

UNIVERSIDAD DEL ROSARIO



**Universidad del
Rosario**

**FORMULACIÓN Y DESARROLLO DE LA BASE DE DATOS PARA LA EMPRESA
PALACIO DEL TIEMPO**

**Mariana Álvarez Carvajal
Daniel Alejandro Moreno Bravo
Lina Stefanny Peña Cubillos
Sergio Alejandro Velandia Rodriguez**

9 de abril de 2025

TABLA DE CONTENIDOS

| | |
|--|-----------|
| Resumen Ejecutivo | 4 |
| Contextualización y Necesidad | 4 |
| Definición Problema | 4 |
| Objetivos | 4 |
| Objetivo general | 4 |
| Objetivos específicos | 5 |
| Alternativas de Solución | 5 |
| Soluciones existentes | 5 |
| Posibles soluciones iniciales: | 6 |
| I. Primera solución: Propuesta por Mariana Álvarez | 6 |
| II. Segunda Solución: Propuesta Daniel Moreno | 7 |
| III. Tercera Solución: Propuesta Lina Peña | 8 |
| IV. Cuarta Solución: Propuesta Sergio Velandia | 9 |
| Alternativa de solución seleccionada | 11 |
| Alcance | 13 |
| Metodología | 13 |
| Levantamiento de información y RQFs | 14 |
| Requerimientos funcionales | 15 |
| Historias de Usuario | 31 |
| Mapa de StakeHolders | 50 |
| Modelamiento de base de datos SQL | 51 |
| Diagrama de clases: | 51 |
| A continuación, se presenta el diagrama de clases para el proyecto final. Su interpretación es la siguiente: | 51 |
| Modelo Conceptual: | 53 |
| El siguiente Modelo de Entidad Relación muestra las relaciones que se aplican entre los módulos establecidos con anterioridad. | 53 |
| Modelo Lógico | 54 |
| Modelo Físico | 54 |
| Diccionario de Datos | 55 |
| Script de procesos DDL y DML de la base de datos | 56 |
| Conclusiones por Fases de Desarrollo del Proyecto | 64 |
| Referencias | 65 |
| Anexo | 65 |
| Evidencia de comunicación con el cliente: WorkShop con el Cliente | 65 |

| | |
|---------------------------------------|-----------|
| INFORME REUNION | 68 |
| INFORME REUNIÓN CON EL CLIENTE | 68 |

Resumen Ejecutivo

En el presente informe, se aborda una problemática crítica de un cliente real a la cual, se propone una solución basada en la optimización de los procesos de la empresa, particularmente en el manejo de datos; esto afecta directamente la eficiencia y productividad en la atención al cliente.

A través del enfoque brindado durante las sesiones de clase, se busca proporcionar un sistema estructurado que facilite:

1. La gestión eficiente de la organización de los clientes.
2. La organización y agilización de los procesos de la empresa
3. Mejorar la experiencia con el cliente
4. La toma de decisiones de mejor calidad con respecto a la producción y a los clientes que más consumen.

De esta manera, garantizamos que el usuario pueda usar de manera efectiva la base de datos, y que pueda agilizar la organización a nivel interno de la empresa.

Contextualización y Necesidad

La empresa Palacio del Tiempo, liderada por los emprendedores Oswald y Patricia nos proponen la siguiente problemática: actualmente, los procesos y la toma de decisiones se realizan de forma manual y con un sistema digital que no cumple con todas sus necesidades, y teniendo en cuenta la gran cantidad de información generada diariamente, se presentan algunas dificultades como tomar decisiones con respecto a cuáles son los productos con más ventas, los procesos de búsqueda de información relacionada con cada cliente son muy demorados, el sistema de facturación no es muy eficiente, se duplica la carga laboral de las personal que trabajan en la empresa y se dificulta la gestión del stock disponible por producto, pues el software que ellos utilizan, es sumamente limitado y costoso. Además, han tenido inconvenientes con el funcionamiento y mantenimiento del servicio que ellos emplean.

Definición Problema

En este contexto surge la siguiente pregunta problema: ¿Cómo desarrollar un sistema de gestión de base de datos que permita a la empresa Palacios el Tiempo mejorar la gestión de clientes, stock y el proceso de facturación superando las limitaciones del software actual?

Objetivos

Objetivo general

Diseñar y construir una base de datos relacional y una base de datos no relacional para simplificar listas con datos de clientes para la empresa Palacio Del Tiempo, mostrando relaciones con productos, facturas electrónicas, cantidad de dinero, fechas de pago, inventario y demás.

Objetivos específicos

- *Fase de Análisis:* Identificar y analizar la necesidad de contar con un sistema de gestión de datos para clientes
- *Fase de Planeación:* Definir los requerimientos funcionales y no funcionales que debe cumplir nuestro proyecto, considerando tanto la parte relacional como la no relacional
 - Creación de diagrama de clases para representar las entidades principales y sus relaciones
- *Fase de Diseño:* Diseñar la base de datos relacional para almacenar toda la información relacionada a los clientes (productos, facturas, cantidades de dinero, fechas de pago...)
 - Proyectar la base de datos no relacional, especialmente para gestionar datos no estructurados como imágenes, documentos, o archivos privados de los clientes.
- *Fase de Desarrollo:* Implementar las bases de datos usando gestores como MySQL y MongoDB
- *Fase de Pruebas:* Realizar pruebas iniciales con volúmenes de datos moderados para asegurar que las BDs cumplen los requerimientos y que están alineadas a la necesidad del cliente.
- *Fase de Mantenimiento y Documentación:* Monitorear el sistema y proponer mejoras según el cliente

Alternativas de Solución

Soluciones existentes

A lo largo de diversos artículos se ha destacado la importancia de las bases de datos, tanto para instituciones como para empresas, que las utilizan con diversos objetivos. Estas bases de datos son fundamentales para mantener la información debidamente organizada y preservada. Como se menciona en el artículo "La importancia de contar con información precisa, confiable y oportuna en las bases de datos", contar con datos precisos, confiables y actualizados favorece y respalda la toma de decisiones estratégicas y operativas.

En el mismo artículo, se subraya cómo estos sistemas de información facilitan una gestión adecuada de los datos, permitiendo que los niveles jerárquicos medios y altos de las

organizaciones accedan a información relevante. De esta manera, se convierten en una herramienta crucial para los gerentes, ya que centralizan y articulan los datos, ofreciendo visibilidad de manera accesible, completa, segura, oportuna y verificable.

Por otro lado, el artículo académico "Análisis descriptivo de bases de datos relacional y no relacional" destaca la efectividad de las bases de datos no relacionales cuando el crecimiento de los datos ocurre en tiempo real, lo que exige proporcionar información procesada a partir de grandes volúmenes de datos.

En definitiva, las bases de datos son realmente versátiles y permiten gestionar y consultar la información de acuerdo con las necesidades específicas. En nuestro caso, donde uno de los objetivos es la facturación, un sistema de bases de datos es absolutamente necesario. De hecho, en "Análisis, diseño, implementación y gestión de un sistema de bases de datos para automatizar los procesos de facturación de la empresa 'OKI DOKI'", se resalta la relevancia del manejo de MySQL en el sistema de facturación propuesto y el éxito que se ha logrado con esta herramienta.

Una base de datos adecuada resolverá el problema planteado a raíz de la necesidad de nuestros clientes y optimizará el acceso a la información de su empresa.

Posibles soluciones iniciales:

Anteriormente, se presentaron diferentes soluciones planteadas por textos académicos que pueden implementarse dentro de nuestro contexto, sin embargo, dado que nuestra problemática tiene un diseño muy puntual, cada uno de los integrantes presentó una posible solución que se describirán a continuación:

I. Primera solución: Propuesta por Mariana Álvarez

Propone desarrollar un sistema de gestión que se centre en mejorar los procesos que están presentando inconvenientes dentro de la empresa. Para la creación de dicha interfaz, la idea es estructurarla de la siguiente manera:

- Se van a verificar los permisos del usuario que va a entrar a la base de datos mediante un Inicio de Sesión que solo permita ingresar a los empleados.
- Puede contar con 5 módulos: Clientes, Ventas, Factura, Stock y Flujo de Trabajo.
- La parte de Flujo de Trabajo está planeada para que funcione con el ideal de una ETL, donde se eliminan tareas repetidas, se eviten duplicaciones de datos y se limpie la información para que sea un proceso mucho más automático.

