

## MANUAL TECNICO MÓVIL

### 1. ALCANCE:

El sistema tiene como objetivo permitir a la ferretería Ferremateriales El Maestro (ubicada en Campeche) ofrecer una experiencia digitalizada y simplificada para sus clientes. A través de la app móvil, los usuarios podrán interactuar directamente con la ferretería para consultar productos, realizar cotizaciones y gestionar pedidos.

La app móvil servirá como un canal para que los clientes naveguen por el inventario de productos de la ferretería, reciban presupuestos de cotización personalizados y gestionen sus compras, con la posibilidad de recibir soporte directo del administrador de la ferretería.

### Módulos funcionales:

- Visualización de Productos.
- Promociones.
- Soporte o ayuda con la ferretería.
- Carrito de compra.
- Métodos de pago o compras en línea.
- Vista de favoritos.

### 2. Descripción de procesos:

- Visualización de Productos.
  - Filtros de búsqueda.
  - Vista detallada de cada producto, con especificaciones adicionales (por ejemplo, dimensiones, marca, tipo de material, etc.).
  - Los usuarios podrán ver los productos disponibles en la ferretería, con información detallada como nombre, descripción, precio, fotos y disponibilidad.
  - Opción para agregar a favoritos.
  - Opción para agregar al carrito y hacer compras.
- Autenticación y Gestión de Usuario:
  - Registrarse y autenticar su cuenta para poder guardar sus cotizaciones, datos de contacto.
  - Funcionalidad de recuperación de contraseña si es necesario.
  - Capacidad para editar perfil y agregar o actualizar dirección de envío y métodos de pago.

- Carrito de compras:
  - Agregar productos al carrito de compras.
  - Editar el carrito: modificar cantidades, eliminar productos o agregar nuevos artículos.
  - Cotizar productos.
  - Enviar una cotización por correo electrónico o mediante otros canales de contacto para recibir un presupuesto detallado de la ferretería.
- Información de Contacto:
  - Funcionalidad para contactar a la ferretería por diferentes canales: correo electrónico, teléfono y WhatsApp.
  - Mapa interactivo con la dirección de la ferretería y opciones de navegación utilizando Google Maps (mediante un botón).
  - Formulario de contacto directo para realizar consultas o pedir soporte.
- Multicanalidad y Opciones de Pago:
  - Opciones de pago integradas, como transferencias bancarias, tarjetas de crédito/débito.
  - Integración con sistemas de pago locales si es necesario.

### **Requerimientos funcionales:**

- Los usuarios podrán ver los productos disponibles en la ferretería sin necesidad de registrarse o iniciar sesión. Cada producto tendrá su nombre, descripción, precio, imágenes y cualquier otra especificación relevante.
- Los usuarios podrán agregar productos a su carrito(sin necesidad de realizar la compra solo para cotizar). Los productos agregados se podrán editar.
- Los usuarios podrán agregar productos a favoritos (sin necesidad de iniciar sesión). Los productos agregados se podrán editar.
- Los usuarios podrán enviar mensajes mediante un formulario para consultar más detalles sobre productos o servicios. El formulario incluirá campos como nombre, email, consulta.

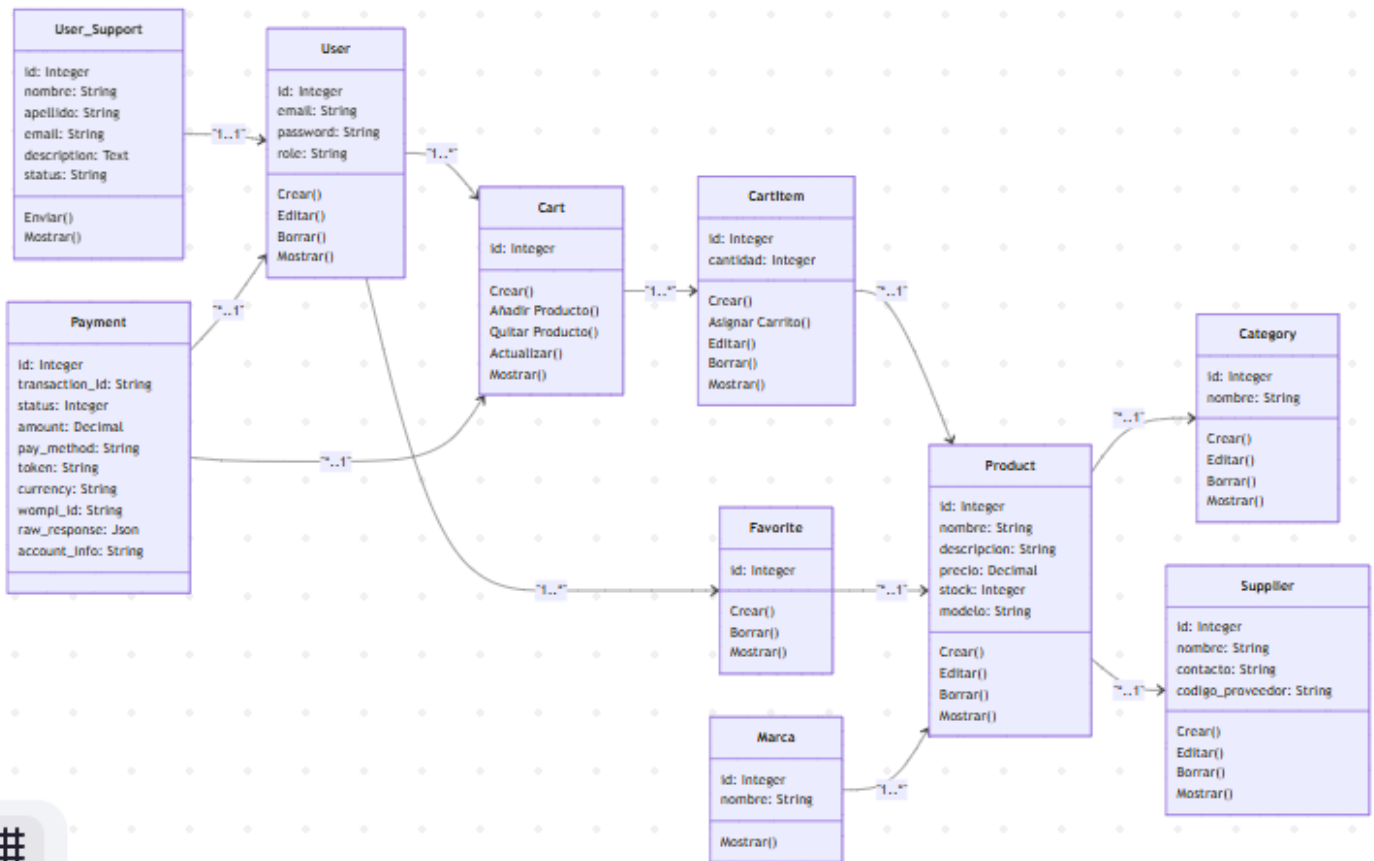
- Los usuarios podrán registrarse, iniciar sesión y gestionar su perfil. El sistema almacenará información básica de los usuarios como nombre, dirección de envío, datos de contacto.

### **Requerimientos No Funcionales**

- El acceso al sistema de la app móvil no requerirá autenticación para visualizar productos o realizar cotizaciones. Si el usuario decide registrarse o iniciar sesión, la autenticación será segura mediante tokens JWT o un sistema de autenticación similar.
- Aunque la app no gestionará contraseñas de la parte administrativa, la información del usuario (como datos personales y métodos de pago) se almacenará de forma segura mediante cifrado, en cumplimiento con las normas de protección de datos.
- La app debe ofrecer una interfaz sencilla y clara, fácil de navegar para usuarios sin experiencia técnica. Se empleará un diseño limpio con un menú de navegación sencillo y pantallas bien estructuradas. La interacción del usuario se simplificará para facilitar el uso.
- La app mostrará mensajes de error claros y comprensibles, por ejemplo, si el usuario no puede agregar productos al carrito o hay un error en la cotización. Los errores no serán técnicos, sino que estarán orientados a los usuarios (ej. “No se pudo procesar el pago, por favor intente nuevamente”).
- La app deberá responder rápidamente a las consultas de productos. El tiempo de respuesta para cargar productos y mostrar los resultados de la búsqueda será mínimo.
- La app debe ser completamente adaptable a diferentes dispositivos móviles (smartphones y tablets) con un diseño responsive. Las pantallas de la app se ajustarán adecuadamente a diferentes tamaños de pantalla, garantizando una experiencia fluida en dispositivos iOS y Android.
- Escalabilidad: El sistema debe permitir la integración de nuevas funcionalidades en el futuro, como la implementación de nuevos métodos de pago o características adicionales en la app.

