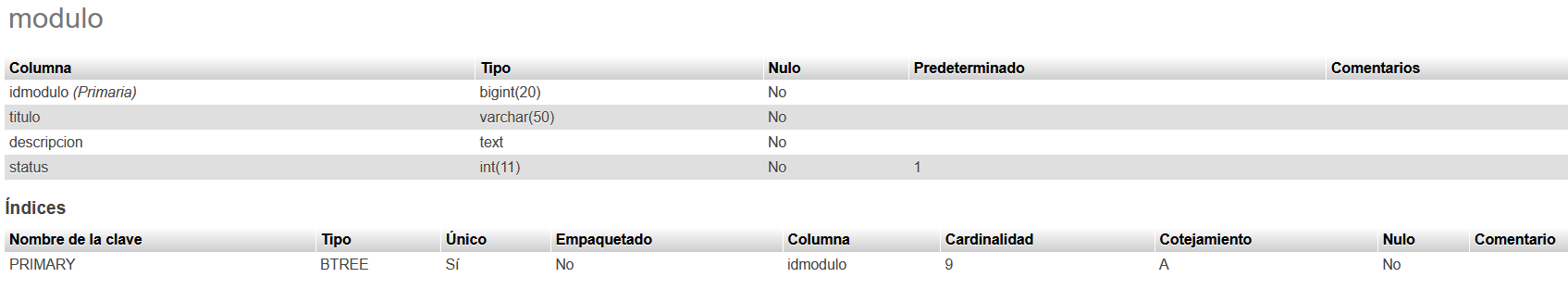
# 9.7 Modelo de datos diagrama Er(normalización base de datos (forma normal)).

## 9.8 Diccionario de datos

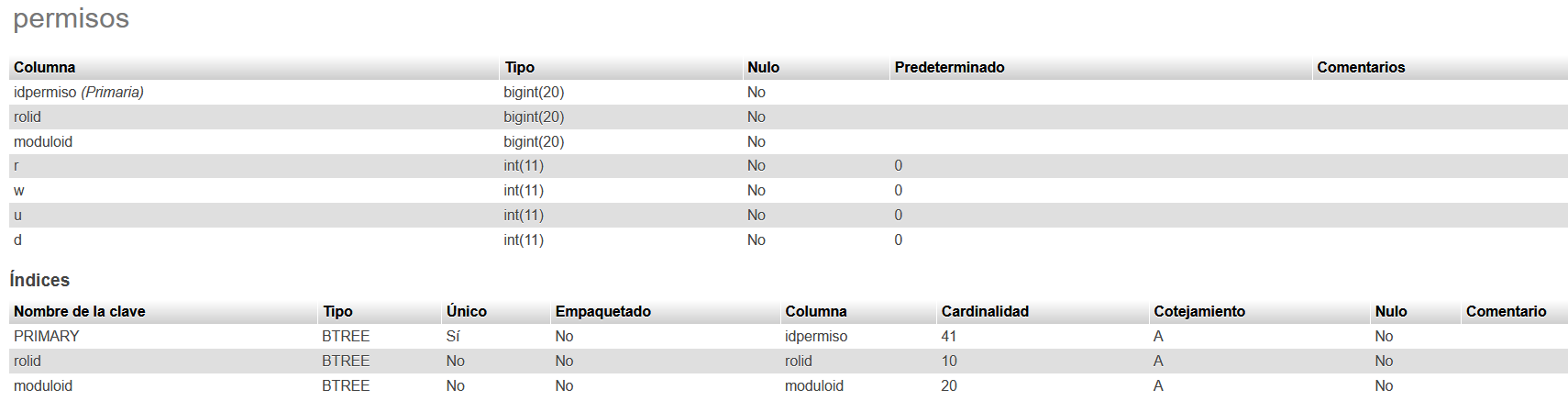


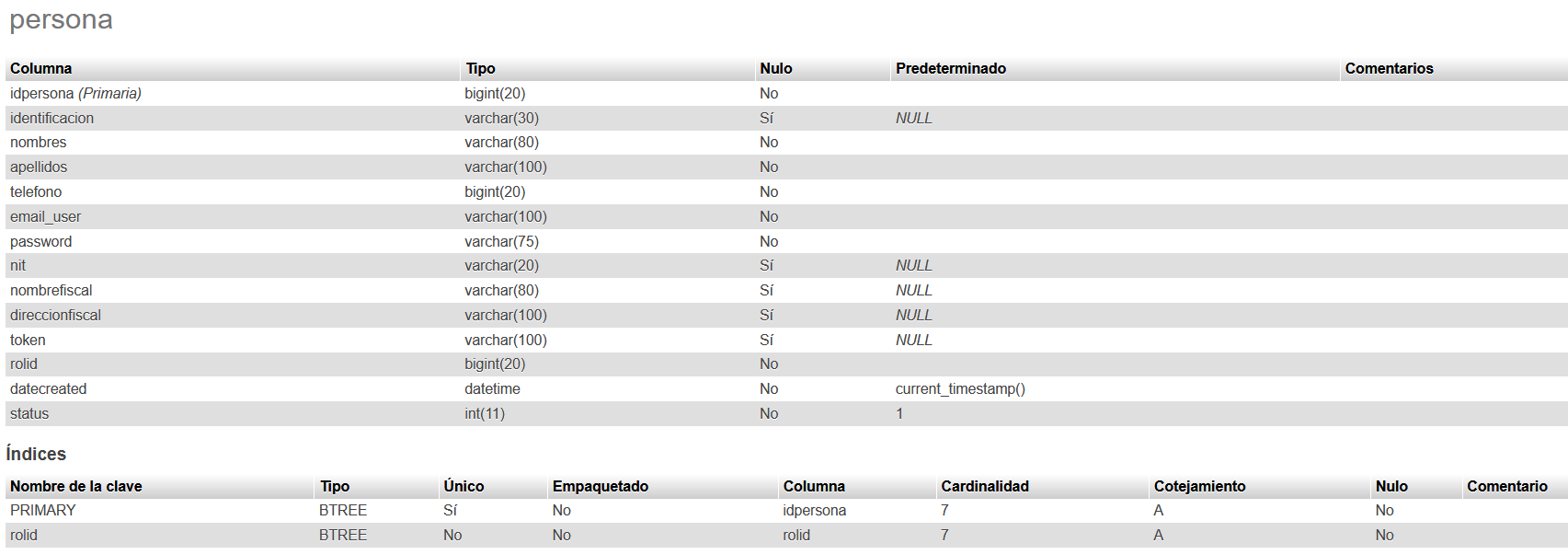








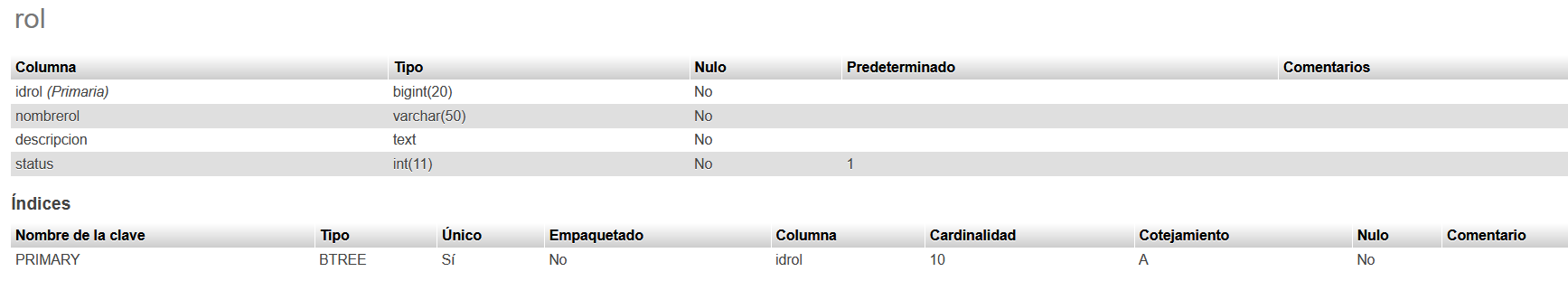




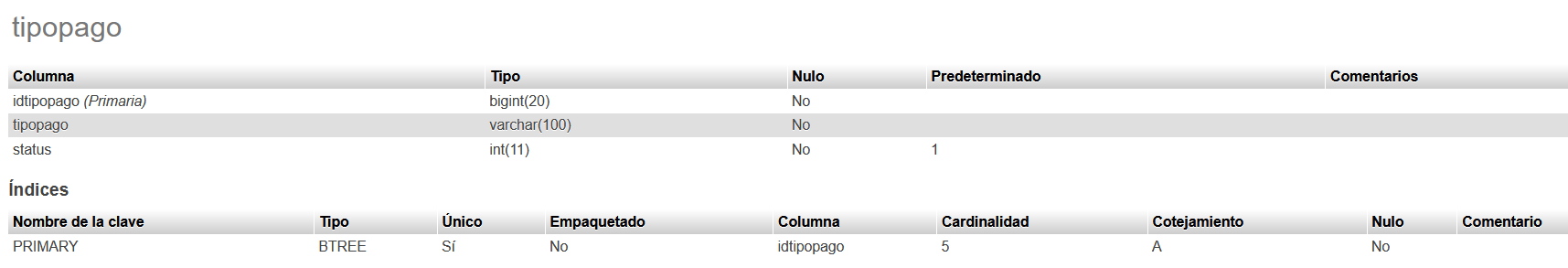












## 9.9 Políticas de seguridad de los datos

Política de Seguridad de Datos

1. Objetivo

Definir las medidas de seguridad necesarias para proteger la información y garantizar la confidencialidad, integridad y disponibilidad de los datos en el sistema desarrollado con PHP, MySQL y JavaScript.

2. Alcance

Esta política aplica a todos los desarrolladores, administradores y usuarios que interactúen con el sistema, así como a las aplicaciones y bases de datos relacionadas.

3. Políticas de Seguridad

A. PHP (Backend)

Validación de entradas:

Toda entrada proveniente del cliente debe ser validada antes de almacenarse en la base de datos.

Se recomienda el uso de librerías de validación para garantizar datos seguros.