En la Figura 1 se presenta el diagrama de flujo que representa esta solución:

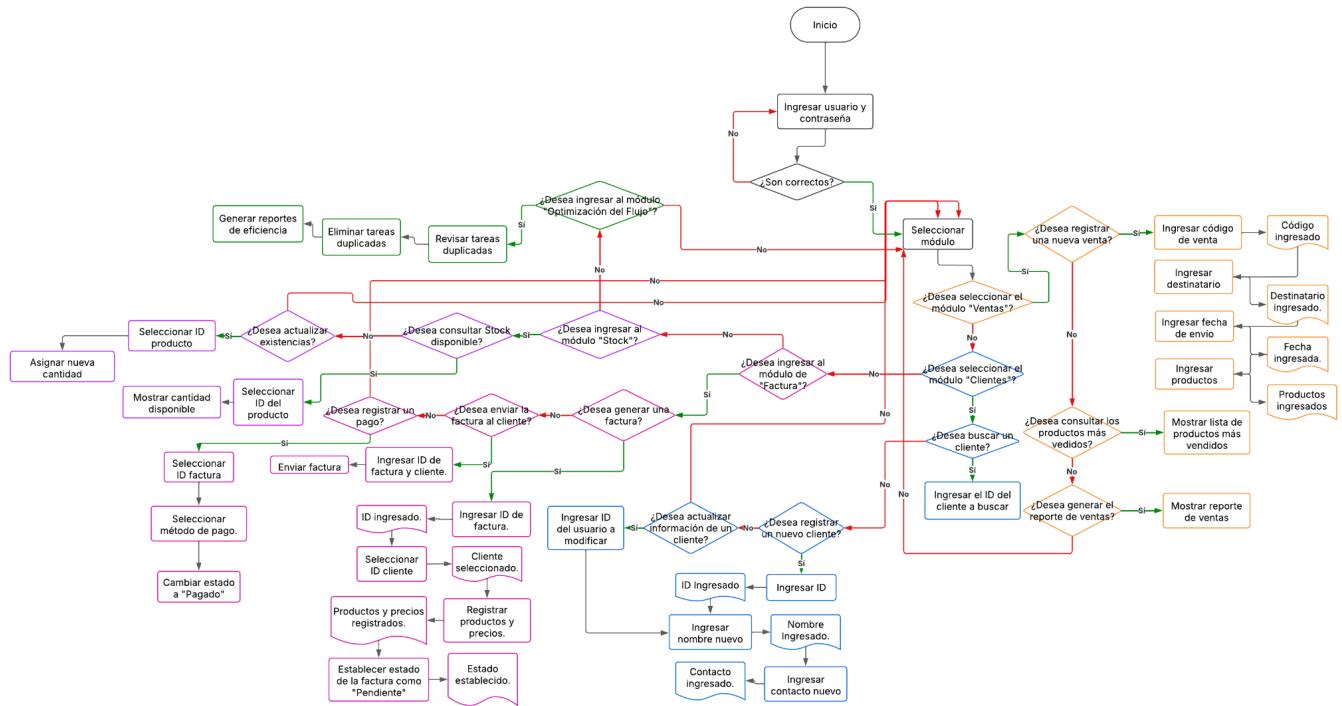


Figura 1: Diagrama de Flujo - Solución Mariana Álvarez

II. Segunda Solución: Propuesta Daniel Moreno

Propone, teniendo en cuenta la información proporcionada por el cliente, en crear un sistema el cual mejore la veracidad y autenticidad del seguimiento a las ventas realizadas por el cliente, y que optimice procesos de búsqueda referentes a cualquier apartado de una venta.

Dentro del sistema, a cada persona que acceda, según su usuario y contraseña, se le determinará un rol entre administrador y vendedor. Cada rol tiene funciones específicas y capacidades únicas. El usuario, tendrá acceso a los siguientes módulos:

- Factura: Módulo encargado de almacenar y generar las facturas de las ventas realizadas
- Producto: Almacena información respecto a los productos, como idProducto, descripciónProducto, etc.
- Inventario: Se relaciona con Producto, pero determina cual es la cantidad de producto disponibles, cuáles son, etc.

Esta solución se implementa mediante BD SQL y en caso dado que sea posible, se implementaría con herramientas noSQL. A continuación, se presenta un diagrama de flujo que representa la solución.

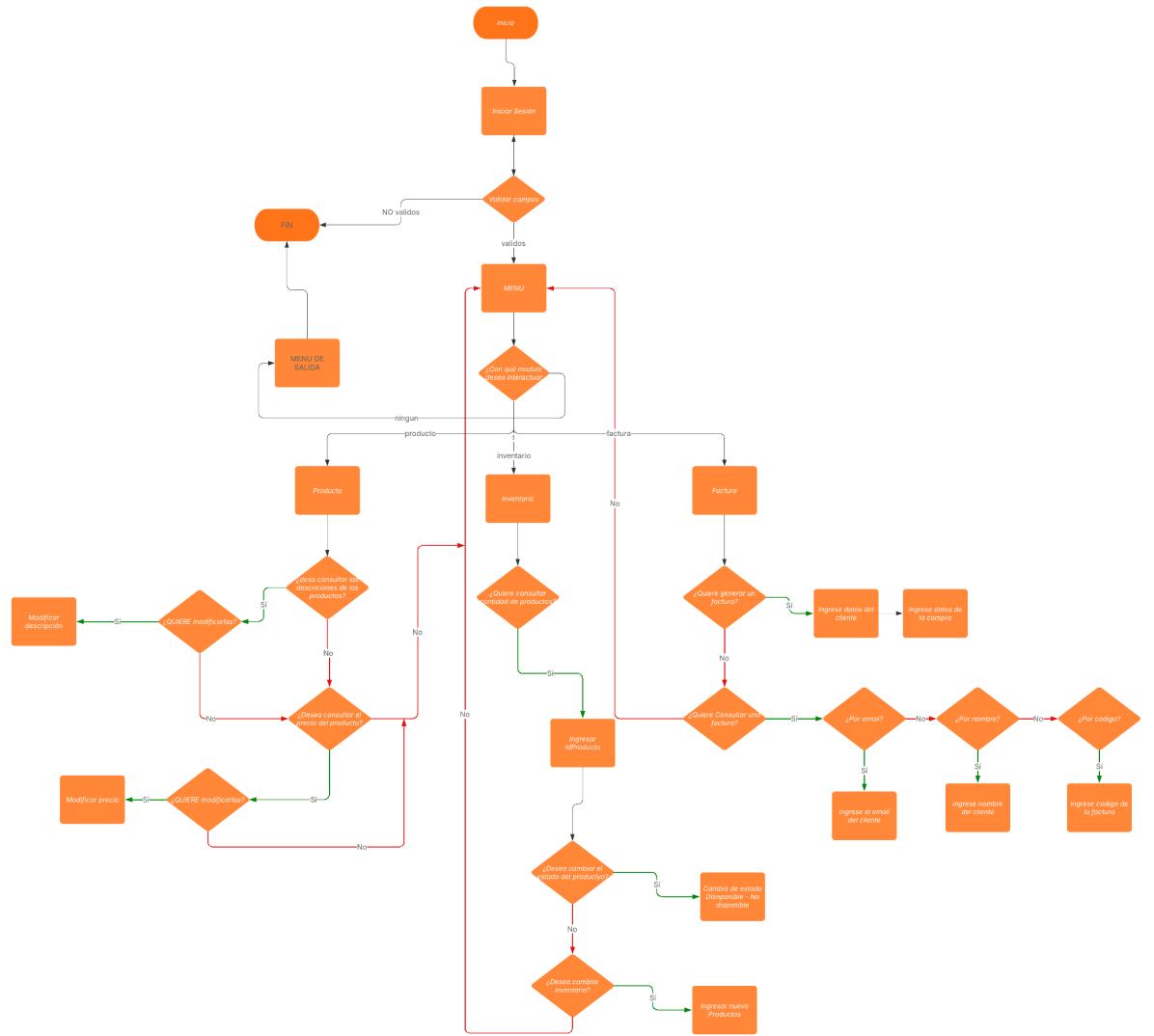


Figura 2: Diagrama de Flujo - Solución Daniel Moreno

III. Tercera Solución: Propuesta Lina Peña

Propone diseñar un sistema de gestión enfocado en optimizar tanto la administración de la información como el proceso de facturación. En cuanto a la creación de esta interfaz, la propuesta es organizada de la siguiente forma:

Módulos:

- Usuario: Controla el acceso al sistema (empleados, gerentes, administradores) con permisos definidos.
- Cliente: Gestiona la información de clientes, como nombre, dirección, contacto.
- Venta: Se encarga del control de las ventas realizadas con detalles como productos vendidos, precio, cliente, fecha de venta, etc.
- Producto: Permite el acceso a información sobre los productos, incluyendo categoría, precio, descripción, etc.
- FacturaciónVenta: Este módulo genera facturas electrónicas asociadas a las ventas realizadas.
- Inventario: Gestiona el stock y lleva entradas y salidas de productos.

Así la base de datos relacional se tendrá para almacenar información crítica y estructurada como clientes, ventas, inventarios, productos, facturas y usuarios. Además, se podrá implementar con sistemas como MySQL, el cual es gratuito y por ende no produce costo.

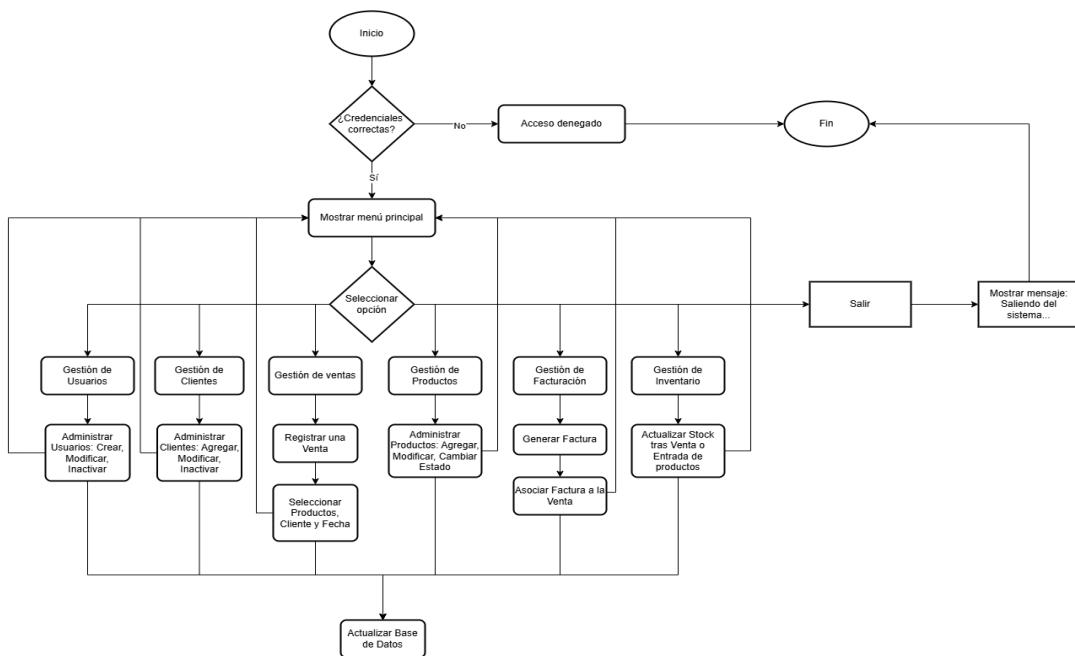


Figura 3: Diagrama de Flujo - Solución Lina Peña

IV. Cuarta Solución: Propuesta Sergio Velandia

Propone desarrollar un sistema de facturación y gestión enfocado a la simplicidad y la sencillez de uso, para que su aplicación sea efectiva, rápida e intuitiva. Se usará base de datos no relacional, por lo tanto usaremos NoSQL o MongoDB, el cual nos ayudará a darle una mayor flexibilidad y escalabilidad a los datos manejados masivamente de la empresa:

- FrontEnd y BackEnd.

- Base de datos NoSQL.
- Almacenamiento de Facturas.
- Autenticación y seguridad.

El sistema se divide en 5 módulos principales, cada uno con su propia colección en la base de datos NoSQL.

- Módulo Usuario: Este módulo gestiona los accesos al sistema y define los permisos según el tipo de usuario.
- Módulo Cliente: Este módulo almacena la información de cada cliente para facilitar la búsqueda y vinculación con facturas.
- Módulo Stock: Permite gestionar el stock de productos y registrar entradas/salidas de inventario en tiempo real.
- Módulo Factura: Aquí se genera la factura electrónica y se almacena de forma segura.
- Módulo Reporte: Este módulo permite generar estadísticas y reportes para la toma de decisiones.

De esta manera, la base de datos será efectiva y podrá manejar datos mucho más grandes, lo cual hace que su flujo sea más efectivo, en el sistema de los clientes registrados y el stock disponible en el momento de hacer una compra/venta de algún producto.

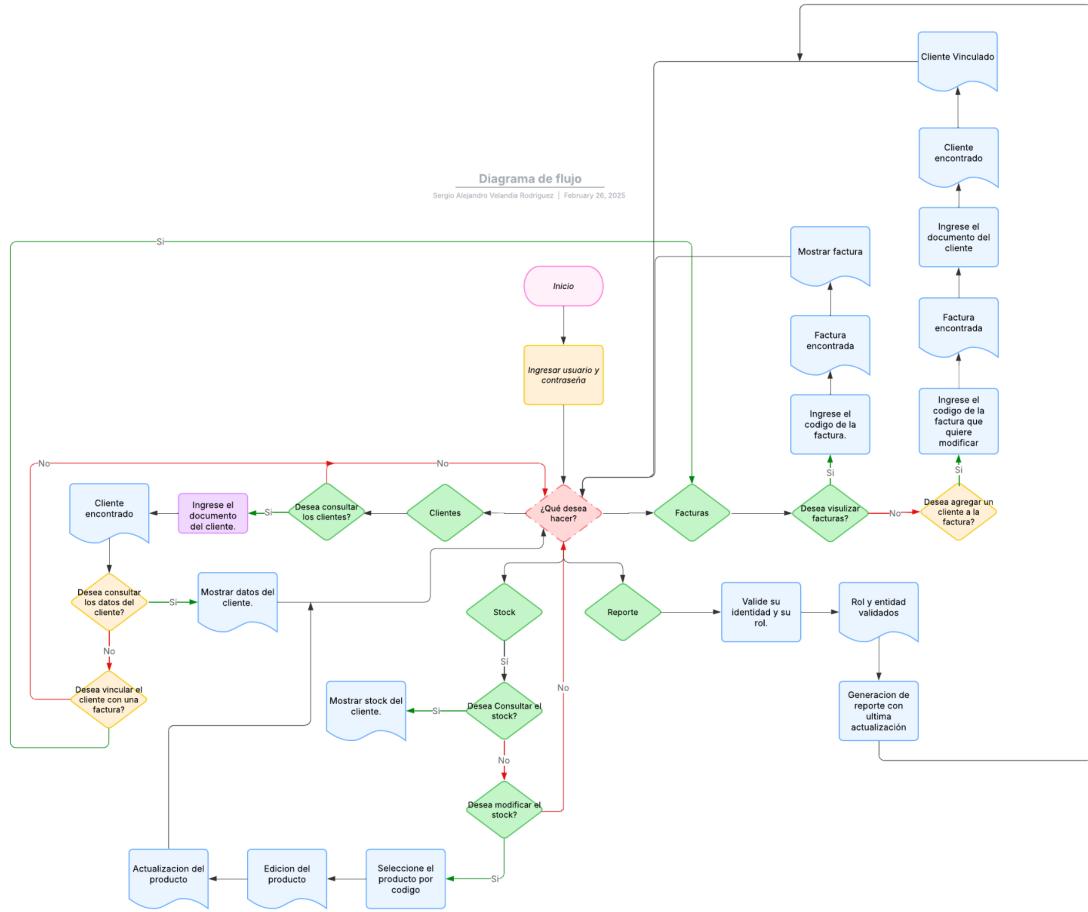


Figura 4: Diagrama de Flujo - Solución Sergio Velandia

Alternativa de solución seleccionada

Se ha decidido hacer una mezcla de las 4 propuestas, eliminando algunos módulos y combinando micro-procesos de cada uno de los diagramas de flujo. En el Diagrama de Clases se puede evidenciar esta estructura.