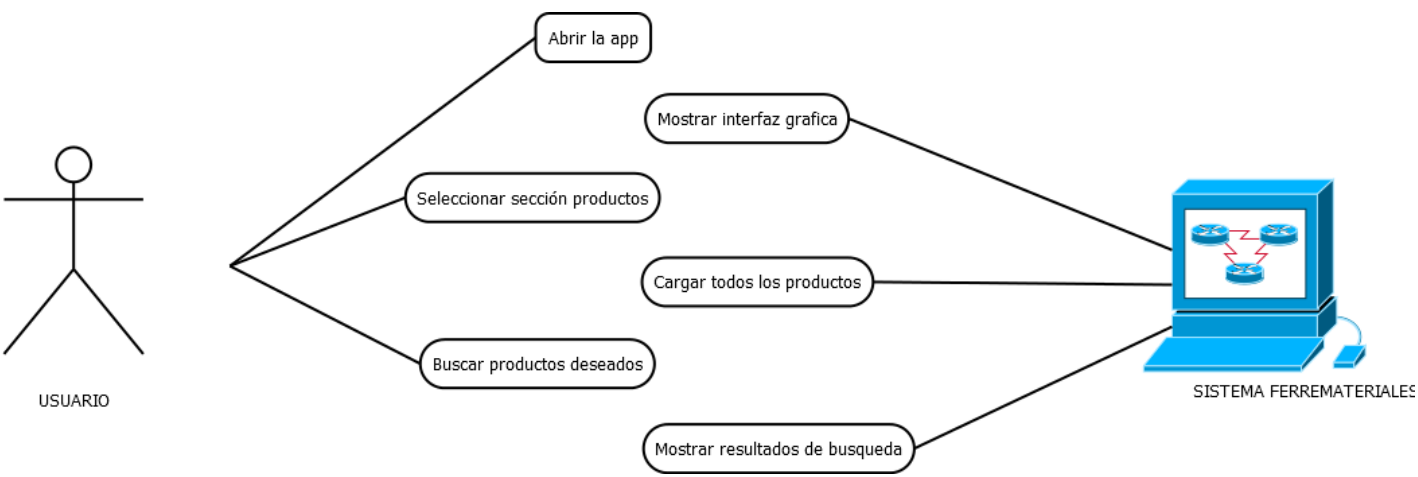
### 3. Diagramas UML (clase y caso de usos y ficha)

#### DIAGRAMA DE CLASE



R01

NOMBRE: VER PRODUCTOS SIN REGISTRARSE



|  |       |  |
|--|-------|--|
| FECHA : 27/09/2025   |       |  |
| CASO DE USO 01   |       |  |
| NOMBRE DEL CASO DE USOS: VER PRODUCTOS SIN REGISTRARSE   |       |  |
| AUTOR: ELBERTO PERTUZ  |       |  |
| ACTORES: USUARIO Y SISTEMA   |       |  |
| PRECONDICIÓN: EL USUARIO DEBE TENER CONEXIÓN A INTERNET, EL USUARIO DEBE TENER LA APP DESCARGADA YA SEA EN SU TELEFONO MOVIL O TABLETA.                                    |       |  |
| FLUJO NORMAL   | PASOS | DESCRIPCIONES  |
|  | 1     | EL USUARIO ENTRA A LA APLICACIÓN S.I.E   |
|  | 2     | EL SISTEMA MUESTRA PÁGINA DE INICIO  |
|  | 3     | EL USUARIO SELECCIONA LA OPCIÓN DE VER PRODUCTOS   |
|  | 4     | EL SISTEMA LES MUESTRA TODOS LOS PRODUCTOS EN STOCK DE LA FERRETERIA CON SU NOMBRE, DESCRIPCIÓN, PRECIO, IMAGEN. |
|  | 5     | EL USUARIO BUSCA EL PRODUCTO DESEADO YA SEA POR NOMBRE, CATEGORIA O PRECIO.                                      |
|  | 6     | EL SISTEMA ARROJA RESULTADOS DE LA BUSQUEDA.   |
| FLUJO ALTERNAT   | PASOS | DESCRIPCIONES  |
|  | 2.1   | EL SISTEMA PRESENTA FALLA Y NO MUESTRA LA PÁGINA DE INICIO.  |
|  | 4.1   | EL SISTEMA NO CARGA PRODUCTOS  |
|  | 6.1   | EL SISTEMA NO ARROJA RESULTADOS DE BUSQUEDA.   |
| POSTCONDICIÓN: EL USUARIO PUEDE OBSERVAR LOS PRODUCTOS FILTRADOS O NO PARA ENCONTRAR EL QUE SE ADAPTE A SUS NECESIDADES Y GUSTOS SIN NECESIDAD DE CREAR E INICIAR SECCIÓN. |       |  |

R02

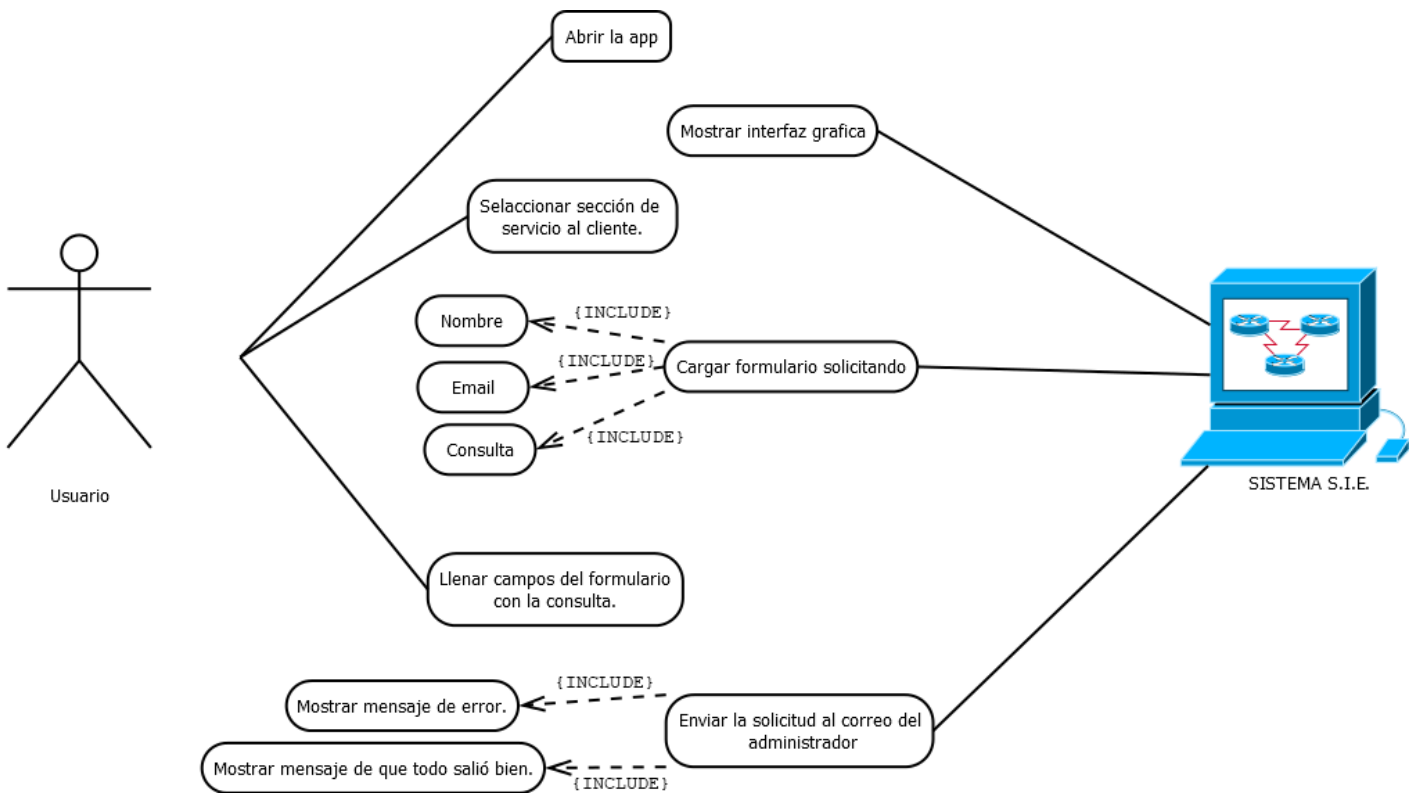
NOMBRE: AGREGAR PRODUCTOS A CARRITO DE COMPRA



|  |       |  |
|--|-------|--|
| FECHA : 27/09/2025   |       |  |
| NOMBRE DEL CASO DE USOS: AGREGAR PRODUCTOS A CARRITO DE COMPRA   |       |  |
| CASO DE USO 02   |       |  |
| AUTOR: ELBERTO PERTUZ  |       |  |
| ACTOR: USUARIO Y SISTEMA   |       |  |
| PRECONDICIÓN: EL USUARIO DEBE TENER CONEXIÓN A INTERNET, EL USUARIO DEBE TENER LA APP DESCARGADA YA SEA EN SU TELEFONO MOVIL O TABLETA.                            |       |  |
| FLUJO NORMAL   | PASOS | DESCRIPCIONES  |
|  | 1     | EL USUARIO ENTRA A LA APLICACIÓN S.I.E   |
|  | 2     | EL SISTEMA MUESTRA PÁGINA DE INICIO  |
|  | 3     | EL USUARIO SELECCIONA LA OPCIÓN DE VER PRODUCTOS   |
|  | 4     | EL SISTEMA LES MUESTRA TODOS LOS PRODUCTOS EN STOCK DE LA FERRETERIA CON SU NOMBRE, DESCRIPCIÓN, PRECIO, IMAGEN. |
|  | 5     | SELECCIONA PRODUCTOS DESEADOS Y LOS AGREGA AL CARRITO DE COMPRAS   |
|  | 6     | EL SISTEMA GUARDA TODOS Y CADA UNO DE LOS PRODUCTOS SELECCIONADOS AL CARRITO RELACIONADO CON ESE USUARIO         |
|  | 7     | EL USUARIO TOCA EL BOTON DEL CARRITO   |
|  | 8     | EL SISTEMA MUESTRA ESE CARRITO CON LOS PRODUCTOS RECIENTE AGREGADOS Y EL TOTAL                                   |
|  | 9     | EL USUARIO O LOS COMPRO O DEJA LA COTIZACIÓN AHÍ   |
| FLUJO ALTERNATIVO  | PASOS |  |
|  | 2.1   | EL SISTEMA PRESENTA FALLA Y NO MUESTRA LA PÁGINA DE INICIO.  |
|  | 4.1   | EL SISTEMA NO CARGA PRODUCTOS  |
|  | 6.1   | EL SISTEMA NO GUARDA TODOS LOS PRODUCTOS SELECCIONADOS   |
|  | 8.1   | NO CARGA EL CARRITO DE COMPRA O NO GENERA TOTAL  |
|  | 8.2   | EL SISTEMA NO GENERA TOTAL   |
| POSTCONDICIÓN: El usuario puede ver la cotización final de sus productos para hacerse un presupuesto y puede decidir comprar dichos productos o solo dejarlos ahí. |       |  |