Protección contra inyección SQL:

Todas las consultas deben realizarse mediante sentencias preparadas con PDO.

Gestión de sesiones:

Utilizar session\_start() con cookies seguras (HttpOnly, Secure).

Regenerar el ID de sesión después de un inicio de sesión exitoso para prevenir secuestro de sesión.

B. MySQL (Base de Datos)

Gestión de roles y permisos:

Se deben crear usuarios con privilegios específicos siguiendo el principio de mínimo privilegio.

Se establecen tres roles principales:

Administrador: acceso total a todas las operaciones.

Vendedor: acceso restringido a módulos específicos de ventas.

Usuario: acceso únicamente al catálogo.

Cifrado de contraseñas:

Las contraseñas deben almacenarse cifradas utilizando el algoritmo SHA-256.

Auditoría de datos:

El sistema debe registrar en tablas de auditoría los accesos, cambios y operaciones relevantes realizadas en la base de datos.

C. JavaScript (Frontend)

Comunicación con el servidor:

Todas las peticiones se realizarán mediante AJAX, y el servidor responderá en formato JSON.

Seguridad en el código:

Se deben emplear librerías seguras y actualizadas para evitar vulnerabilidades.

Transmisión segura:

Todo intercambio de información se realizará bajo el protocolo HTTPS.

Gestión de tokens:

El almacenamiento de tokens se realizará en cookies seguras, configuradas con banderas HttpOnly y Secure.

4. Cumplimiento Legal

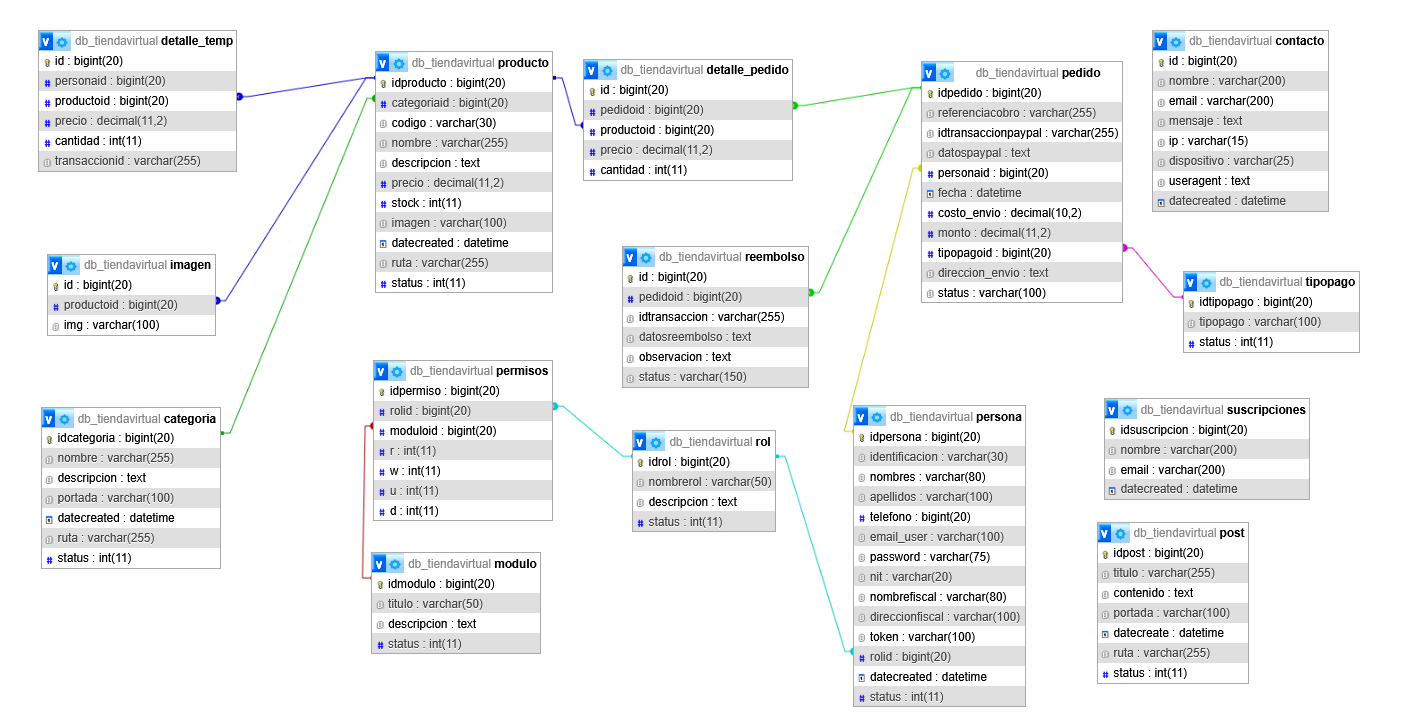
La aplicación deberá cumplir con la normativa vigente de protección de datos personales ley 1581 de 2012 en Colombia.

5. Revisión y Actualización

Esta política será revisada y actualizada al menos una vez al año o cuando se realicen cambios significativos en la arquitectura del sistema o la normativa aplicable.

# Construcción del software

## 10.1 Base de datos para el software a partir del modelo de datos



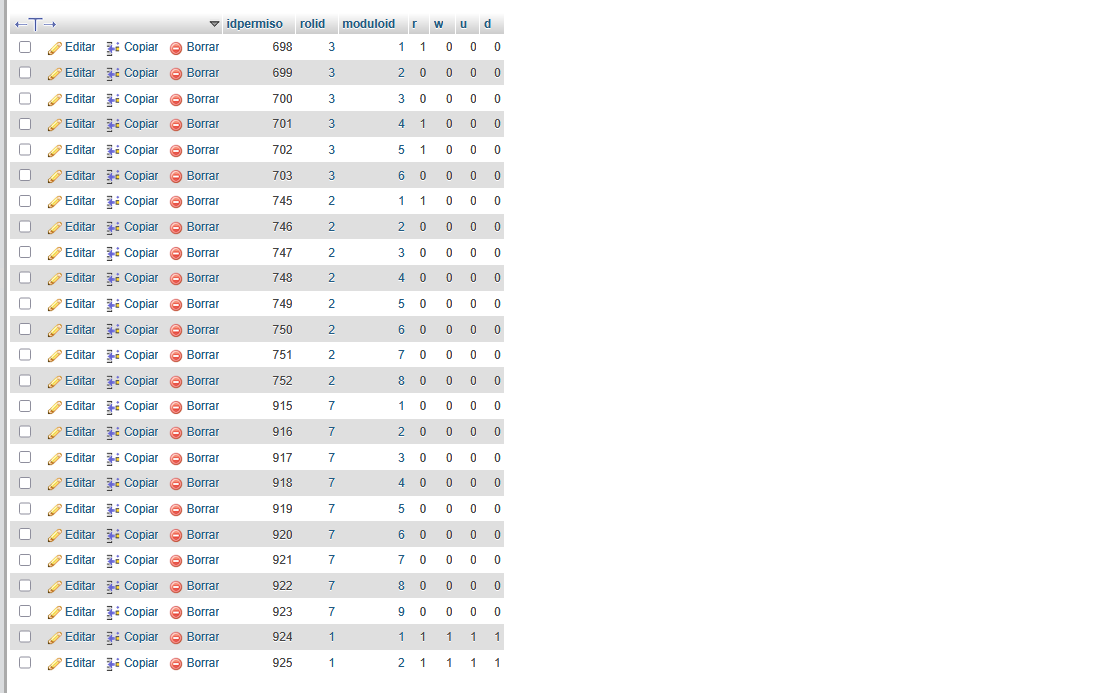


## 10.1.1 Objetos de la base de datos (procedimientos almacenados, vistas, disparadores)

No tenemos

## 10.1.2 Esquema de seguridad de los datos

Creamos esta tabla que va relacionada con la tabla roles para los permisos y protección de datos



La llave primaria es idpermiso que con un identificador único del 698 al 925

Como foránea tenemos rolid la cual es el identificador único del rol en este ejemplo creamos 7 roles distintos

Al igual como foránea tenemos moduloid el cual hace referencia a los módulos ejemplo: personas, productos, reportes etc…

Permisos de acción:

Estos son banderas (0 o 1) que indican si un rol tiene permiso o no:

r (Read / Leer):  
1 = puede ver el módulo, 0 = no puede verlo.

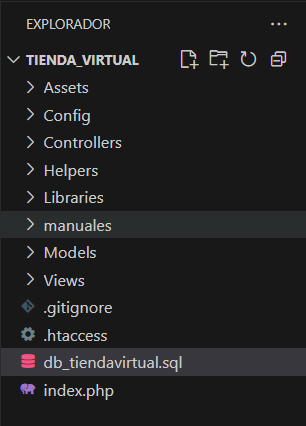
w (Write / Escribir o Crear):  
1 = puede crear o registrar datos, 0 = no puede.

u (Update / Actualizar):  
1 = puede modificar datos, 0 = no puede.

d (Delete / Eliminar):  
1 = puede eliminar registros, 0 = no puede.