Para esto, los módulos que se seleccionaron, con su respectiva funcionalidad se describen a continuación:

1. Usuario: Que permitirá asignar un tipo según la persona que desee entrar al sistema. Además de contener los datos personales y una clave de cada uno de los registros.

2. Cliente: Garantiza que se tenga recopilada la información de todos los clientes de la empresa con sus datos personales.
3. Venta: Gestionará los pedidos de las ventas que se hagan teniendo en cuenta todos los factores que pueden variar, tanto la cantidad de productos como sus totales.
4. FacturaVenta: Tendrá toda la información de las facturas que se relacionan con las ventas, incluyendo el destinatario y el proveedor.
5. Producto: Tendrá las especificaciones de cada uno de los artículos que ofrece la empresa, teniendo en cuenta especialmente su categoría (Relojes o Joyería)
6. Stock: Gestionará la cantidad de productos disponibles dentro del inventario para su respectiva comercialización.

la solución se representa en el siguiente diagrama de flujo:

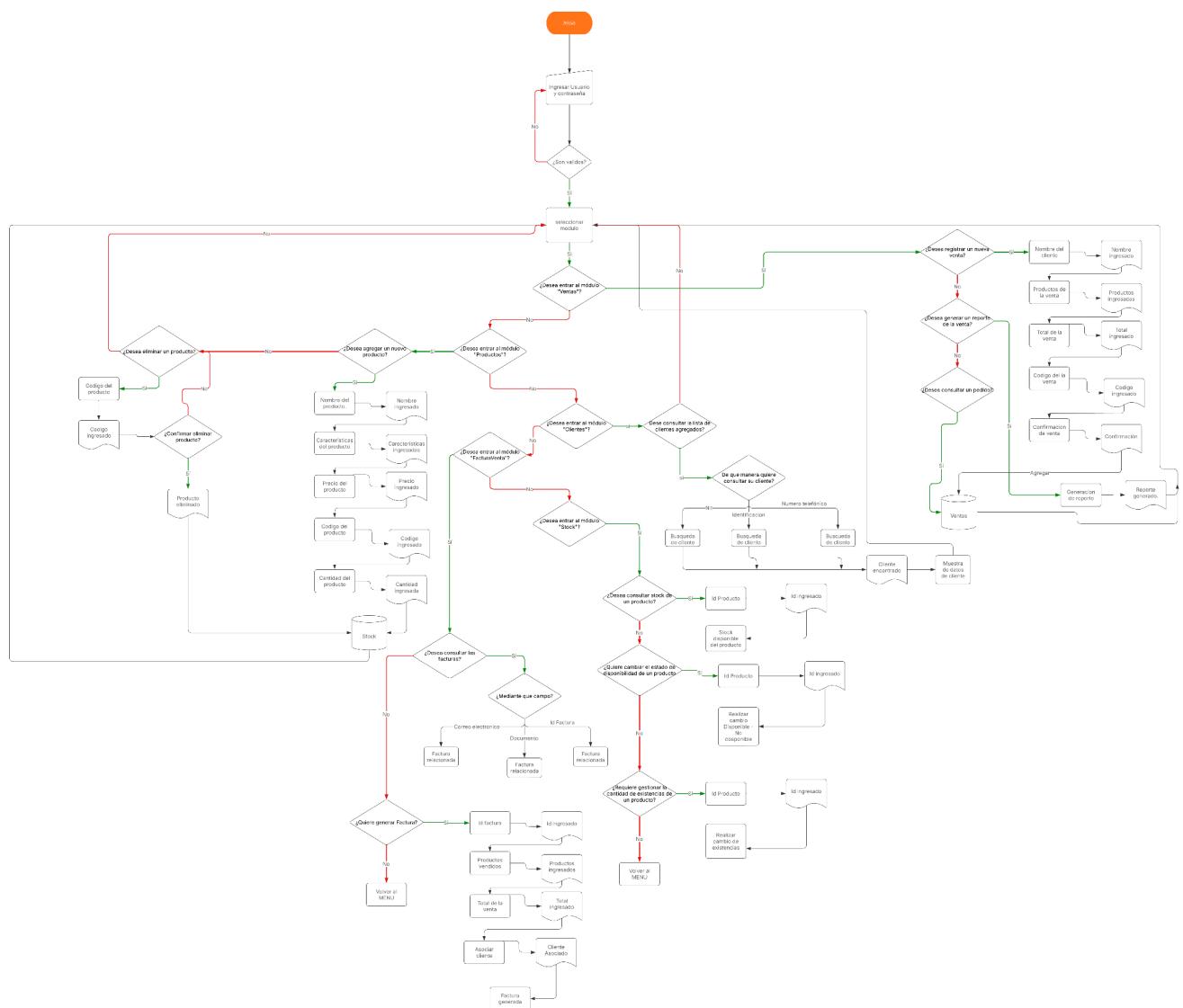


Figura 5: Diagrama de Flujo - Solución final

En caso de no poder apreciar correctamente el diagrama, el siguiente link lo dirigirá a la página donde se realizó el diagrama:

https://lucid.app/lucidchart/e208fade-b202-4fcd-a992-738650b56292/edit?invitationId=inv_ef7479d-67f5-4700-b2da-3f46844cc82a

Alcance

El sistema de facturación para la empresa “Palacio Del Tiempo” es un proyecto ambicioso, el cual abarca la digitalización y automatización del proceso de facturación, gestión de clientes, control de inventario y análisis de ventas. Los problemas que actualmente presenta el Palacio del Tiempo se verán solucionados con este sistema, usaremos métodos tales como: Facturación Electrónica, Gestión de Clientes, Inventario de Productos, Reportes y Análisis, Gestión de Usuarios y Permisos. Con la ayuda de un lenguaje de consulta estructurada, como lo es SQL, y con su correspondiente gestor, en este caso MySQL Workbench.

El proyecto propone un desarrollo a lo largo de los siguientes 2 meses, con plazo máximo para su culminación siendo el 29 de mayo, con el fin de dar una solución grande a problemas, tales como; Reducción de errores humanos, optimización del tiempo y procesos internos, mejora en la gestión de clientes y pagos, mayor control y seguridad en el inventario, toma de decisiones basada en datos. Todo esto para que algunas tareas hechas por mano humana dejen de ser repetitivas y en algunos casos, torpes.

Este proyecto, además de solucionar problemas, también busca inspirar a futuros proyectos e investigaciones, ser una referencia global para la modernización de sistemas de gestión y de administración en todos los sectores posibles. Queremos que sea bien valorado por su eficiencia, además de que otros negocios y empresas puedan usarlo y hasta usarlos en su favor, los cuales pueden mejorar este sin duda, es una evolución digital en el ámbito empresarial.

Metodología

La metodología que se está implementando para el desarrollo del presente proyecto, es la Metodología Ágil Kanban; que permite mediante un tablero colaborativo, diseñar las estrategias, el cronograma, subir tareas, cargar avances, e incluso, rectificar ciertas partes del desarrollo. Esto garantiza que el trabajo en equipo se realice de la manera más eficiente posible y que la organización sea mucho más ágil.