R03

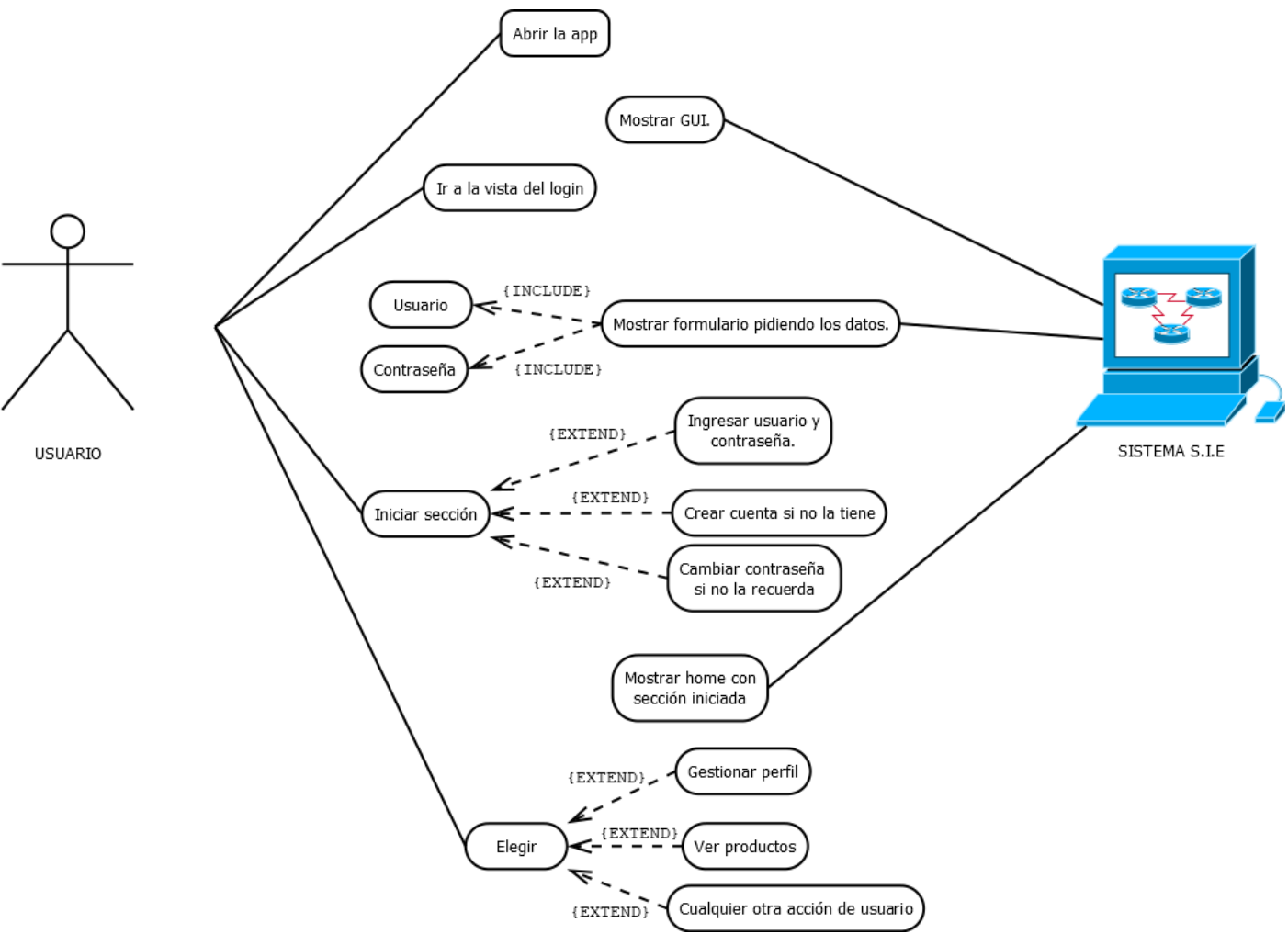
NOMBRE: FORMULARIO PARA ENIVAR CONSULTAS



|  |       |  |
|--|-------|--|
| FECHA : 27/09/2025   |       |  |
| NOMBRE DEL CASO DE USOS: FORMULARIO PARA ENIVAR CONSULTAS  |       |  |
| CASO DE USO 03   |       |  |
| AUTOR: ELBERTO PERTUZ  |       |  |
| ACTOR: USUARIO Y SISTEMA   |       |  |
| PRECONDICIÓN: EL USUARIO DEBE TENER CONEXIÓN A INTERNET, EL USUARIO DEBE TENER LA APP DESCARGADA YA SEA EN SU TELEFONO MOVIL O TABLETA.  |       |  |
| FLUJO NORMAL   | PASOS | DESCRIPCIONES  |
|  | 1     | EL USUARIO ENTRA A LA APLICACIÓN S.I.E   |
|  | 2     | EL SISTEMA MUESTRA PÁGINA DE INICIO  |
|  | 3     | EL USUARIO SELECCIONA LA OPCIÓN DE SERVICIO AL CLIENTE   |
|  | 4     | EL SISTEMA CARGA EL FORMULARIO CON LOS CAMPOS DE NOMBRE, EMAIL, CONSULTA.  |
|  | 5     | EL USUARIO LLENA LOS CAMPOS DEL FORMULARIO Y ENVÍA   |
|  | 6     | EL SISTEMA ENVIA DICHA CONSULTA AL CORREO DEL ADMINISTRADOR Y A SU VEZ ARORJA EL MENSAJE YA SEA DE ERROR O DE QUE TODO SE ENVIÓ CORRECTAMENTE Y PRONTO EL ADMINISTRADOR DE LA FERRETERIA |
| FLUJO ALTERNAT   | PASOS |  |
|  | 2.1   | EL SISTEMA PRESENTA FALLA Y NO MUESTRA LA PÁGINA DE INICIO.  |
|  | 4.1   | EL SISTEMA NO MUESTRA EL FORMULARIO DE CONSULTA  |
|  | 6.1   | EL SISTEMA NO LOGRA PROCESAR Y ENVIAR LA SOLICITUD AL CORREO DEL ADMINISTRADOR   |
| POSTCONDICIÓN: EL USUARIO LOGRA ENVIAR SU SOLICITUD Y HACER LA CONSULTA LOGRANDO ASI QUE LA RESPUESTA A SUS DUDAS LE LLEGUE A SU CORREO Y PUEDA SEGUIR CON SU COMPRA DE MANERA TRANQUILA Y EFECTIVA. |       |  |