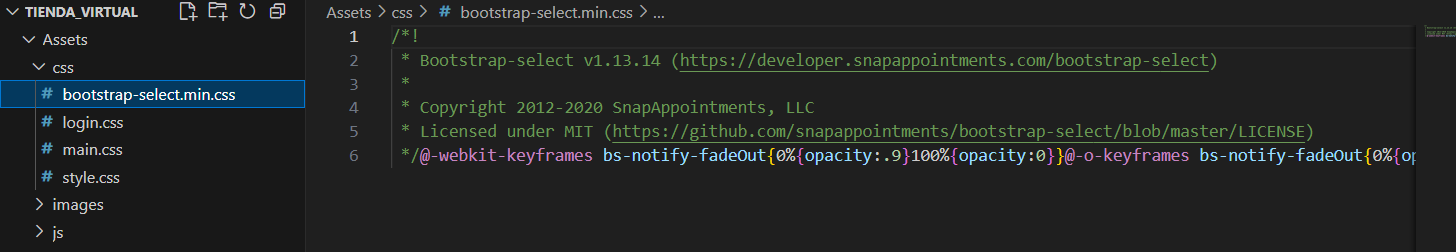
## 10.2 Codificación del software de acuerdo con el diseño establecido

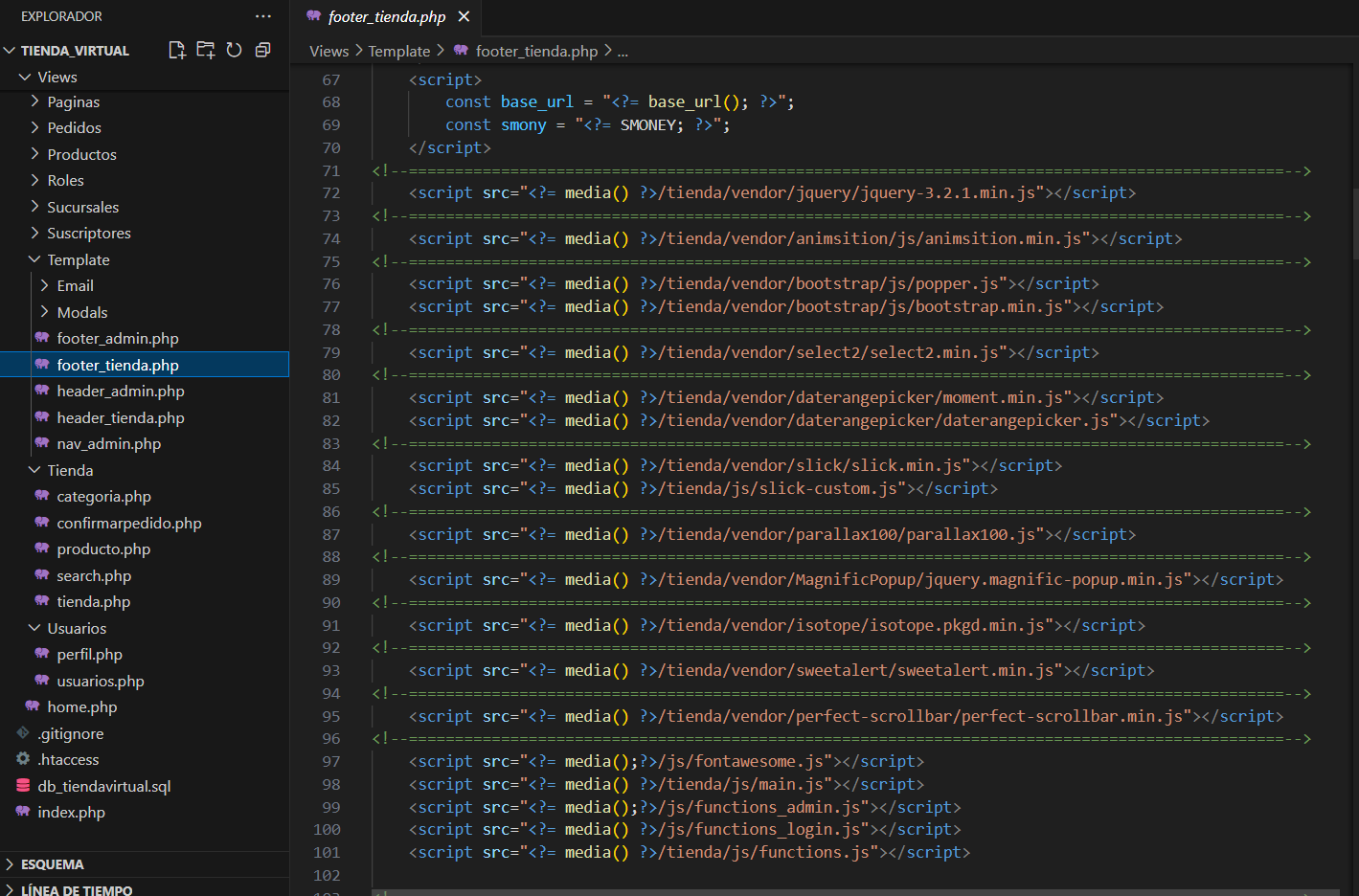
Ya teniendo la base de datos bien definida y la estructura de nuestro software empezamos con la codificación en visual code, con una estructura MVC que es la recomendada para ecommerse y utilizando PHP, MYSQL Y Java script.

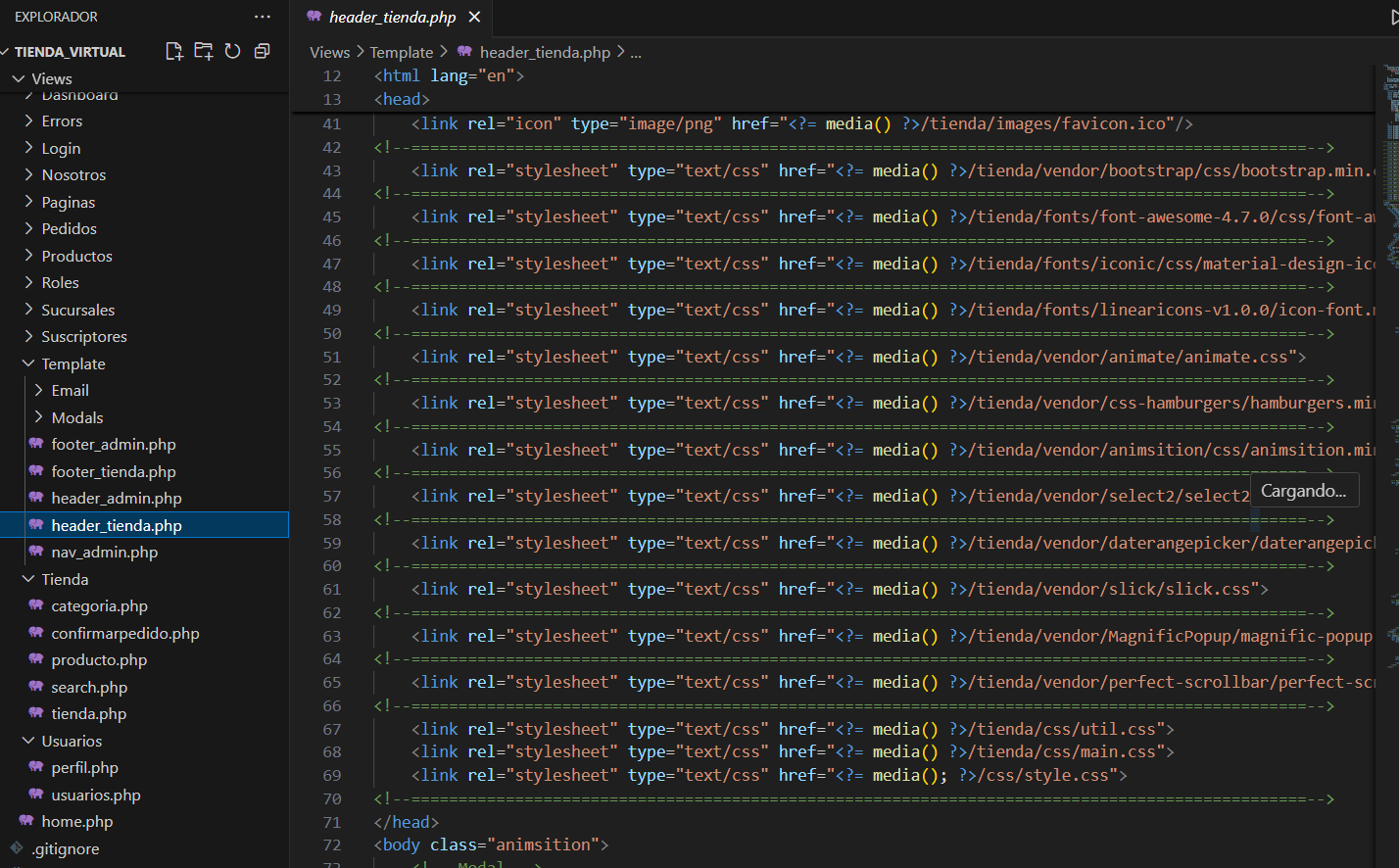


## 10.2.1 Front-end

Para el front-end, manejamos Bootstrap como plantilla de diseño la cual ya viene con bibliotecas incluidas