Además, es una alternativa interactiva en la que todos podemos ir editando todo lo necesario, y en este caso, usamos la herramienta de Trello para la creación del tablero. A continuación, se muestran algunas imágenes del formato que se está implementando y adjuntamos el link de visualización:

<https://trello.com/invite/b/67a127ea431f6a8cee73c7dd/ATTI56cf19056247ed5f80cc82dfabc470eeAC8CA021/proyecto-ingenieria>

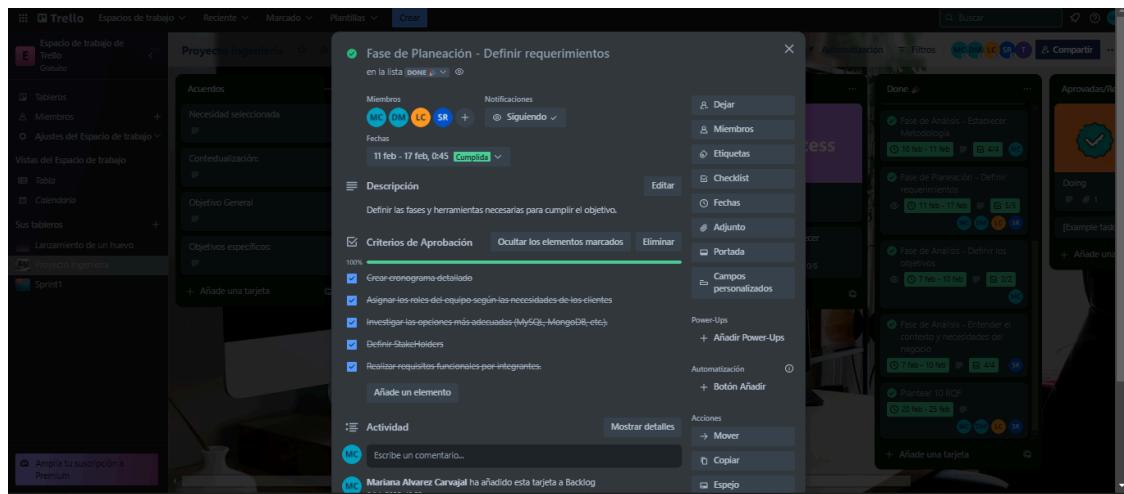


Figura 6: evidencia metodología Kanban

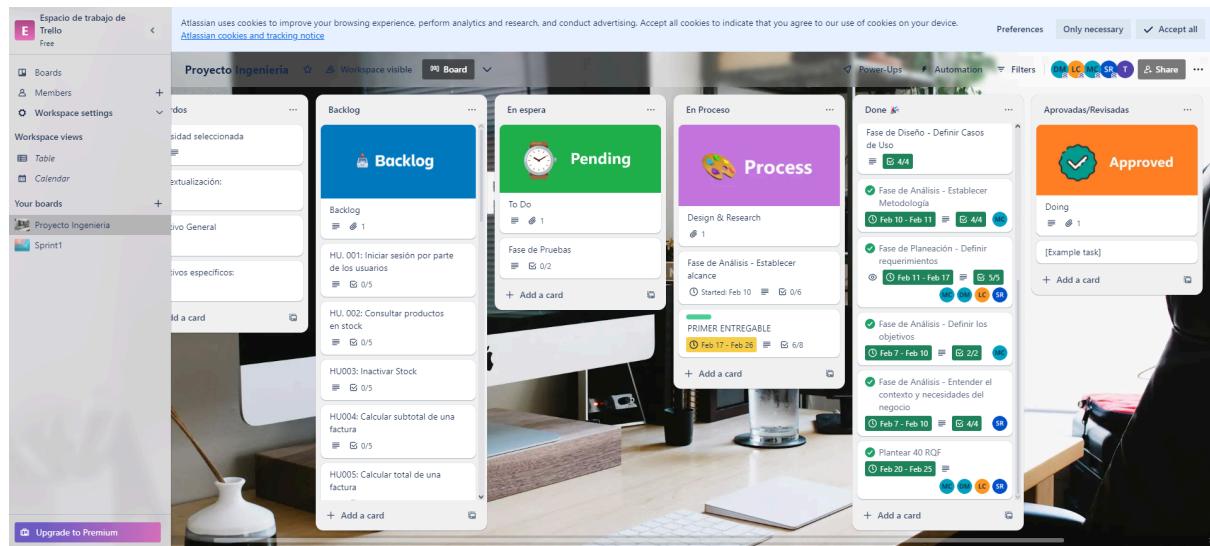


Figura 7: evidencia metodología Kanban

Levantamiento de información y RQFs

Se realizó mediante un método interactivo, en donde discutimos las necesidades del cliente gracias a un WorkShop y nos brindaron la información de la empresa dentro de un excel, al cual, cada uno de los integrantes le realizó una ETL para tener el archivo de mejor calidad. La ETL final se realizó en EXCEL

Requerimientos funcionales

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF001 | <p>Nombre: Iniciar sesión</p> <p>Descripción: El sistema permitirá que los empleados del sistema accedan a su cuenta con un usuario y contraseña registrados. La autenticación debe validar las credenciales ingresadas y permitir el acceso solo a usuarios autorizados. En caso de tres intentos fallidos, se debe bloquear temporalmente la cuenta.</p> <p>Usuarios: Vendedores y administrador</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF002 | <p>Nombre: Consultar productos en stock</p> <p>Descripción: El sistema permite acceder a un listado detallado de los productos disponibles en inventario, incluyendo su código, nombre, descripción, cantidad disponible y precio de venta.</p> <p>Usuarios: Vendedores y administrador</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF003 | <p>Nombre: Inactivar Stock</p> <p>Descripción: El sistema permitirá inactivar el Stock mediante el código del mismo, en caso de que ya no haya demanda suficiente, o deterioros, o porque sencillamente se va a descontinuar un producto.</p> |

| | |
|--|--------------------------------|
| | Usuarios: Administrador |
|--|--------------------------------|

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF004 | <p>Nombre: Calcular Subtotal de una Factura</p> <p>Descripción: El sistema permitirá calcular el total de una factura de venta mediante la cantidad de productos y los precios unitarios de cada uno.</p> <p>Usuarios: Administrador y Vendedores</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF005 | <p>Nombre: Calcular total de una Factura</p> <p>Descripción: El sistema permitirá calcular el total de una factura de venta mediante el SubTotal y se cargará con la sumatoria de los impuestos totales.</p> <p>Usuarios: Administrador y Vendedores</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF006 | <p>Nombre: Generar Factura</p> <p>Descripción: El sistema permitirá crear una factura teniendo en cuenta diferentes atributos como: código, productos, destinatarios, vendedor, total, entre otras.</p> |