R04

NOMBRE DE CASO DE USO: INICIO DE SECCIÓN

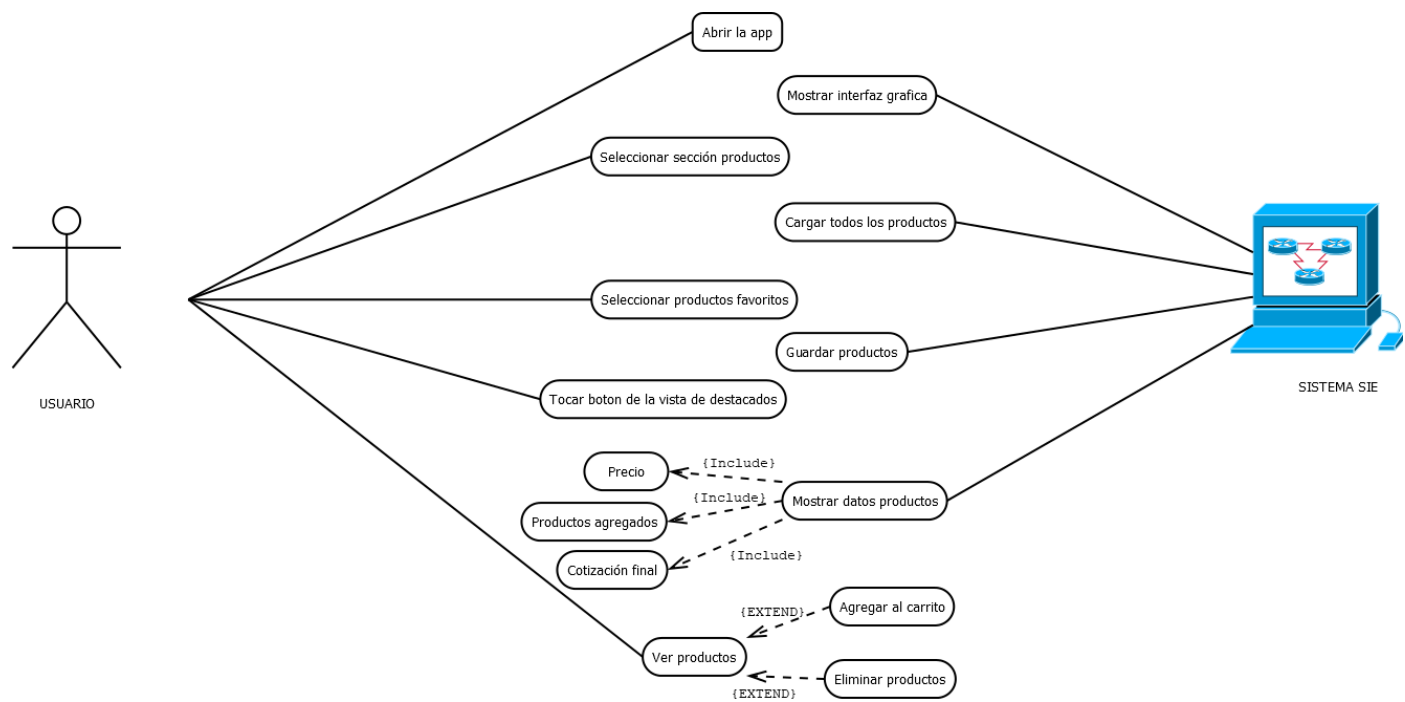


|   |       |   |
|---|-------|---|
| FECHA : 27/09/2025  |       |   |
| NOMBRE DEL CASO DE USOS: INICIO DE SECCIÓN  |       |   |
| CASO DE USO 04  |       |   |
| AUTOR: ELBERTO PERTUZ   |       |   |
| ACTOR: USUARIO Y SISTEMA  |       |   |
| PRECONDICIÓN: EL USUARIO DEBE TENER CONEXIÓN A INTERNET, EL USUARIO DEBE TENER LA APP DESCARGADA YA SEA EN SU TELEFONO MÓVIL O TABLETA.                               |       |   |
| FLUJO NORMAL  | PASOS | DESCRIPCIONES   |
|   | 1.    | EL USUARIO ENTRA A LA APLICACIÓN S.I.E  |
|   | 2.    | EL SISTEMA MUESTRA PÁGINA DE INICIO   |
|   | 3.    | EL USUARIO TOCA EL BOTÓN PARA INICIAR SECCIÓN   |
|   | 4.    | EL SISTEMA MUESTRA EL FORMULARIO PIDIENDO USUARIO Y CONTRASEÑA  |
|   | 5.    | EL USUARIO SE LOGEA YA SEA DESPUES DE CREAR SU CUENTA SI NO TIENE O CAMBIAR LA CONTRASEÑA DE HABERSELA OLVIDADO |
|   | 6.    | EL SISTEMA VUELVE SA MOSTRAR LA GUI DEL HOME PERO YA CON LA SECCIÓN INICIADA                                    |
|   | 7.    | EL USUARIO SE PASEA POR EL APLICATIVO A SU GUSTO CON TODAS LAS FUNCIONES DE USUARIO DISPONIBLES.                |
| FLUJO ALTERNATIVO   | PASOS |   |
|   | 2.1   | EL SISTEMA PRESENTA FALLA Y NO MUESTRA LA PÁGINA DE INICIO.   |
|   | 4.1   | EL SISTEMA NO MUESTRA EL FORMULARIO PARA INICIAR SECCIÓN  |
|   | 6.1   | EL SISTEMA NO ARROJA EL HOME NI TRAE RESPUESTA.   |
| POSTCONDICIÓN: EL USUARIO LOGRA ENVIAR SU SOLICITUD Y HACER LA CONSULTA LOGRANDO ASI QUE LA RESPUESTA A SUS DUDAS LE LLEGUE A SU CORREO Y PUEDA SEGUIR CON SU COMPRA. |       |   |