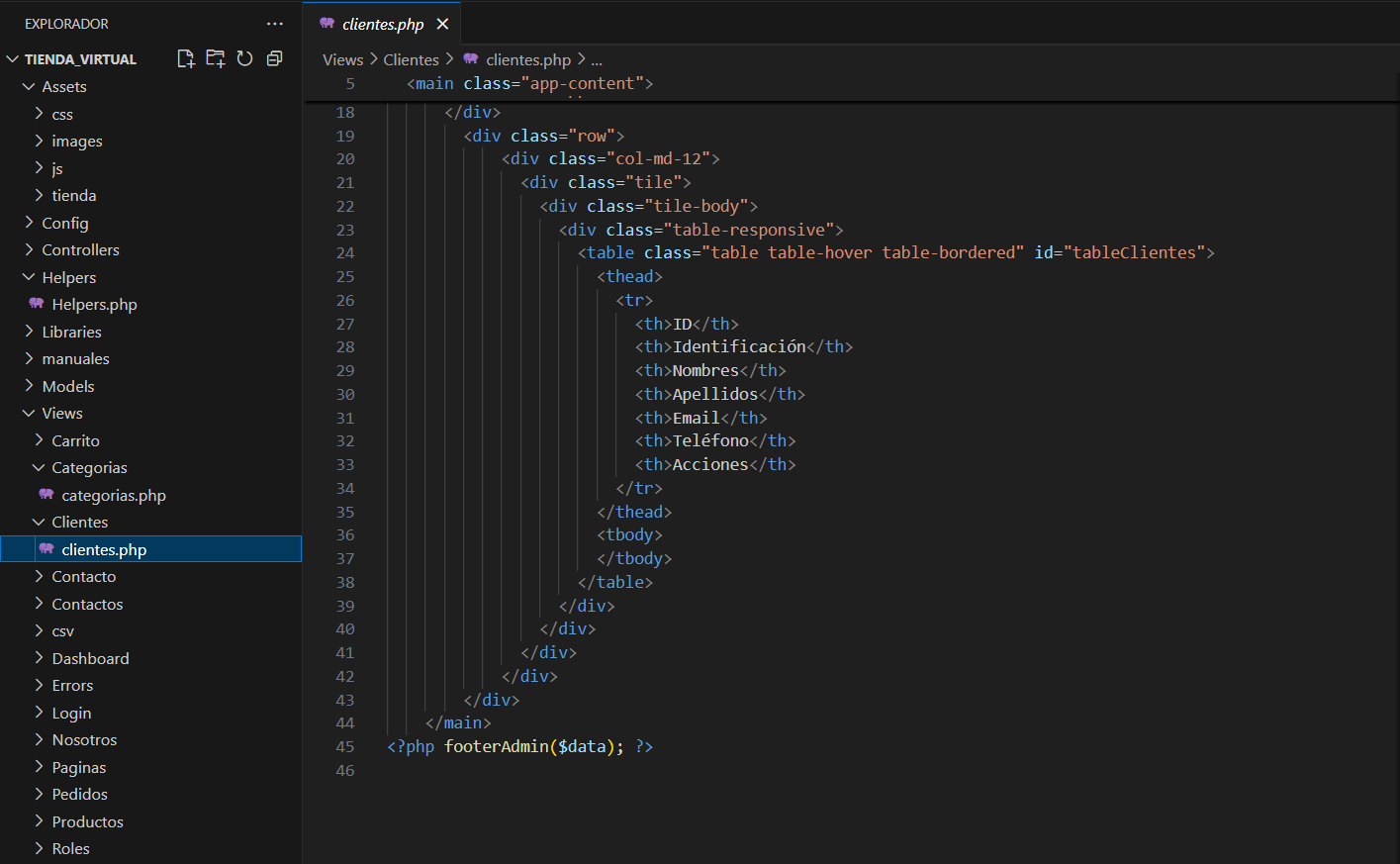




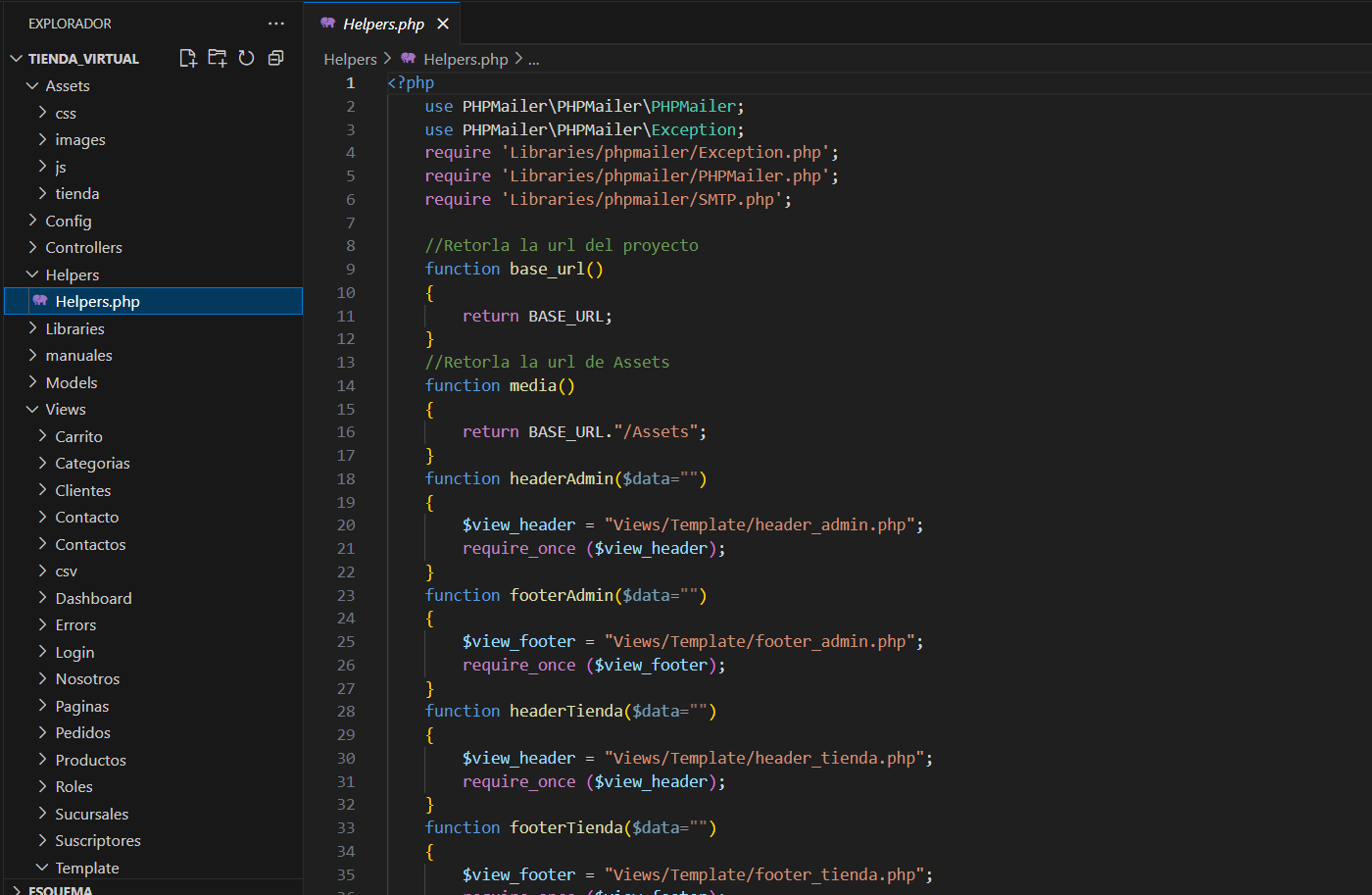


En general cada vista la manejamos por separado y cada vista la conectábamos con el footer y el header que lo realizamos en una carpeta aparte y en ella tenemos todos los scripts que vienen por defecto del Bootstrap

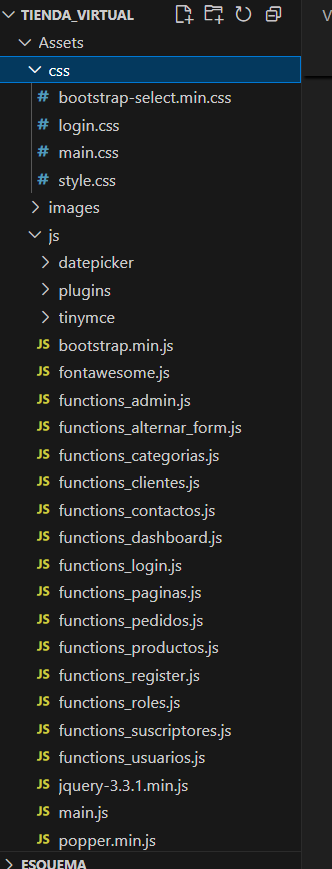
Ejemplo con cliente:



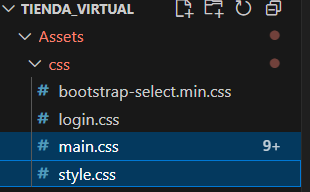
Manejamos unas carpetas de helpers las cuales nos ayudaron con las rutas por eso en el código la ruta de (src="<?= media(); ?>)

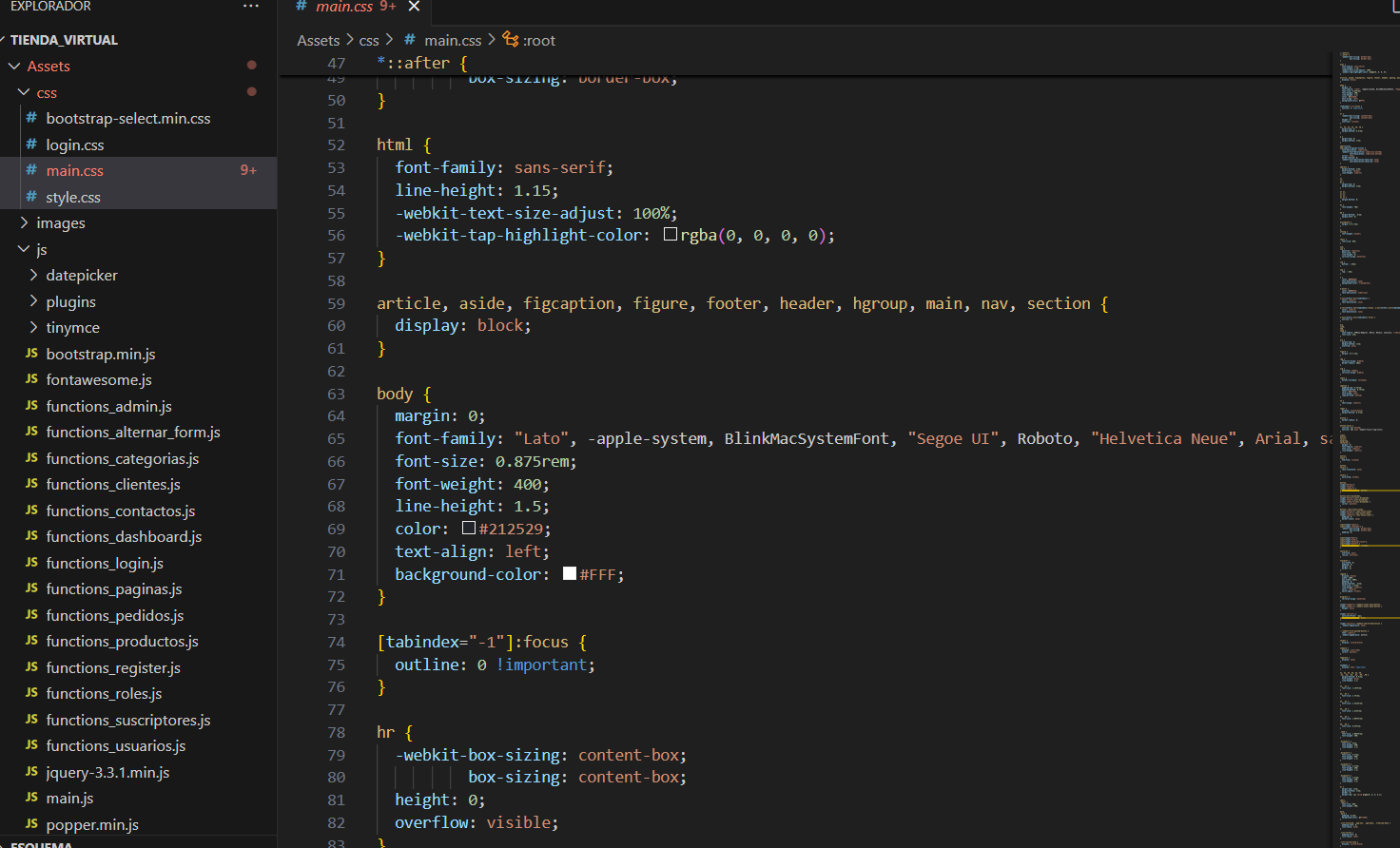


Para el css y el js lo pusimos en una carpeta de archivos estáticos llamada Assets