| | |
|--|---|
| | Usuarios: Administrador y Vendedores |
|--|---|

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF007 | <p>Nombre: Inactivar Factura</p> <p>Descripción: El sistema permitirá inactivar el estado de Factura en caso de que una venta sea cancelada. Esto quiere decir que la factura nunca se cobró y también dejará de estar pendiente.</p> <p>Usuarios: Administrador y Vendedores</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF008 | <p>Nombre: Modificar la información de la Venta</p> <p>Descripción: El sistema permitirá modificar los datos de una venta existente mediante el código de la misma. Una vez se accede a la venta, se registran los nuevos datos.</p> <p>Usuarios: Administrador y Vendedores</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF009 | <p>Nombre: Consultar productos enviados en una venta</p> <p>Descripción: El sistema permitirá consultar los productos que se enviaron según el código de venta registrado. Se visualizarán tanto el nombre de los productos, como su respectiva cantidad.</p> <p>Usuarios: Administrador y Vendedores</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF010 | <p>Nombre: Modificar la información de un producto</p> <p>Descripción: El sistema permitirá modificar los datos de un producto existente mediante el código del mismo. Una vez se accede al producto, se registran los nuevos datos.</p> <p>Usuarios: Administrador</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF011 | <p>Nombre: Consultar clientes por documento</p> <p>Descripción: El sistema permitirá al usuario filtrar según el documento de cada persona o empresa (Número de Identificación Tributaria) la cual haya sido cliente del Palacio del Tiempo.</p> <p>Usuarios: Vendedor, administrador</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF012 | <p>Nombre: Cambiar clave</p> <p>Descripción: Se permitirá a un selecto grupo de personas tener la capacidad de cambiar la contraseña de aquellas personas que lo necesiten</p> <p>Usuarios: administrador</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF013 | <p>Nombre: Consultar envío por fecha de reparto</p> <p>Descripción: El sistema, dentro de sus diferentes campos, tiene el apartado de Venta (de un producto), este campo se podrá filtrar por fecha de reparto de cada envío.</p> <p>Usuarios: administrador, vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF014 | <p>Nombre: Consultar factura por fecha</p> <p>Descripción: El sistema permitirá al vendedor realizar la consulta de cada factura según su fecha de expedición.</p> <p>Usuarios: vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF015 | <p>Nombre: Consultar clientes por ciudad</p> <p>Descripción: El sistema almacena la ubicación de cada compra realizada, guardando ciudad y localidad. Por tanto, se permitirá consultar las compras según la ciudad donde fueron realizadas.</p> <p>Usuarios: vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF016 | <p>Nombre: Consultar Venta por fecha de entrega</p> <p>Descripción: El sistema almacena, al igual que la fecha de reparto, la fecha de entrega de una Venta. Por tanto, se permitirá consultar las ventas mediante su fecha de entrega.</p> <p>Usuarios: vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF017 | <p>Nombre: Consultar factura por email</p> <p>Descripción: Cada factura tiene un email que le corresponde a cada comprador, el sistema permitirá filtrar las facturas, según el email de cada quien.</p> <p>Usuarios: vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF018 | <p>Nombre: Generar Tablero de datos</p> <p>Descripción: El programa permitirá generar al menos una forma de visualizar la información, ya sea mediante power BI u otra herramienta.</p> <p>Usuarios: Vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF019 | <p>Nombre: Modificar la disponibilidad de producto</p> <p>Descripción: cada producto tiene un estado, ya sea disponible o no. El programa permitirá el administrador y vendedor modificar estado cuando se requiera</p> <p>Usuarios: Administrador, vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF020 | <p>Nombre: Registrar clientes</p> <p>Descripción: Cuando se requiera, el sistema permitirá el vendedor crear un nuevo campo en la lista clientes</p> <p>Usuarios: Vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF021 | <p>Nombre: Editar información clientes</p> <p>Descripción: Cuando se requiera, el sistema debe permitir a los usuarios modificar la información habida de cada cliente registrado con anterioridad según corresponda.</p> |

| | |
|--|--|
| | Usuarios: Administrador, vendedor |
|--|--|

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF022 | <p>Nombre: Inactivar clientes</p> <p>Descripción: Cuando se requiera, el sistema debe permitirle al usuario modificar el estado del cliente de activo a inactivo y viceversa según la situación.</p> <p>Usuarios: Administrador, vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF023 | <p>Nombre: Modificar estado de pago Factura</p> <p>Descripción: El sistema debe permitir al usuario modificar el estado de pago de las facturas generadas de pendiente a cobrado una vez que estas sean canceladas por el cliente.</p> <p>Usuarios: Administrador, vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---------------------------------------|
| RQF024 | Nombre: Asignar rol al usuario |

| | |
|--|--|
| | <p>Descripción: El sistema debe permitir la asignación de roles según las acciones que pueda o no realizar el usuario en este.</p> <p>Usuarios: Administrador.</p> |
|--|--|

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF025 | <p>Nombre: Consultar usuario por documento</p> <p>Descripción: El sistema debe permitir al usuario la búsqueda y consulta de datos de clientes dado el documento de identificación de este.</p> <p>Usuarios: Administrador, vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF026 | <p>Nombre: Notificar desabastecimientos</p> <p>Descripción: El sistema debe generar notificaciones de fácil visibilidad para al usuario que alerte el bajo stock de un producto dado su umbral mínimo.</p> <p>Usuarios: Administrador, vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF027 | <p>Nombre: Consultar factura por estado</p> <p>Descripción: El sistema debe permitir al usuario la consulta y búsqueda de una factura dado su estado, ya sea pendiente o cobrada.</p> <p>Usuarios: Administrador, vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF028 | <p>Nombre: Editar información usuario</p> <p>Descripción: El sistema debe permitir al usuario la modificación o actualización de datos de cada usuario según se requiera.</p> <p>Usuarios: Administrador.</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF029 | <p>Nombre: Inactivar usuario</p> <p>Descripción: El sistema debe permitir al usuario con los permisos necesarios, la inactivación de otro usuario según su papel con la empresa.</p> <p>Usuarios: Administrador.</p> |

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

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF030 | <p>Nombre: Notificar vencimiento de factura</p> <p>Descripción: El sistema debe mostrar una alerta al usuario cuando una factura en estado pendiente esté próxima a vencer.</p> <p>Usuarios: Administrador y Vendedor</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF031 | <p>Nombre: Registrar productos en inventario.</p> <p>Descripción: El sistema debe permitir a los administradores y vendedores registrar nuevos productos en el inventario, especificando su nombre, categoría, precio, proveedor y cantidad inicial en stock.</p> <p>Usuarios: Administrador, vendedor.</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| | <p>Nombre: Generar reportes de ventas diarias.</p> |

| | |
|---------------|---|
| RQF032 | <p>Descripción: El sistema deberá generar automáticamente un reporte diario de ventas con información sobre productos vendidos, cantidad, total de ingresos y vendedor responsable. Este reporte debe poder exportarse en formatos Excel.</p> <p>Usuarios: Administradores, Propietarios.</p> |
|---------------|---|

| CÓDIGO | Requisitos Funcionales |
|---------------|--|
| RQF033 | <p>Nombre: registrar de entrada del producto.</p> <p>Descripción: El sistema debe registrar cuando entra un producto al stock, que ha sido distribuido por entidades externas a la empresa, siendo fechas y referencia del producto, con todas sus características.</p> <p>Usuarios: Administrador, vendedor, propietarios.</p> |

| CÓDIGO | Requisitos Funcionales |
|---------------|--|
| RQF034 | <p>Nombre: Generar un historial de modificaciones en facturas.</p> <p>Descripción: El sistema debe almacenar un historial de cambios realizados en cada factura, registrando qué usuario realizó la modificación, qué cambios hizo y la fecha en que se realizaron.</p> <p>Usuarios: Administrador, propietarios.</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| | <p>Nombre: Cargar productos desde archivo CSV.</p> |

| | |
|--------|--|
| RQF035 | <p>Descripción: El sistema permitirá la carga masiva de productos al inventario mediante la importación de archivos en formato CSV, reduciendo el tiempo de registro manual.</p> <p>Usuarios: Administrador.</p> |
|--------|--|

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF036 | <p>Nombre: Buscar productos por características.</p> <p>Descripción: El sistema permitirá a los usuarios buscar productos utilizando palabras como “moderno”, “deportivo”, “clásico”, etc.</p> <p>Usuarios: Administrador, vendedor.</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF037 | <p>Nombre: Agregar usuario nuevo.</p> <p>Descripción: En el sistema, los administradores y los propietarios podrán agregar nuevos usuarios, para poder asignarles un tipo, documento, estado y demás, este es el perfil general que tendrán todas las personas y entidades en la empresa.</p> <p>Usuarios: Administrador, propietarios.</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF038 | <p>Nombre: Consultar información de pagos.</p> <p>Descripción: El sistema debe mostrar toda la información de un pago: hora, total monetario, método de pago, por o sin cuotas, además debe</p> |

| | |
|--|--|
| | decir si el cliente aún debe o ya pagó todo el resultante de la factura, es decir, si ya se efectuó el pago completo o tiene saldo disponible. |
| | Usuarios: Administrador. |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF039 | <p>Nombre: Actualizar el stock</p> <p>Descripción: Cuando un vendedor realice una venta, el sistema debe actualizar automáticamente el stock del producto en la base de datos y reflejar el cambio en todos los dispositivos conectados, este cambiará automáticamente todos los estados de un producto si este fue vendido.</p> <p>Usuarios: Administrador, vendedor, propietarios.</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|--|
| RQF040 | <p>Nombre: Filtrar los productos más costosos.</p> <p>Descripción: El sistema filtrará en el apartado de productos los más costosos, los cuales aparecen en orden descendiente.</p> <p>Usuarios: Administrador, propietarios, vendedores.</p> |