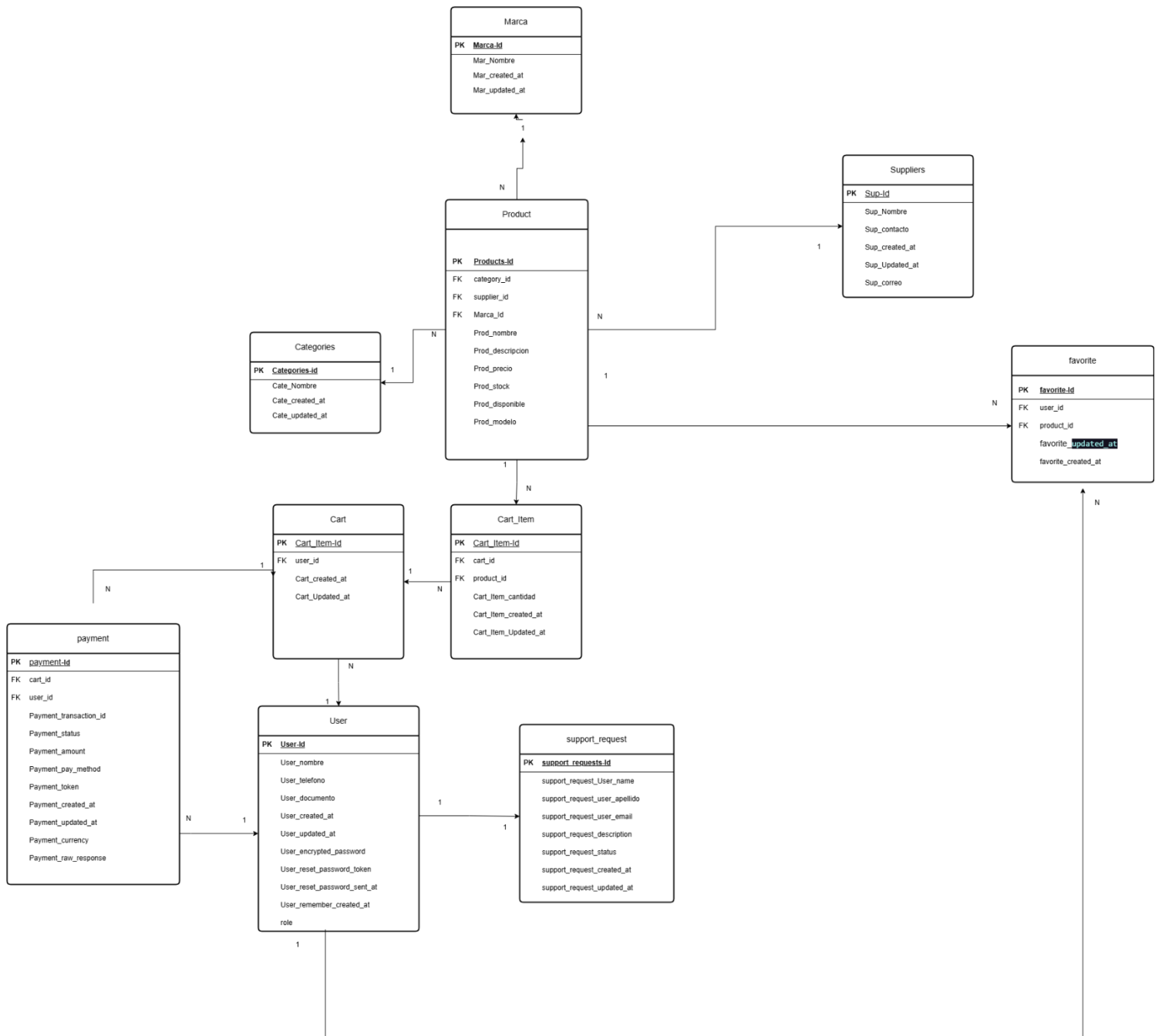
R5

NOMBRE DE CASO DE USO: AGREGAR PRODUCTOS A FAVORITO



|  |       |  |
|--|-------|--|
| FECHA : 27/09/2025   |       |  |
| NOMBRE DEL CASO DE USOS: AGREGAR PRODUCTOS A FAVORITO  |       |  |
| CASO DE USO 05   |       |  |
| AUTOR: ELBERTO PERTUZ  |       |  |
| ACTOR: USUARIO Y SISTEMA   |       |  |
| PRECONDICIÓN: EL USUARIO DEBE TENER CONEXIÓN A INTERNET, EL USUARIO DEBE TENER LA APP DESCARGADA YA SEA EN SU TELEFONO MOVIL O TABLETA.                            |       |  |
| FLUJO NORMAL   | PASOS | DESCRIPCIONES  |
|  | 1     | EL USUARIO ENTRA A LA APLICACIÓN S.I.E   |
|  | 2     | EL SISTEMA MUESTRA PÁGINA DE INICIO  |
|  | 3     | EL USUARIO SELECCIONA LA OPCIÓN DE VER PRODUCTOS   |
|  | 4     | EL SISTEMA LES MUESTRA TODOS LOS PRODUCTOS EN STOCK DE LA FERRETERIA CON SU NOMBRE, DESCRIPCIÓN, PRECIO, IMAGEN.       |
|  | 5     | SELECCIONA PRODUCTOS DESEADOS Y LOS AGREGA A FAVORITOS AL TOCARLE LA ESTRELLITA  |
|  | 6     | EL SISTEMA GUARDA TODOS Y CADA UNO DE LOS PRODUCTOS SELECCIONADOS A LA VISTA DE DESTACADOS RELACIONADO CON ESE USUARIO |
|  | 7     | EL USUARIO TOCA EL BOTON DE FAVORITO   |
|  | 8     | EL SISTEMA MUESTRA LA VISTA FSVORITOS CON LOS PRODUCTOS RECIENTE AGREGADOS.  |
|  | 9     | EL USUARIO CUANDO DESEE PUEDE VER SOLO LOS SELECCIONADOS Y AGREGARLOS AL CARRITO, ELIMINARLOS O SOLO DEJARLOS AHÍ.     |
| FLUJO ALTERNAT   | PASOS |  |
|  | 2.1   | EL SISTEMA PRESENTA FALLA Y NO MUESTRA LA PÁGINA DE INICIO.  |
|  | 4.1   | EL SISTEMA NO CARGA PRODUCTOS  |
|  | 6.1   | EL SISTEMA NO GUARDA TODOS LOS PRODUTOS SELECCIONADOS  |
|  | 8.1   | NO CARGA LA VISTA FAVORITOS.   |
| POSTCONDICIÓN: El usuario puede ver la cotización final de sus productos para hacerse un presupuesto y puede deiodir comprar dichos productos o solo dejarlos ahí. |       |  |

#### 4. Modelo relacional de la base de datos



## DICCIONARIO DE DATOS:

### cart\_items

| LLAVE | NOMBRE                | TIPO DE DATO               | LONGITUD                      | DESCRIPCIÓN   |
|-------|-----------------------|----------------------------|-------------------------------|---|
| PK    | cart_items_id         | integer<br>(autoincrement) | ---                           | Identificador único de la nueva tabla de la relación      |
| FK    | cart_id               | Integer                    | ---                           | Identificador único del carrito                           |
| FK    | product_id            | Integer                    | ---                           | Identificador único del producto                          |
|       | cart_items_Cantidad   | Integer                    | ---                           | Total de ítems en el carrito de ese usuario en específico |
|       | cart_items_created_at | datetime                   | tamaño de almacenamiento fijo | Registro de cuando se hizo el registro.                   |
|       | cart_items_updated_at | datetime                   | tamaño de almacenamiento fijo | Registro en fecha y hora de cuando se edita, si las hay.  |

### carts

| LLAVE | NOMBRE          | TIPO DE DATO               | LONGITUD                      | DESCRIPCIÓN  |
|-------|-----------------|----------------------------|-------------------------------|--|
| PK    | Id_carts        | integer<br>(autoincrement) | ---                           | Identificador único del carrito                          |
| FK    | user_id         | Integer                    | ---                           | Identificador único del usuario                          |
|       | Cart_created_at | datetime                   | tamaño de almacenamiento fijo | Fecha de cuando se hizo el registro.                     |
|       | Cart_updated_at | datetime                   | tamaño de almacenamiento fijo | Registro en fecha y hora de cuando se edita, si las hay. |

## Categories

| LLAVE | NOMBRE                | TIPO DE DATO | LONGITUD                      | DESCRIPCIÓN  |
|-------|-----------------------|--------------|-------------------------------|--|
| PK    | Id_Categories         | Integer      | ---                           | Número que identifica la categoría                       |
|       | Categories_nombre     | String       | 50                            | Nombre de la categoría                                   |
|       | Categories_created_at | datetime     | tamaño de almacenamiento fijo | Fecha de cuando se creó el registro.                     |
|       | Categories_updated_at | datetime     | tamaño de almacenamiento fijo | Registro en fecha y hora de cuando se edita, si las hay. |

## Users

| LLAVE | NOMBRE               | TIPO DE DATO            | LONGITUD                      | DESCRIPCIÓN  |
|-------|----------------------|-------------------------|-------------------------------|--|
| PK    | Id_Users             | integer (autoincrement) | ---                           | Número que identifica al usuario                         |
|       | User_Nombre          | String                  | 60                            | Nombre de quien realizó la compra                        |
|       | User_Teléfono        | String                  | 60                            | Número de contacto del cliente                           |
|       | User_created_at      | datetime                | tamaño de almacenamiento fijo | Fecha de cuando se creó el registro.                     |
|       | User_updated_at      | datetime                | tamaño de almacenamiento fijo | Registro en fecha y hora de cuando se edita, si las hay. |
|       | User_email           | string                  | 50                            | Dirección de correo electrónico del usuario registrado   |
|       | encrypted_password   | string                  | 255                           | Contraseña encriptada                                    |
|       | reset_password_token | string                  | 100                           | Token de recuperación                                    |

|  |           |        |    |                        |
|--|-----------|--------|----|------------------------|
|  | User_Role | string | 20 | Rol (User, admin, etc) |
|--|-----------|--------|----|------------------------|

#### Favorites

| LLAVE | NOMBRE      | TIPO DE DATO | LONGITUD                      | DESCRIPCIÓN                      |
|-------|-------------|--------------|-------------------------------|----------------------------------|
| PK    | Favorite_Id | Integer      | ---                           | Identificador único del favorito |
| FK    | product_id  | Integer      | ---                           | Identificador único del producto |
| FK    | user_id     | Integer      | ---                           | Identificador único del usuario  |
|       | created_at  | Datetime     | tamaño de almacenamiento fijo | Fecha de creación del registro   |
|       | updated_at  | Datetime     | tamaño de almacenamiento fijo | Fecha de última actualización    |

#### Marcas

| LLAVE | NOMBRE           | TIPO DE DATO            | LONGITUD                      | DESCRIPCIÓN                     |
|-------|------------------|-------------------------|-------------------------------|---------------------------------|
| PK    | Marca_Id         | integer (autoincrement) | ---                           | Identificador único de la marca |
|       | Marca_nombre     | String                  | 60                            | Nombre de la marca del producto |
|       | Marca_created_at | Datetime                | tamaño de almacenamiento fijo | Fecha de creación del registro  |
|       | Marca_updated_at | Datetime                | tamaño de almacenamiento fijo | Fecha de última actualización   |

#### Payments

| LLAVE | NOMBRE | TIPO DE DATO | LONGITUD | DESCRIPCIÓN |
|-------|--------|--------------|----------|-------------|
|-------|--------|--------------|----------|-------------|

|    |                  |                         |   |   |
|----|------------------|-------------------------|---|---|
| PK | Pay_Id           | integer (autoincrement) | ----  | Identificador único del pago  |
| FK | cart_id          | Integer                 | ----  | ID del carrito asociado (relación con tabla carts)                                |
| FK | user_id          | Integer                 | ----  |   |
|    | transaction_id   | String                  | 50  | ID único generado por Wompi   |
|    | status           | integer (enum)          | ----  | Estado del pago (pending, approved, failed, paid)                                 |
|    | Amount           | Decima                  | precision: 12, scale: 2                         | Monto total de la transacción. Hasta 999,999,999,999.99 COP                       |
|    | pay_method       | String                  | 30  | Tipo de pago/ método de pago (CARD, PSE, NEQUI, etc)                              |
|    | Pay_token        | String                  | 100   | Token temporal del pago.  |
|    | Pay_Currency     | String                  | 3   | Código de moneda ISO 4217   |
|    | Wompi_Id         | String                  | 50  | ID único del pago dentro de wompi.  |
|    | Pay_raw_response | Json                    | estructura variable según respuesta de pasarela | Respuesta completa en formato JSON de la pasarela de pago                         |
|    | Pay_account_info | String                  | 150   | Información asociada a la cuenta o método de pago (últimos dígitos, nombre, etc.) |
|    | Pay_created_at   | Datetime                | tamaño de almacenamiento fijo                   | Fecha de creación del registro  |

|  |                |          |                               |                               |
|--|----------------|----------|-------------------------------|-------------------------------|
|  | Pay_updated_at | Datetime | tamaño de almacenamiento fijo | Fecha de última actualización |
|--|----------------|----------|-------------------------------|-------------------------------|

products

| LLAVE | NOMBRE          | TIPO DE DATO | LONGITUD                      | DESCRIPCIÓN                             |
|-------|-----------------|--------------|-------------------------------|---|
| PK    | Products_Id     | Integer      | ----                          | codigo_producto                         |
|       | category_id     | Integer      | ----                          |   |
|       | supplier_id     | Integer      | ----                          |   |
|       | marca_id        | Integer      | ----                          |   |
|       | Pro_nombre      | String       | 100                           | Nombre del producto                     |
|       | Pro_descripcion | String       | 200                           | Descripción breve                       |
|       | Pro_precio      | decimal      | 12,2                          | Precio del producto                     |
|       | Pro_stock       | Integer      | ----                          | Cantidad en la tienda de dicho producto |
|       | Pro_modelo      | String       | 50                            | Modelo o referencia                     |
|       | Pro_disponible  | Boolean      | ----                          | Indica si está disponible               |
|       | Pro_created_at  | Datetime     | tamaño de almacenamiento fijo | Fecha de creación del registro          |
|       | Pro_updated_at  | Datetime     | tamaño de almacenamiento fijo | Fecha de última actualización           |