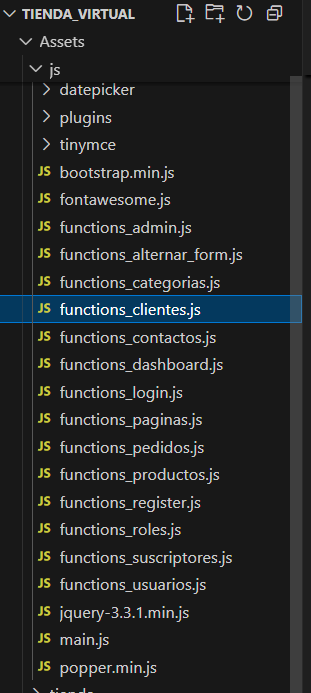
Main.css y style.css son las dos principales, el login lo hicimos por aparte sin tomar el del Bootstrap entonces por eso está por separado



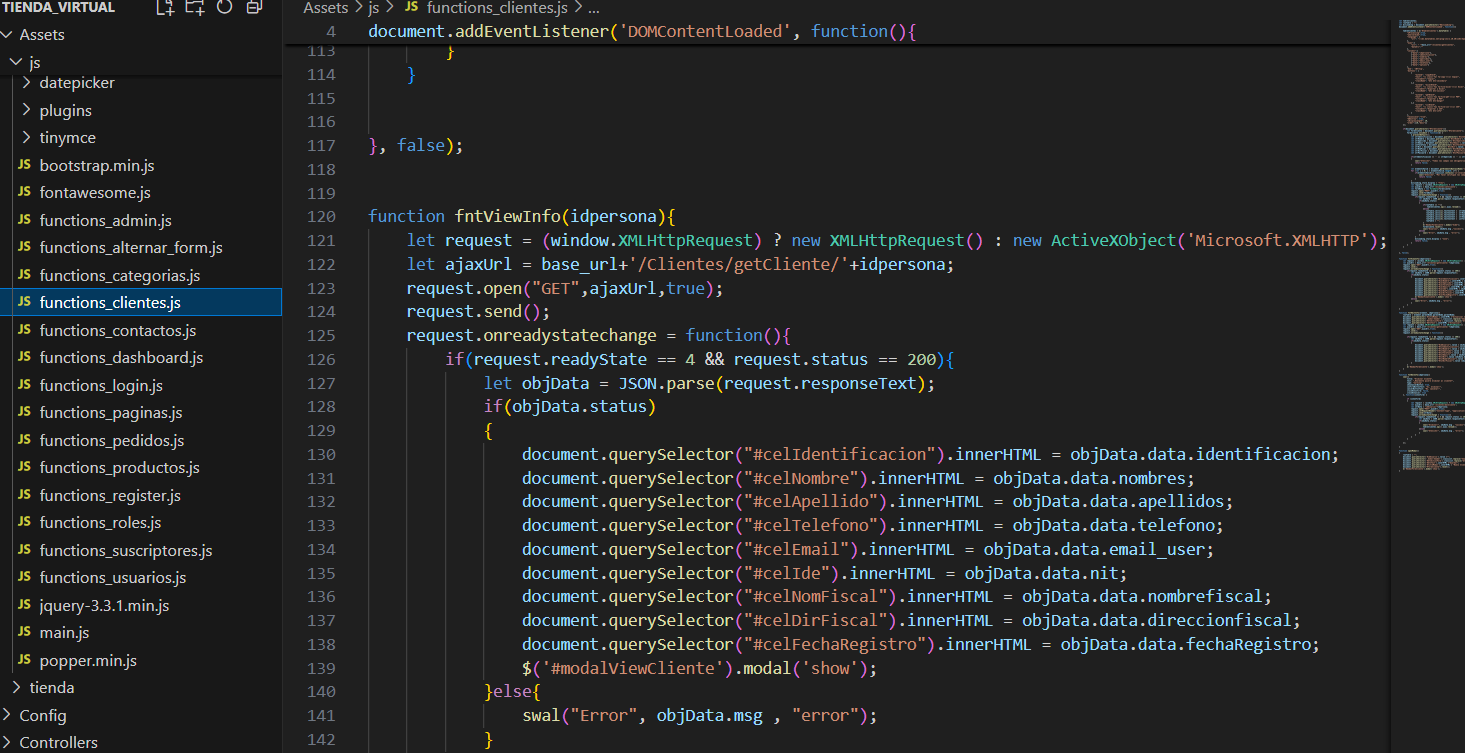
Ejemplo estructura:  


Se toman los nombres de las clases y se le aplica el tipo de letra, posición etc…

Para java script utilizamos funciones para cada módulo donde haya consultas a la base de datos con Ajax y json

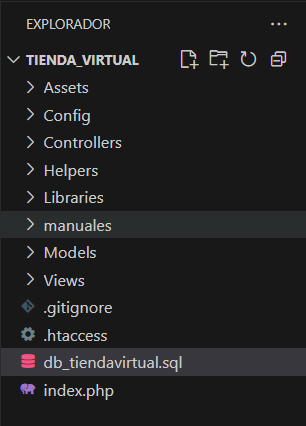


Ejemplo:

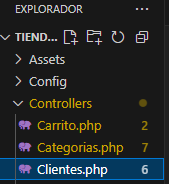


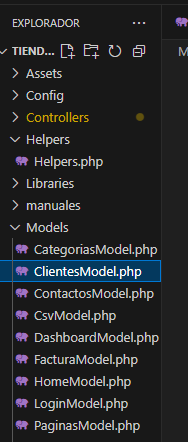
## 10.2.2 Estándar de codificación (Documentación del código, aplicabilidad del estándar de codificación)

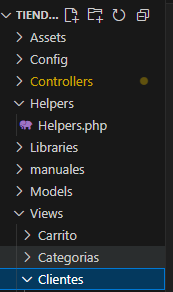
Ya para el back end utilizamos el lenguaje de programación PHP, el cual es el lenguaje mas común en el mundo para la creación de paginas web, en el utilizamos una estructura MVC(modelo, vista y controlador)



Y por separado manejamos cada modulo, ejemplo en el caso de contacto, tiene una vista, tiene su controlador y tiene su models





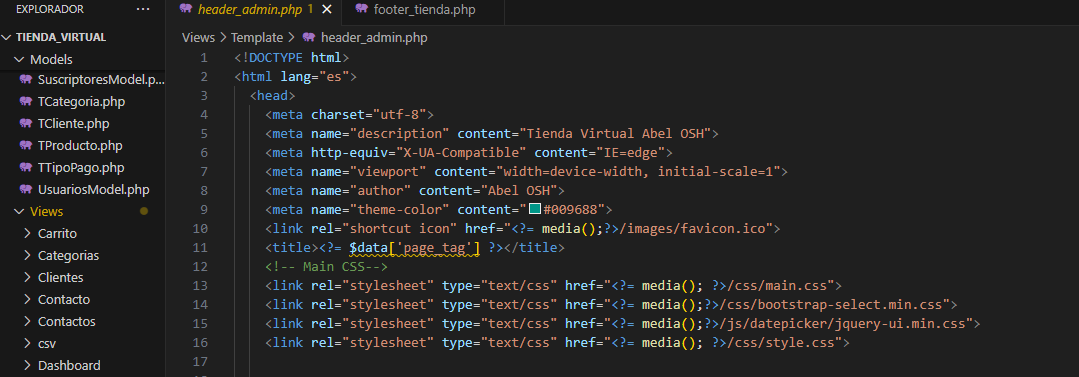


Y el código se encuentra bien documentado, dando una breve explicación de cada función