| CÓDIGO | Requisitos Funcionales |
|--------|---|
| RQF041 | <p>Nombre: insertar productos</p> <p>Descripción: el sistema permitirá insertar productos, en su respectiva tabla, mediante un procedimiento.</p> |

| | |
|--|---|
| | Usuarios: Administrador, propietarios, vendedores. |
|--|---|

Requisitos No funcionales:

| CÓDIGO | Requisitos No Funcionales |
|---------|--|
| RQNF001 | <p>Nombre: Diseño de Interfaz de Usuario</p> <p>Descripción: El sistema debe contar con una interfaz intuitiva y fácil de usar, permitiendo que los usuarios puedan navegar sin dificultades. Debe seguir principios de diseño UX/UI, con menús accesibles, colores contrastantes y elementos visuales claros. Acordes con la imagen corporativa de la empresa.</p> <p>Usuarios: Clientes, Vendedores, Administrador.</p> |

| CÓDIGO | Requisitos No Funcionales |
|---------|---|
| RQNF002 | <p>Nombre: Garantizar disponibilidad para los usuarios</p> <p>Descripción: El sistema debe garantizar una disponibilidad, asegurando que los usuarios puedan acceder a la información sin interrupciones. Esto implica la implementación de servidores redundantes y planes de recuperación ante fallos.</p> <p>Usuarios: Clientes, Vendedores, Administrador.</p> |

| CÓDIGO | Requisitos No Funcionales |
|--------|---------------------------|
|--------|---------------------------|

| | |
|----------------|---|
| RQNF003 | <p>Nombre: Implementar sistema de seguridad en la plataforma</p> <p>Descripción: Se deben implementar mecanismos de autenticación y autorización para proteger la información de toda la empresa.</p> <p>Esto incluye encriptación de datos sensibles y control de acceso basado en roles.</p> <p>Usuarios: Clientes, Vendedores, Administrador.</p> |
|----------------|---|

| CÓDIGO | Requisitos No Funcionales |
|----------------|--|
| RQNF004 | <p>Nombre: Optimizar el tiempo de respuesta dentro del sistema</p> <p>Descripción: Asegurar que las consultas y funcionalidades de las bases de datos, tanto relacional como no relacional, se ejecuten en máximo 2 segundos. Esto, con el objetivo de garantizar una experiencia fluida para el usuario.</p> <p>Usuarios: Clientes, Vendedores, Administrador.</p> |

| CÓDIGO | Requisitos No Funcionales |
|----------------|--|
| RQNF005 | <p>Nombre: Especificaciones Mínimas de Hardware y Software</p> <p>Descripción:</p> <p>Tener acceso a uno o más computadores</p> <p>Mínimo 4 GB RAM (para garantizar un mejor rendimiento en la gestión de bases de datos y aplicaciones web)</p> <p>Sistema Operativo Windows 10 o superior</p> <p>Disco duro de 500 GB HDD o 128 GB SSD (preferiblemente SSD para mayor velocidad en la ejecución de consultas)</p> |

| | |
|--|--|
| | <p>Conexión a Internet estable</p> <p>Que tenga las herramientas y programas necesarios para llevar a cabo el aplicativo, base de datos (MYSQL) y servidor (APACHE).</p> |
| | <p>Usuarios: Clientes, Vendedores, Administrador.</p> |

Histórias de Usuario

Ya con los RQF definidos es pertinente realizar las historias de usuario, donde medio de trabajo equivalen a un (1) punto de esfuerzo:

| CÓDIGO | HISTORIA DE USUARIO |
|--------|---|
| HU001 | <p>Nombre: Iniciar Sesión</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero iniciar sesión para poder ingresar a la plataforma que contiene toda la información de la empresa.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Validar las credenciales ingresadas - Permitir modificar la contraseña - Restringir solo para máximo tres intentos fallidos. - Bloquear la cuenta si se cumple la condición de los tres intentos fallidos - Permitir recuperar la cuenta en caso de olvidar algún dato. <p>Puntos de Esfuerzo: 6</p> |

| | |
|--------------|--|
| HU002 | <p>Nombre: Consultar productos en stock</p> <p>Descripción: Yo, como Vendedor o Administrador, quiero consultar productos en stock para garantizar que se ofrezcan aquellos que todavía tienen existencias y restaurar aquellos que se encuentran en escasez.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Mostrar un listado con todos los productos en stock - Incluir todos los atributos del producto. - Garantizar una barra para filtrar productos por nombre o código. - Permitir ordenar los productos por cantidad, precio o nombre. - Cumplir con las notificaciones de desabastecimiento de stock <p>Puntos de Esfuerzo: 2</p> |
| HU003 | <p>Nombre: Inactivar Stock</p> <p>Descripción: Yo, como administrador, quiero inactivar productos en stock, para evitar que se sigan vendiendo cuando no hay demanda, están deteriorados o van a ser descontinuados.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Permitir buscar el stock a inactivar por código. - Dejar de mostrar el stock cuando se inactiva. - Solicitar una confirmación antes de inactivar un stock. - Registrar fecha y motivo de la inactivación (Según las categorías nombradas antes) - Verificar que el que haga estas acciones sea el administrador. <p>Puntos de Esfuerzo: 3</p> |

| | |
|--------------|---|
| HU004 | <p>Nombre: Calcular subtotal de una Factura</p> <p>Descripción: Yo, como vendedor o administrador, quiero que el sistema calcule el subtotal de una factura, para conocer la base de la venta antes de aplicar el total con impuestos.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Obtener la cantidad de productos - Conocer el precio unitario de cada uno de ellos - Obtener el subtotal con estos valores - Actualizar el subtotal al agregar o quitar productos - Indicar los descuentos de los productos en caso de que existan. <p>Puntos de Esfuerzo: 3</p> |
| HU005 | <p>Nombre: Calcular total de una factura</p> <p>Descripción: Yo, como vendedor o administrador, quiero que se calcule el total de una factura, para conocer el valor final (incluyendo impuestos y cantidad de productos) de una venta.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Sumar el subtotal con los impuestos - Reflejar automáticamente los cambios en el total si se modifica la cantidad de productos. - Cumplir con RQF004 - Visualizar total y subtotal en la factura - Negar números negativos en el total <p>Puntos de Esfuerzo: 3</p> |
| HU006 | <p>Nombre: Generar Factura</p> <p>Descripción: Yo, como vendedor o administrador, quiero generar una factura con los datos de la venta, para documentar las transacciones y los movimientos de las ventas de forma correcta.</p> |

| | |
|-------|---|
| | <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Incluir código, productos, destinatario, vendedor y total - Permitir agregar más de un producto en la factura. - Cumplir con el RQF004 - Cumplir con el RQF005 - Permitir descargar la factura en un formato imprimible <p>Puntos de Esfuerzo: 6</p> |
| HU007 | <p>Nombre: Inactivar factura</p> <p>Descripción: Yo, como vendedor o administrador, quiero inactivar una factura cuando una venta es cancelada, para que deje de tener un estado en la que se diga si está pendiente, y que así, no se cobre.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Solicitar el código de la factura a inactivar. - Cambiar el estado de la factura a “inactivo” - Solicitar una confirmación antes de inactivarla para estar seguros del proceso. - Registrar motivo y fecha de inactivación. - Verificar que tanto administradores como vendedores puedan inactivar <p>Puntos de Esfuerzo: 2</p> |
| HU008 | <p>Nombre: Modificar la información de una venta.</p> <p>Descripción: Yo, como vendedor o administrador, quiero modificar la información de una venta existente para corregir errores o actualizar los datos en caso de que sea necesario.</p> <p>Criterios de Aceptación:</p> |