user\_supports

| LLAVE | NOMBRE  | TIPO DE DATO | LONGITUD | DESCRIPCIÓN |
|-------|---------|--------------|----------|-------------|
| PK    | User_Id |              | ---      |             |

|  |                    |                            |                                     |  |
|--|--------------------|----------------------------|-------------------------------------|--|
|  |                    | integer<br>(autoincrement) |                                     | Identificador<br>único del<br>usuario                |
|  | User_<br>User_name | String                     | 100                                 | Nombre del<br>usuario que<br>envía la<br>solicitud   |
|  | user_apellido      | string                     | 100                                 | Apellido del<br>usuario que<br>envía la<br>solicitud |
|  | user_email         | String                     | 100                                 | Correo del<br>usuario                                |
|  | User_description   | Text                       | ----                                | Descripción<br>detallada del<br>problema             |
|  | User_status        | String                     | 30                                  | Estado<br>(pendiente,<br>resuelto, etc.)             |
|  | Pay_created_at     | Datetime                   | tamaño de<br>almacenamiento<br>fijo | Fecha de creación<br>del registro                    |
|  | Pay_updated_at     | Datetime                   | tamaño de<br>almacenamiento<br>fijo | Fecha de<br>última<br>actualización                  |

## Suppliers

| LLAVE | NOMBRE               | TIPO DE<br>DATO | LONGITUD | DESCRIPCIÓN                         |
|-------|----------------------|-----------------|----------|-------------------------------------|
| PK    | Sup_Id               | Integer         | 50       |                                     |
|       | Sup_nombre           | string          | 100      | Nombre del<br>proveedor             |
|       | Sup_contacto         | string          | 100      | Nombre de<br>contacto               |
|       | Sup_codigo_proveedor | string          | 30       | Código interno<br>del proveedor     |
|       | Sup_correo           | string          | 100      | Correo electrónico del<br>proveedor |



|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|

## 5. descripción de la plataforma:

La aplicación móvil para la ferretería Ferremateriales está desarrollada utilizando Flutter y Dart, permitiendo crear una experiencia nativa y fluida tanto en dispositivos Android como iOS con una sola base de código.

El móvil se conecta a un backend basado en Ruby on Rails (versión 7.x), que funciona como una API RESTful para manejar la lógica de negocio y la gestión de datos. Este backend mantiene la arquitectura MVC tradicional y utiliza PostgreSQL como base de datos en producción.

En el móvil, la interfaz de usuario está diseñada para ser intuitiva, responsiva y adaptada a pantallas pequeñas, facilitando la navegación y consulta rápida de productos, promociones y funcionalidades para usuarios finales como:

- Visualización pública de productos, con filtros por categoría, nombre.
- Acceso al carrito de compras.
- Consulta de promociones en un carrusel interactivo.
- Funcionalidades de favoritos para gestionar productos de interés.
- Formulario de contacto para soporte o ayuda.

La app móvil se comunica con el servidor mediante APIs seguras, autenticando al usuario cuando es necesario, garantizando privacidad y seguridad en la gestión de sesiones.

El uso de Flutter permite que la interfaz sea moderna y fluida, con componentes visuales como listas dinámicas, modales, y navegación sencilla, optimizada para el rendimiento en dispositivos móviles.

La aplicación móvil está enfocada exclusivamente en la experiencia del usuario final, sin acceso a las funciones administrativas o de gestión que se mantienen en el panel web.

En cuanto al framework principal como ya se mencionó se utilizó flutter un framework gratuito y de código abierto que permite crear aplicaciones multiplataforma con un único código base. Flutter utiliza el lenguaje de programación Dart y construimos interfaces de usuario atractivas y personalizadas mediante un sistema de widgets.

Sistema de gestión de bases de datos (SGBD) ... (en producción) / ... (en desarrollo local): Base de datos relacional utilizada para almacenar información de productos, compras, ventas, proveedores, usuarios, etc.

Frontend (interfaz de usuario): widgets

Se utilizaron componentes dinámicos como tarjetas, carrusel, botones flotantes, formularios interactivos, etc.

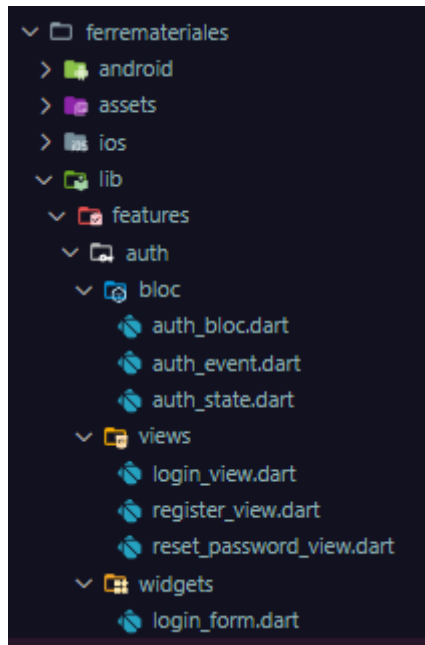
Sistema operativo compatible: Windows 10/11 para desarrollo local

Control de versiones

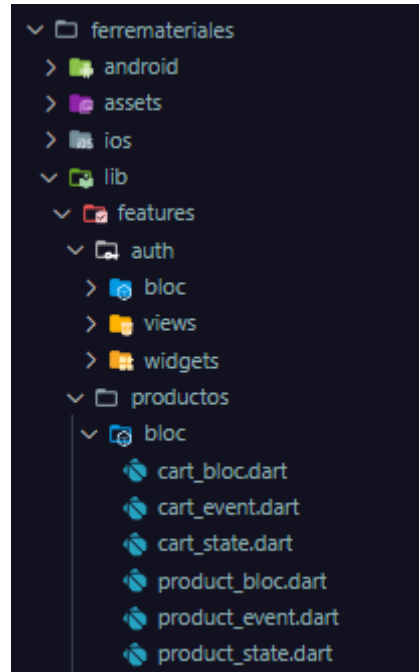
GitHub: Almacenamiento del código fuente y colaboración en el desarrollo.

## 6. Documentación del código fuente

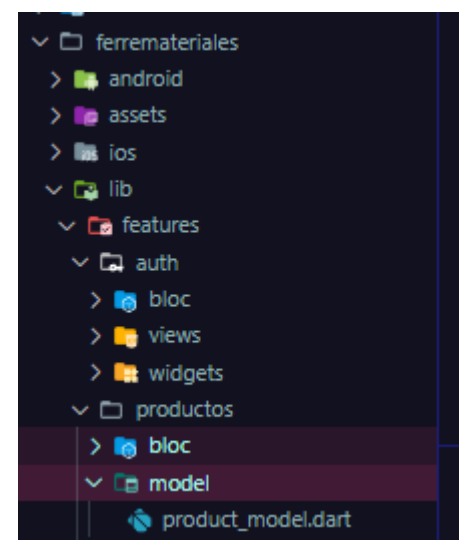
### Inicio de sección



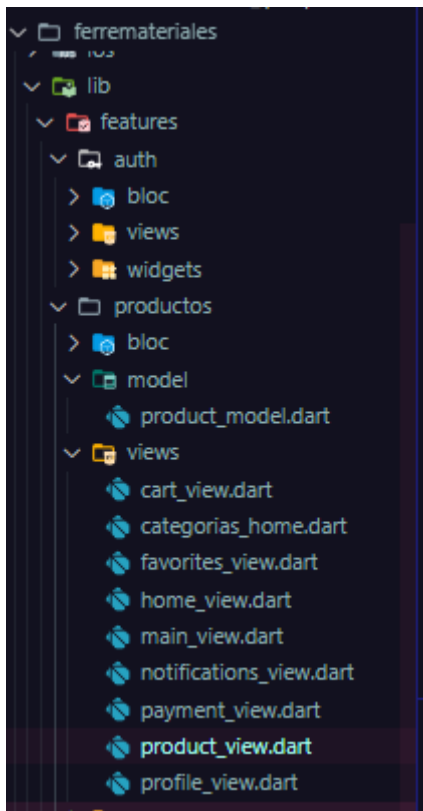
### Productos block



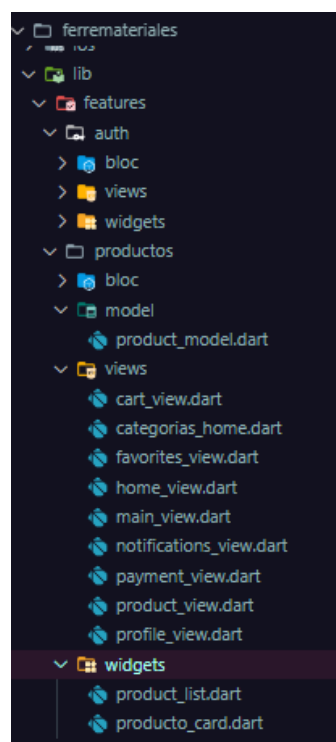
### Productos modelos



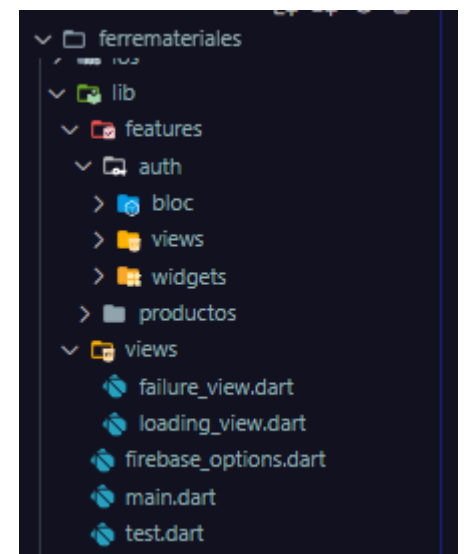
### Vistas



### Widget producto



### Vistas de estados



## 2. Estructura de directorios:

| DIRECTORIO                                    | DESCRIPCIÓN  |
|---|--|
| Ferremateriales/lib/features/auth/bloc        | Block de inicio de sección   |
| Ferremateriales/lib/features/auth/viws        | Vistas del inicio de sección   |
| Ferremateriales/lib/features/auth/widget      | Inicio de sección  |
| Ferremateriales/lib/features/productos/bloc   | Bloc de productos donde se hace la petición para mostrar todos los productos |
| Ferremateriales/lib/features/productos/models | Modelo del producto  |