FALTA

## 10.2.3 Código fuente de los módulos del software web y móvil

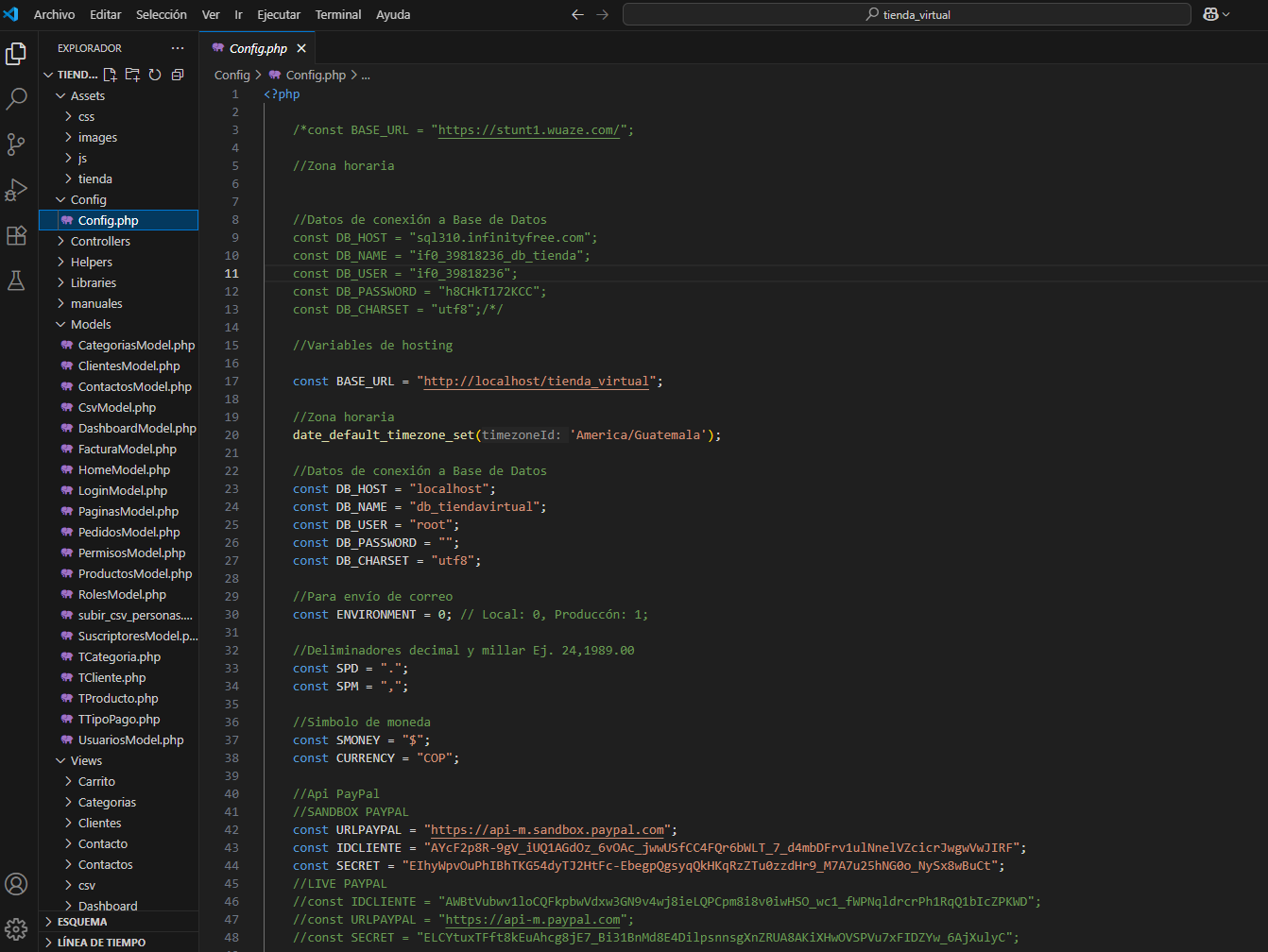
En mi código en la parte de las vistas en la carpeta de témplate tengo el footer y el header, en el cual en tengo la línea de código (<meta name="viewport" content="width=device-width, initial-scale=1">) la cual indica que el ancho de la página debe ajustarse al ancho del dispositivo (sea un monitor, una tablet o un celular) y la parte del código que dice (initial-scale=1) , define el nivel de zoom inicial en 100%.



Código fuente

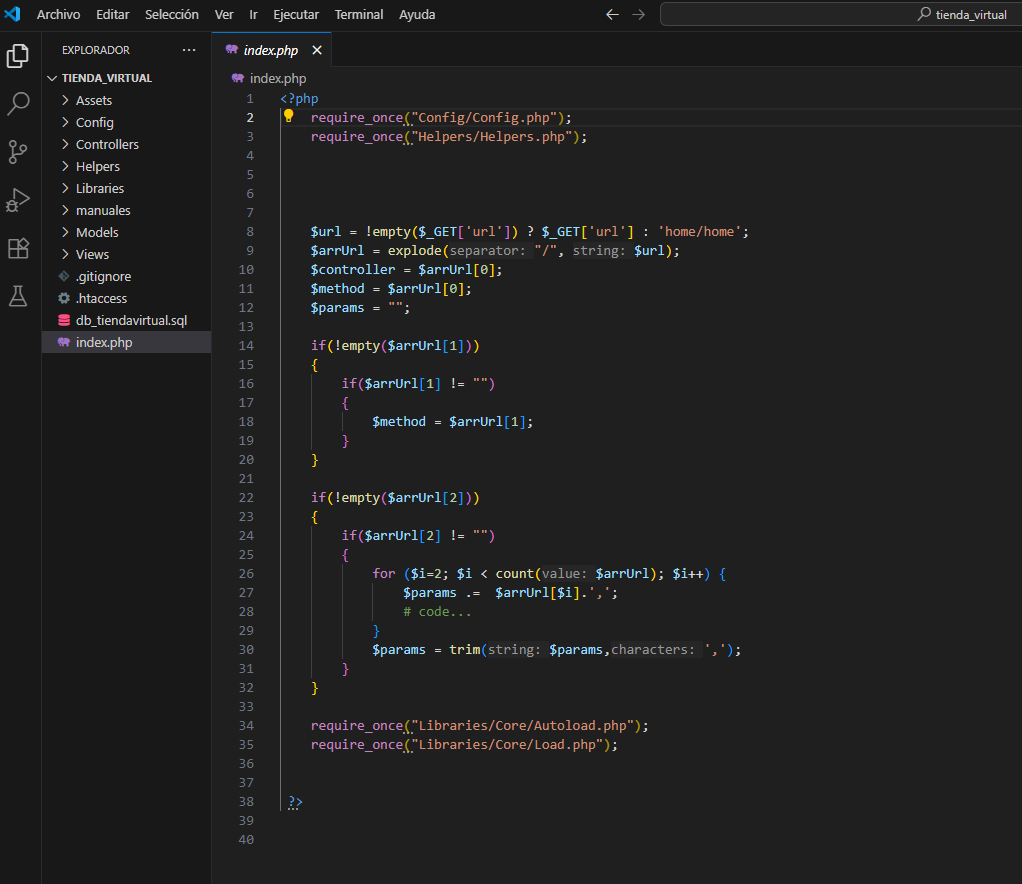
Archivo config:

Archivo config es el encargado de la conexión de la base de datos con el código fuente

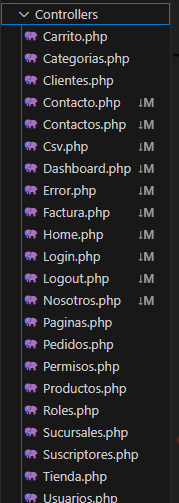
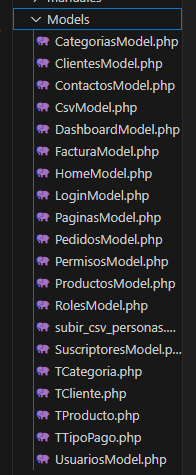
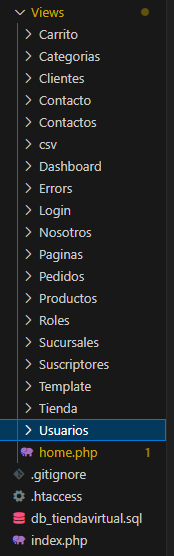


Archivo index:

index es uno de los mas importante ya que es el primer archivo en cargarse al iniciar un programa por eso es el que tiene un require\_once(“Config/Config.php”)



Cada modulo esta por separado y tiene su propio archivo controller, models y vista



Archivo .htaccess:

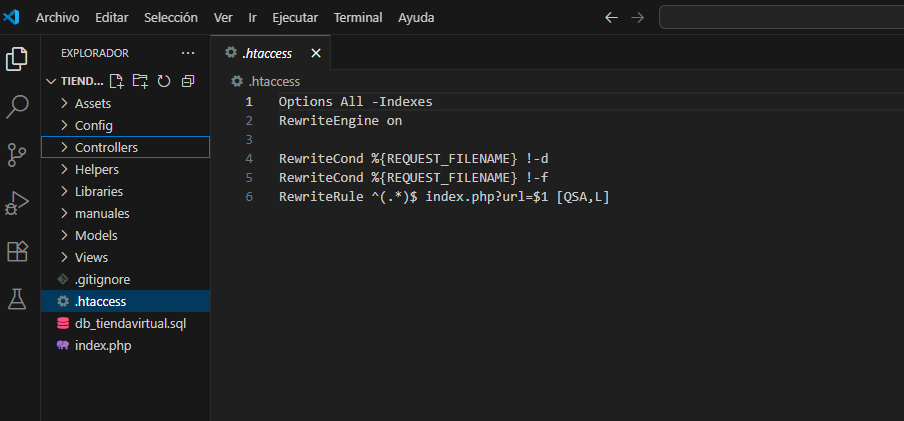
El archivo .htaccess es como un portero de tu aplicación web.

Hace 2 cosas principales:

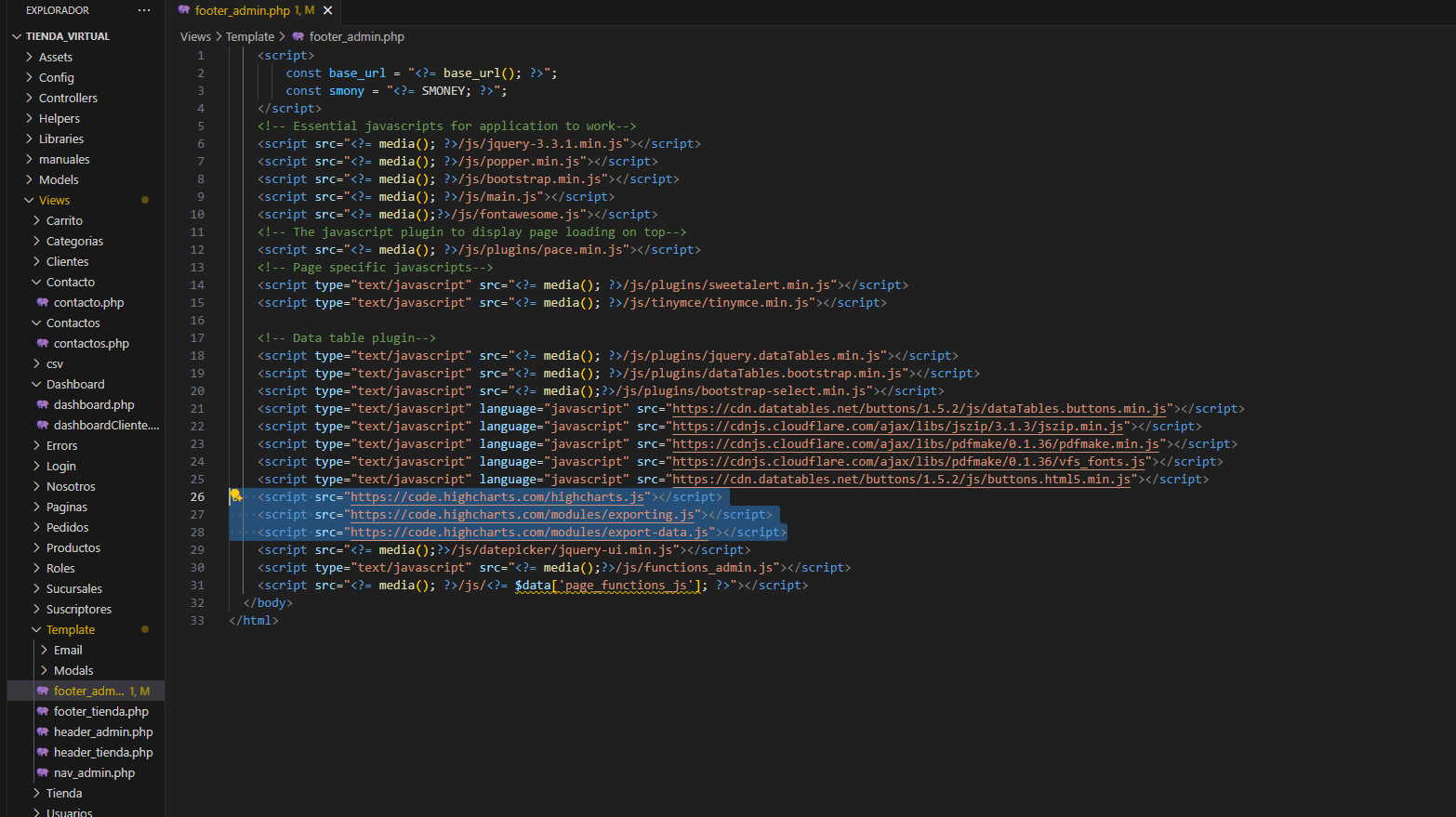
No deja que la gente vea carpetas y archivos directamente.

Si alguien intenta entrar a una carpeta de tu proyecto, no podrá ver lo que hay adentro.

Convierte las direcciones largas y feas en direcciones cortas y bonitas.



## 10.2.4 Servicios web



Mi sistema consume servicios web de terceros para añadir funcionalidades:

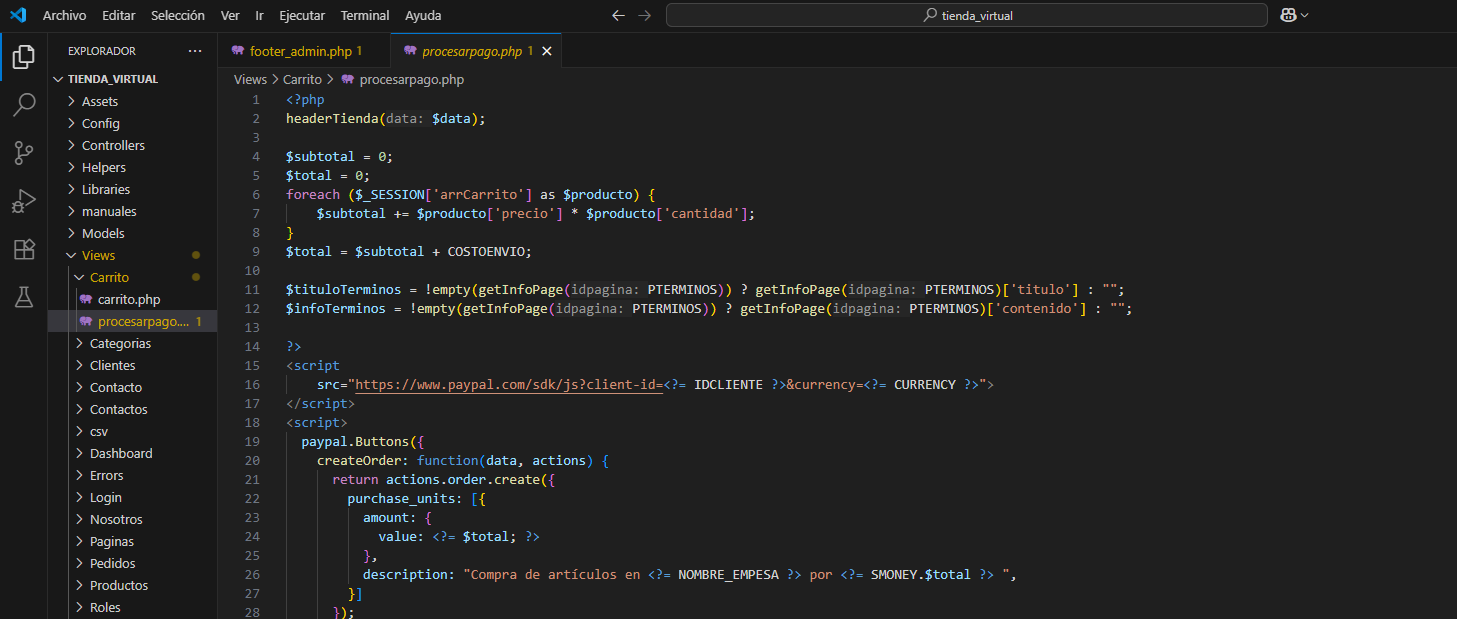
Highcharts para estadísticas en gráficos.

DataTables con CDN para tablas interactivas y exportación de datos.

Esto permite que no tengas que programar todo desde cero, sino aprovechar servicios disponibles en la web.



También tenemos el chat bot en el footer de la tienda, con un servicio web de voiceflow.

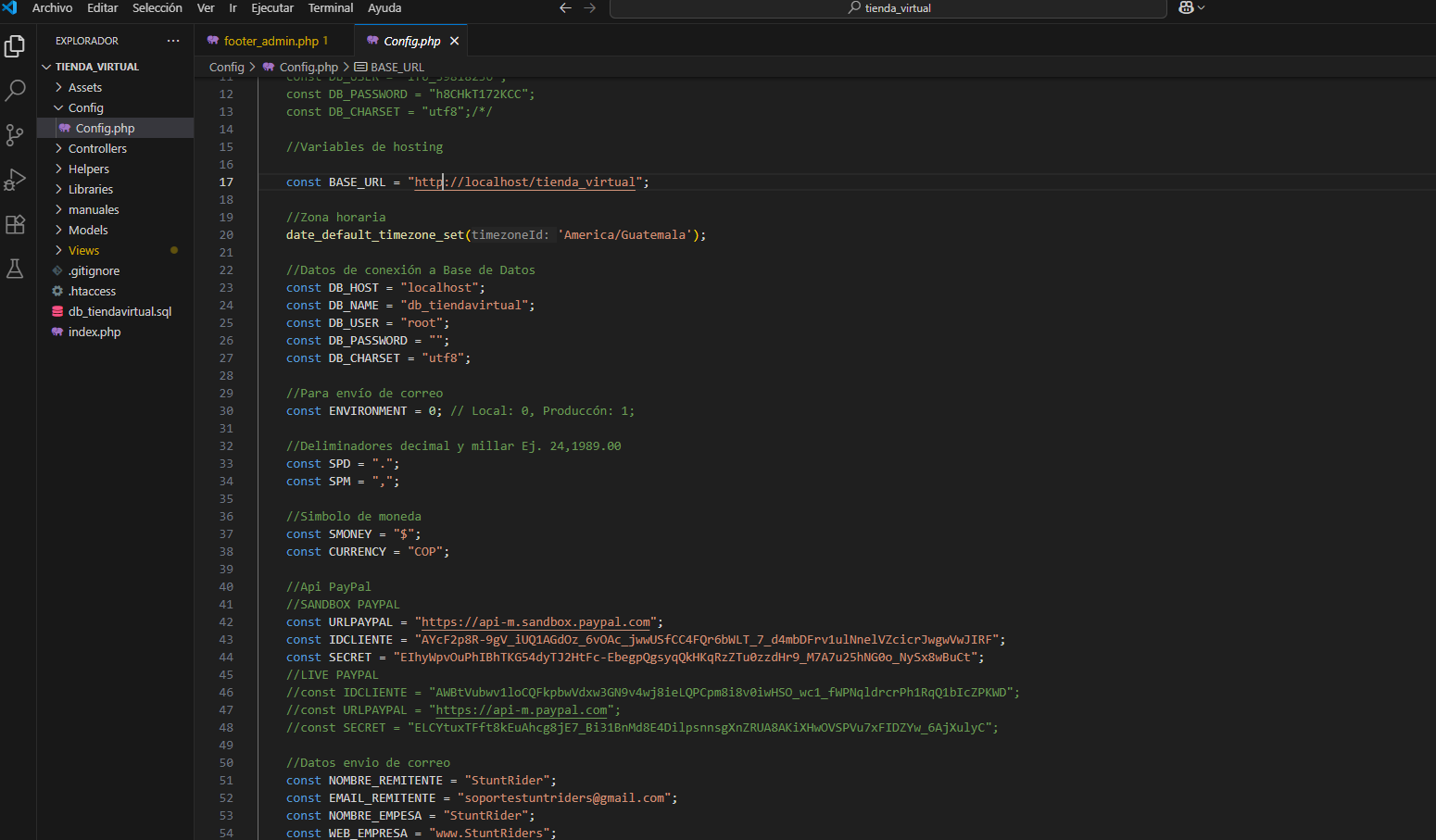


También utilizamos un servicio web como el api de paypal.

PayPal ofrece una API.

Mi sistema envía datos como el monto total y la descripción de la compra.

PayPal responde autorizando o rechazando el pago.



Aquí sí estás definiendo las credenciales y endpoints para consumir el servicio web de PayPal.

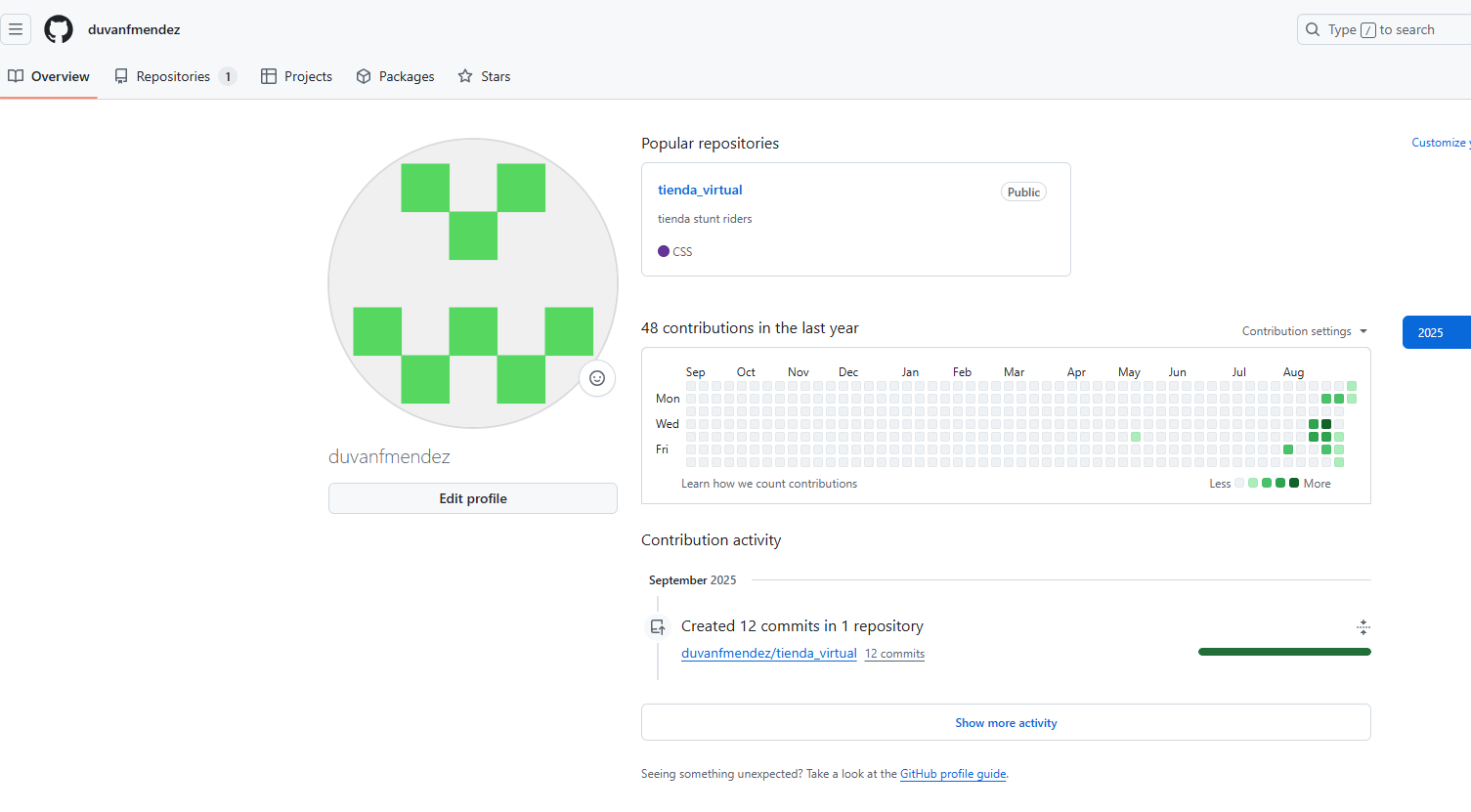
sandbox entorno de pruebas.

live entorno real de producción.

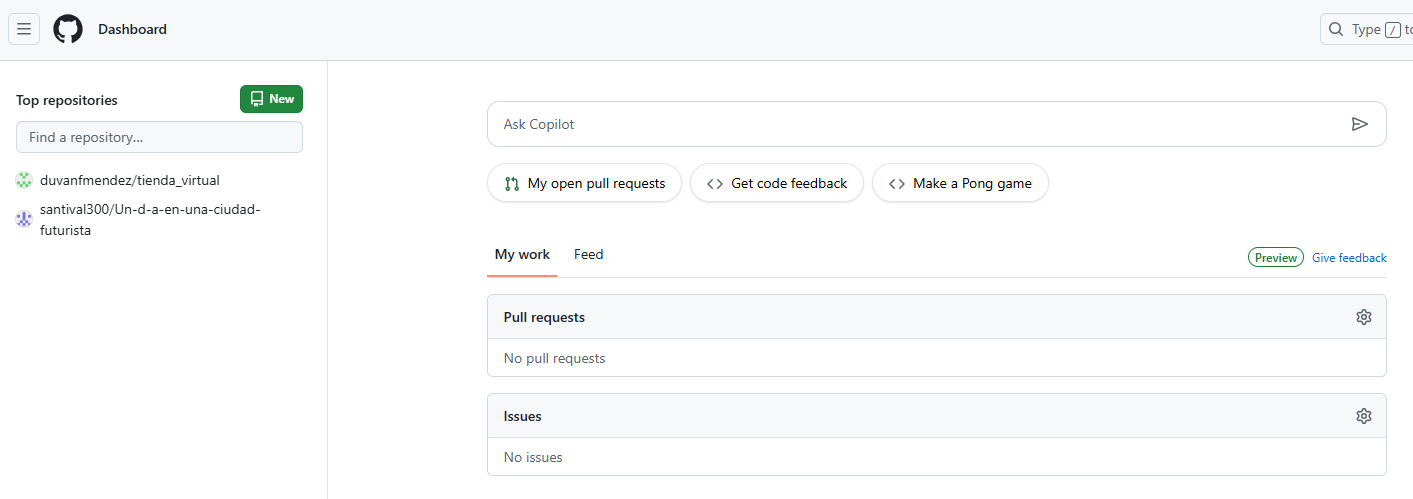
## 10.2.5 Control de versiones

Para el control de versiones utilizamos git hub complementado con git desktop

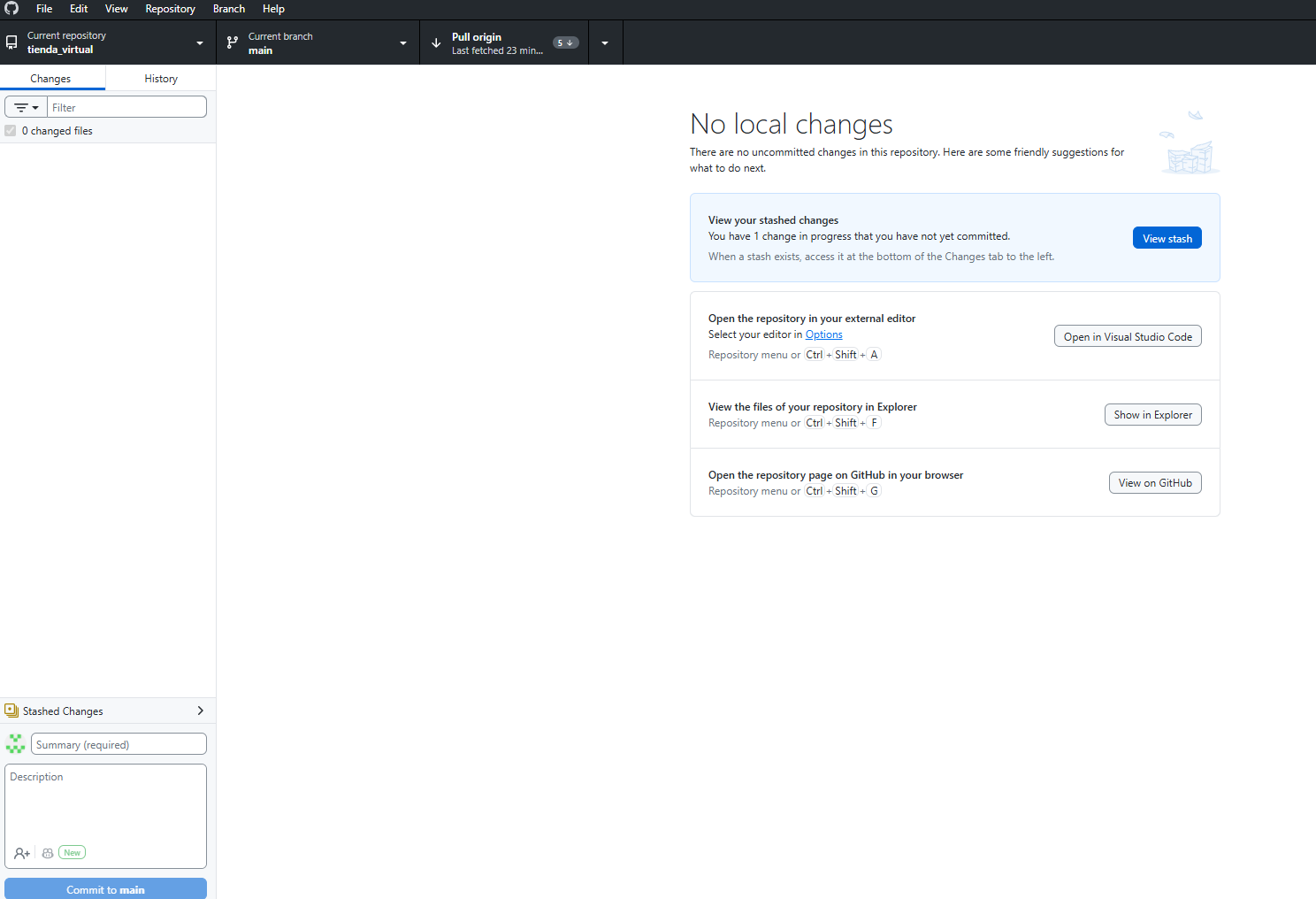
Paso inicial crear perfil de git hub



Creamos el repositorio a trabajar



Después descargamos git desktop el cual nos pide conectar con un git y hacemos la conexión con el git hub y desde ahí cargamos los archivos



Después revisamos en el GitHub que se haya cargado con éxito todos los archivos