## Login

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_bloc/flutter_bloc.dart';
3 import 'package:google_fonts/google_fonts.dart';
4 import '../bloc/auth_bloc.dart';
5 import '../bloc/auth_state.dart';
6 import '../widgets/login_form.dart';
7 import '../productos/views/home_view.dart';
8 import 'register_view.dart';
9 import 'reset_password_view.dart';
10
11 class LoginView extends StatelessWidget {
12   @override
13   Widget build(BuildContext context) {
14     return BlocListener<AuthBloc, AuthState>(
15       listenWhen: (previous, current) => current.status == AuthStatus.success,
16       listener: (context, state) {
17         Navigator.pushReplacement(
18           context,
19           MaterialPageRoute(builder: (_) => HomeView()),
20         );
21       },
22       child: Scaffold(
23         body: Container(
24           decoration: BoxDecoration(
25             image: DecorationImage(
26               image: AssetImage('assets/images/degrade.png'),
27               fit: BoxFit.cover,
28             ),
29           ),
30           child: Center(
31             child: SingleChildScrollView(
32               padding: const EdgeInsets.symmetric(horizontal: 24.0),
33               child: Column(
34                 mainAxisAlignment: MainAxisAlignment.center,
35                 children: [
36                   // Logo usuario
37                   CircleAvatar(
38                     backgroundColor: Colors.orange,
39                     radius: 40,
```

```
40       child: Icon(Icons.person, color: Colors.white, size: 40),
41     ),
42     const SizedBox(height: 16),
43
44     // título
45     Text(
46       'Inicia sesión para continuar',
47       style: GoogleFonts.montserrat(
48         fontSize: 28,
49         fontWeight: FontWeight.bold,
50         color: Color.fromARGB(255, 219, 222, 227),
51       ),
52       textAlign: TextAlign.center,
53     ),
54     // const SizedBox(height: 10),
55     // Text(
56     //   'Inicia sesión para continuar',
57     //   style: GoogleFonts.montserrat(
58     //     fontSize: 16,
59     //     color: Colors.grey[200], // más contraste con fondo oscuro
60     //   ),
61     // ),
62     const SizedBox(height: 30),
63
64     // LoginForm sin tarjeta
65     LoginForm(),
66
67     const SizedBox(height: 16),
68
69     // Botón de registrarse
70     TextButton(
71       onPressed: () {
72         Navigator.push(
73           context,
```

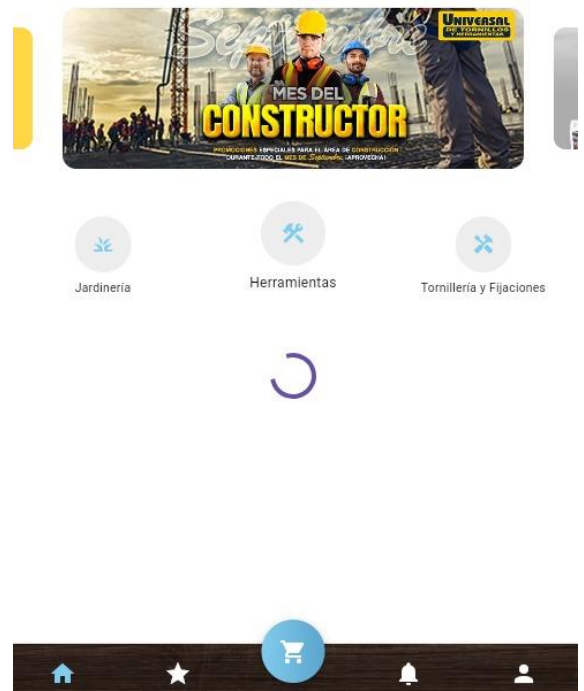
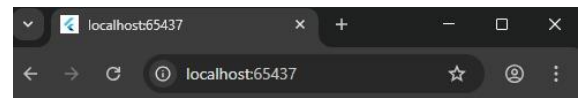


## Inicio/index

```
15 class _HomeViewState extends State<HomeView> {
24   @override
25   Widget build(BuildContext context) {
26     final bannerImages = [
27       'assets/images/oferta.jpg',
28       'assets/images/oferta2.png',
29       'assets/images/oferta3.jpg',
30     ];
31     final categories = [
32       {'icon': Icons.construction, 'label': 'Herramientas'},
33       {'icon': Icons.handyman, 'label': 'Tornillería y Fijaciones'},
34       {'icon': Icons.plumbing, 'label': 'Plomería'},
35       {'icon': Icons.flash_on, 'label': 'Electricidad'},
36       {'icon': Icons.business, 'label': 'Construcción y Materiales'},
37       {'icon': Icons.format_paint, 'label': 'Pintura y Acabados'},
38       {'icon': Icons.home_repair_service, 'label': 'Ferretería para el hogar'},
39       {'icon': Icons.cleaning_services, 'label': 'Limpieza y Mantenimiento'},
40       {'icon': Icons.sticky_note_2, 'label': 'Adhesivos y Selladores'},
41       {'icon': Icons.grass, 'label': 'Jardinería'},
42     ];
43     final categoryItems =
44       categories.map((category) {
45         return _HoverCategoryButton(
46           icon: category['icon'] as IconData,
47           label: category['label'] as String,
48           onTap: () {
49             print('Clic en: ${category['label']}');
50           },
51         );
52       }).toList();
53   }
54   return Scaffold(
55     backgroundColor: Colors.white,
56     body: SafeArea(
57       child: SingleChildScrollView(
58         padding: const EdgeInsets.only(bottom: 80),
59         child: Column(
60
```

```
62     children: [
63       const SizedBox(height: 24),
64       // Carrusel de imágenes
65       CarouselSlider(
66         options: CarouselOptions(
67           height: 140,
68           autoplay: true,
69           enlargeCenterPage: true,
70         ),
71       ),
72       items:
73         bannerImages.map((path) {
74           return Builder(
75             builder: (BuildContext context) {
76               return ClipRect(
77                 borderRadius: BorderRadius.circular(12),
78                 child: Image.asset(
79                   path,
80                   fit: BoxFit.cover,
81                   width: double.infinity,
82                 ),
83               );
84             },
85           );
86         }).toList(),
87     ],
88
```

```
67     options: CarouselOptions(
68       height: 140,
69       autoplay: true,
70       enlargeCenterPage: true,
71     ),
72     items:
73       bannerImages.map((path) {
74         return Builder(
75           builder: (BuildContext context) {
76             return ClipRect(
77               borderRadius: BorderRadius.circular(12),
78               child: Image.asset(
79                 path,
80                 fit: BoxFit.cover,
81                 width: double.infinity,
82               ),
83             );
84           },
85         );
86       }).toList(),
87   ),
88   const SizedBox(height: 28),
89   // Carrusel de categorías
90   CarouselSlider(
91     options: CarouselOptions(
92       height: 110,
93       viewportFraction: 0.33,
94       enableInfiniteScroll: true,
95       autoplay: true,
96       autoplayInterval: const Duration(seconds: 3),
97       enlargeCenterPage: true,
98     ),
99   ),
100
```



## Carro de compras

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_bloc/flutter_bloc.dart';
3 import '../bloc/cart_bloc.dart';
4 import '../bloc/cart_event.dart';
5 import '../bloc/cart_state.dart';
6 import 'payment_view.dart';
7
8 class CartView extends StatelessWidget {
9   const CartView({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return BlocBuilder<CartBloc, CartState>(
14       builder: (context, state) {
15         final totalItems = state.items.fold<int>(
16           0,
17           (sum, item) => sum + (item["quantity"] as int),
18         );
19
20         return Scaffold(
21           appBar: AppBar(
22             title: Text("Mi Carrito ($totalItems)"),
23             centerTitle: true,
24           ),
25           body: state.items.isEmpty
26             ? const Center(child: Text("Tu carrito está vacío 🛒"))
27             : Column(
28               children: [
29                 Expanded(
30                   child: ListView.builder(
31                     itemCount: state.items.length,
32                     itemBuilder: (context, index) {
33                       final item = state.items[index];
34                       return Card(
35                         margin: const EdgeInsets.symmetric(
36                           horizontal: 8,
37                           vertical: 4,
38                         ),
```

```
39         child: ListTile(
40           leading: const Icon(Icons.shopping_bag),
41           title: Text(item["name"]),
42           subtitle: Row(
43             children: [
44               // Botón restar
45               IconButton(
46                 icon: const Icon(
47                   Icons.remove_circle_outline,
48                   color: Colors.orange,
49                 ),
50                 onPressed: () {
51                   context
52                     .read<CartBloc>()
53                     .add(DecreaseQuantity(item["name"]));
54                 },
55             ),
56             // Cantidad
57             Text(
58               "${item["quantity"]}",
59               style: const TextStyle(fontSize: 16),
60             ),
61             // Botón sumar
62             IconButton(
63               icon: const Icon(
64                 Icons.add_circle_outline,
65                 color: Colors.green,
66               ),
67               onPressed: () {
68                 context
69                   .read<CartBloc>()
70                   .add(IncreaseQuantity(item["name"]));
71               },
72             ),
73           ),
```

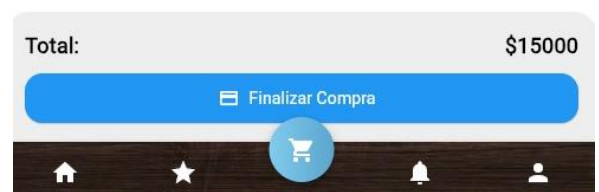
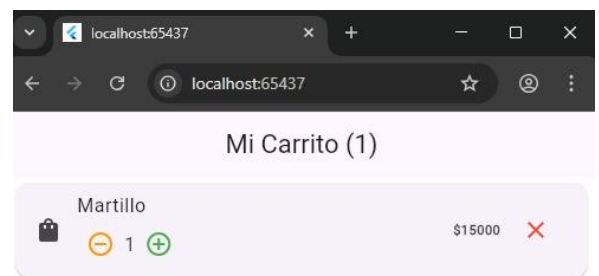
```
74         ),
75         trailing: Row(
76           mainAxisAlignment: MainAxisAlignment.min,
77           children: [
78             Text(
79               "\${item["price"] * item["quantity"]}",
80               style: const TextStyle(
81                 fontWeight: FontWeight.bold,
82               ),
83             ),
84             const SizedBox(width: 10),
85             IconButton(
86               icon: const Icon(Icons.close,
87                 color: Colors.red,
88               ),
89               onPressed: () {
90                 context
91                   .read<CartBloc>()
92                   .add(RemoveFromCart(item["name"]));
93               },
94             ),
95           ],
96         ),
97       ],
98     ),
99   ),
100 ],
101 // Footer con el total
102 Container(
103   padding: const EdgeInsets.all(16),
104   decoration: BoxDecoration(
105     color: Colors.grey.shade200,
```

```
134   onPressed: () {
135     // 🏠 Navegamos a la vista de pago
136     Navigator.push(
137       context,
138       MaterialPageRoute(
139         builder: (_) => PaymentView(total: state.total),
140       ),
141     );
142   },
143   icon: const Icon(Icons.payment),
144   label: const Text("Finalizar Compra"),
145   style: ElevatedButton.styleFrom(
146     backgroundColor: const Color(0xFF2196F3),
147     foregroundColor: Colors.white,
148     minimumSize: const Size(double.infinity, 50),
149     shape: RoundedRectangleBorder(
150       borderRadius: BorderRadius.circular(15),
151     ),
152   ),
153 ),
154 ],
155 ),
156 ),
157 ),
158 ),
159 ),
160 ),
161 ),
162 ],
163 ),
164 ),
```

```

106         borderRadius: const BorderRadius.only(
107           topLeft: Radius.circular(20),
108           topRight: Radius.circular(20),
109         ),
110       ),
111       child: Column(
112         children: [
113           Row(
114             mainAxisAlignment: MainAxisAlignment.spaceBetween,
115             children: [
116               const Text(
117                 "Total:",
118                 style: TextStyle(
119                   fontSize: 18,
120                   fontWeight: FontWeight.bold,
121                 ),
122               ),
123               Text(
124                 "\${state.total}",
125                 style: const TextStyle(
126                   fontSize: 18,
127                   fontWeight: FontWeight.bold,
128                 ),
129               ),
130             ],
131           ),
132           const SizedBox(height: 10),
133           ElevatedButton.icon(
134             onPressed: () {
135               // 🚀 Navegamos a la vista de pago
136               Navigator.push(
137                 context,
138                 MaterialPageRoute(
139                   builder: (_) => PaymentView(total: state.total),

```



## **Descripción de los acuerdos de niveles de servicios o ANS**

NO APLICA Toda esta parte es manejada desde el aplicativo web.

### **8. Administración de usuarios**

Creación de roles y perfiles de usuario:

El sistema actualmente solo cuenta con un único rol: el del usuario

No requieren autenticación, pueden: Navegar el catálogo de productos, enviar mensajes mediante el formulario de contacto y también puede solo si lo desea registrarse para tener una cuenta propia en donde podrá administrar su información, hacer pedidos y demás.

## **GLOSARIO:**

- **Framework:** Un framework, o marco de trabajo, es un conjunto de herramientas, bibliotecas, estructuras y convenciones estandarizadas que proporciona una base para desarrollar software o aplicaciones, simplificando el proceso y evitando empezar desde cero.
- **Frontend:** Parte de un sitio web o aplicación con la que los usuarios interactúan directamente, es decir, todo lo que el usuario ve y experimenta en su navegador o dispositivo
- **Backend:** es la parte "detrás de escena" de una aplicación o sitio web, responsable de la lógica, el funcionamiento y los datos que no son visibles para el usuario final. Se encarga de las tareas cruciales como la gestión de bases de datos, el procesamiento de información, la seguridad, la autenticación de usuarios y la comunicación con el servidor
- **MVC:** (Modelo-Vista-Controlador) es un patrón de arquitectura que divide una aplicación en tres componentes principales para una mejor organización del código y un desarrollo más mantenible. El Modelo gestiona los datos y la lógica de negocio, la Vista se encarga de la interfaz de usuario y de presentar la información, y el Controlador actúa como intermediario, enrutando las solicitudes del usuario al Modelo y luego a la Vista. MVC:
- **PostgreSQL:** sistema avanzado de gestión de bases de datos relacionales (RDBMS) de código abierto que se utiliza para almacenar, organizar y gestionar grandes cantidades de datos de manera confiable y eficiente. Soporta tanto datos estructurados como no estructurados (como JSON), ofrece características de alto rendimiento, escalabilidad, y es compatible con los principales sistemas operativos y lenguajes de programación.
- **API:** Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y protocolos que permite a diferentes aplicaciones de software comunicarse entre sí, intercambiar datos y utilizar funcionalidades. Actúa como un intermediario o puente que define cómo se realizan las solicitudes y las respuestas, simplificando el desarrollo al permitir a los programadores usar funciones existentes en lugar de crearlas desde cero.
- **BASE DE DATOS:** colección organizada y estructurada de información, o datos, que se almacena y gestiona electrónicamente en un sistema informático, controlada generalmente por un Sistema de Gestión de Bases de Datos (SGBD). Las bases de datos en programación permiten a los desarrolladores almacenar, consultar, modificar y administrar de forma eficiente grandes volúmenes de datos para crear aplicaciones, analizar información o tomar decisiones informadas.
- **BASE DE DATOS RELACIONAL:** es un sistema que organiza y almacena datos en una o más tablas (llamadas también "relaciones") compuestas por filas y columnas. Permite relacionar datos entre tablas, lo que facilita la búsqueda y el análisis de la información, utilizando un lenguaje estandarizado llamado SQL para interactuar con los datos.
- **MODELO DE DATOS:** representación visual de un sistema de información, que define las estructuras de datos, los atributos que contienen y las relaciones



entre ellas, facilitando la organización, comprensión y gestión de la información para el desarrollo de sistemas y bases de datos.

- **UML:** (Lenguaje Unificado de Modelado) es un lenguaje gráfico estandarizado y de propósito general que se usa para especificar, visualizar, construir y documentar sistemas de software, no para programar directamente. A través de sus diferentes diagramas (estructurales y de comportamiento), permite modelar la arquitectura, las relaciones y el comportamiento de un sistema, facilitando la comprensión y comunicación de diseños complejos entre los desarrolladores y otros miembros del equipo.
- **DIAGRAMA DE CLASE:** Representación visual, utilizando el lenguaje UML (Lenguaje Unificado de Modelado), que muestra la estructura estática de un sistema de software, detallando sus clases, los atributos (datos) y operaciones (métodos) que definen a cada clase, así como las relaciones (como herencia, asociación o composición) que existen entre ellas. Es una herramienta fundamental en la ingeniería de software para entender, diseñar y comunicar la arquitectura de un sistema de forma clara y concisa.
- **DIAGRAMA DE CASO DE USO:** Herramienta de modelado visual que muestra las interacciones entre los usuarios (actores) y las funcionalidades (casos de uso) de un sistema
- **FICHA DE CASO DE USO:** Documento o plantilla textual que describe una funcionalidad específica de un sistema, detailing cómo un actor (usuario o sistema externo) interactúa con el sistema para lograr un objetivo definido.