| | |
|-------|--|
| | <ul style="list-style-type: none"> - Solicitar el código de la venta a modificar - Verificar que la venta a modificar no esté inactiva - Permitir modificar los productos, cantidades y datos del cliente en la venta. - Notificar cuando una venta sea modificada - Recalcular el total en caso de cambiar los productos. <p>Puntos de Esfuerzo: 2</p> |
| HU009 | <p>Nombre: Consultar productos enviados en una venta.</p> <p>Descripción: Yo, como vendedor o administrador, quiero consultar los productos enviados en una venta para conocer los movimientos y los detalles de los artículos entregados.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Buscar una venta por su código - Mostrar el nombre de los productos y la cantidad enviada - Mostrar un mensaje en caso de que la venta no se haya enviado. - Verificar que la venta no haya sido inactivada <p>Puntos de Esfuerzo: 6</p> |
| HU010 | <p>Nombre: Modificar la información de un producto.</p> <p>Descripción: Yo, como administrador, quiero modificar los datos de un producto, para actualizar su información en caso de que sea necesario.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Solicitar el producto a partir de su código - Verificar que solo los administradores |

| | |
|--------------|--|
| | <p>puedan cumplir esta función.</p> <ul style="list-style-type: none"> - Mostrar un mensaje en caso de seleccionar un producto inactivo. - Permitir modificar los atributos del producto. - Actualizar los datos nuevos del producto en los otros módulos. <p>Puntos de Esfuerzo: 2</p> |
| HU011 | <p>Nombre: Consultar clientes por NIT</p> <p>Descripción: Yo como Vendedor, necesito poder consultar mis clientes por el NIT el cual registraron al momento de la compra, con el fin de facilitar, y dar opciones, en mi búsqueda por cliente.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Relacionar cada cliente con un nit - Tener un interfaz para hacer la consulta - El resultado debe visualizarse en una interfaz clara y ordenada. - se deben devolver todos los datos correspondientes al usuario - Garantizar que la información sea coherente <p>Puntos de Esfuerzo: 1</p> |
| HU012 | <p>Nombre: Cambiar clave</p> <p>Descripción: Yo como administrador, requiero poder cambiar la contraseña de usuarios que lo requieran, con el fin de garantizar el acceso a su cuenta en caso de olvido o problemas de seguridad.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Verificar el rol de Administrador al momento de hacer el cambio - Notificar al usuario sobre su cambio - Implementar método para contacto |

| | |
|-------|---|
| | <p>con un administrador en caso de requerir el cambio de contraseña</p> <ul style="list-style-type: none"> - Generar historial de cambio de claves - Asegurar que la nueva contraseña siga parámetros de seguridad <p>Puntos de Esfuerzo: 2</p> |
| HU013 | <p>Nombre: envio por fecha de reparto</p> <p>Descripción: Como vendedor quiero filtrar los producto vendidos, según su fecha de reparto, en caso de que haya sido enviado, con el fin de tener una mejor gestión de los pedidos realizados</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - permitir ingreso de fechas al sistema para el filtrado - Desplegar información sobre qué producto se vendió y a quien - Únicamente deben de haber fechas coherentes - Buscar por meses - Buscar por año <p>Puntos de Esfuerzo: 3</p> |
| HU014 | <p>Nombre: factura por fecha</p> <p>Descripción: Como vendedor necesito consultar las facturas realizadas según su fecha de expedición puesto que, requiero tener un buen control del sistema de facturación.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Permitir la descarga de la factura - Mostrar detalles sobre el cliente - permitir ingreso de fechas para el filtrado |

| | |
|--------------|--|
| | <ul style="list-style-type: none"> - garantizar la seguridad de los datos personales del cliente. <p>Puntos de Esfuerzo: 3</p> |
| HU015 | <p>Nombre: clientes por ciudad</p> <p>Descripción: Yo como vendedor, necesito consultar clientes según la ciudad donde realizaron su compra, con el fin de mejorar la logística de envíos.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - filtrar clientes por ciudad a través de un campo - devolver datos relevantes de la venta - Herramientas para realizar el filtrado - Actualización de nuevas ventas - Reducir el tiempo de búsqueda por ciudad <p>Puntos de Esfuerzo: 2</p> |
| HU016 | <p>Nombre: Venta fecha de entrega</p> <p>Descripción: Como vendedor requiero acceder a la información de cuanto se entregó el producto vendido al cliente, con el fin de organizar y asegurar que todas las ventas estén en orden.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - garantizar la actualización en tiempo real de los datos. - devolver datos relevantes de la venta - Herramientas para realizar el filtrado - Actualización de nuevas ventas - permitir ingreso de fechas para el filtrado <p>Puntos de Esfuerzo: 2</p> |

| | |
|--------------|--|
| HU017 | <p>Nombre: factura por email</p> <p>Descripción: Como vendedor necesito poder consultar qué facturas están asociadas a cierto correo electrónico, puesto que requiero optimizar mi tiempo al momento de buscar los comprobantes.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Permitir ingresar un email para filtrar las facturas - Permitir descargar la factura - Garantizar información coherente - Actualización de nuevas ventas - Mostrar mensaje en caso de que no se obtenga ningún resultado <p>Puntos de Esfuerzo: 2</p> |
| HU018 | <p>Nombre: Tablero de datos</p> <p>Descripción: Como vendedor requiero ver la información de manera más gráfica, con el fin de facilitar mi comprensión.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Tener herramientas para generar el tablero. - Validar que la información esté organizada correctamente - Implementar herramientas de filtrado - Actualización de nuevas ventas - Permitir únicamente al vendedor generar el tablero <p>Puntos de Esfuerzo: 10</p> |
| HU019 | <p>Nombre: Disponibilidad de producto</p> <p>Descripción: Como vendedor necesito</p> |

| | |
|--------------|---|
| | <p>modificar la disponibilidad de los productos en el sistema, con el fin de mantener actualizada la información para los clientes</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Bloquear acceso a usuarios que no sean vendedores. - Permitir editar varias disponibilidades al tiempo - Preguntar por una confirmación antes de realizar los cambios - Permitir filtrado según el estado - No Permitir que haya un producto sin estado, o con 2 estados <p>Puntos de Esfuerzo: 2</p> |
| HU020 | <p>Nombre: Registrar clientes</p> <p>Descripción: Como vendedor necesito poder registrar nuevos clientes, con el fin de mantener actualizada la base de datos y poder manejar nuevas compras.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Permitir ingresar los datos del cliente a la BD. - Tener una interfaz para el registro - Establecer información obligatoria - Permitir modificar los datos después del registro del cliente - Preguntar por una confirmación antes de realizar los cambios <p>Puntos de Esfuerzo: 2</p> |
| HU021 | <p>Nombre: Editar información clientes</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero editar la información de los clientes para poder arreglar datos mal registrados o actualizar la información.</p> <p>Criterios de Aceptación:</p> |

| | |
|--------------|---|
| | <ul style="list-style-type: none"> - Cumplir con HU028. - Permitir cambiar el nombre del cliente - Permitir cambiar la dirección del cliente. - Permitir cambiar el o los teléfonos del cliente. - Permitir cambiar el correo del cliente. - Permitir guardar los cambios realizados. - Solicitar confirmación del cambio. - Guardar los cambios hechos. <p>Puntos de Esfuerzo: 2</p> |
| HU022 | <p>Nombre: Inactivar Clientes</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero inactivar clientes específicos e identificar a los clientes constantes de la empresa.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Cumplir con HU028. - Solicitar el documento del cliente a inactivar. - Cumplir HU020. - Cambiar de estado a “inactivo”. - Solicitar una confirmación de inactivación. - Registrar motivo de inactivación. - Verificar que tanto administradores como vendedores puedan inactivar al cliente. <p>Puntos de Esfuerzo: 2</p> |
| HU023 | <p>Nombre: Modificar estado de pago Factura</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero modificar el estado de pago Factura para indicar cuando una factura ya ha sido cobrada.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Habilitar consulta de factura por su código. |

| | |
|-------|--|
| | <ul style="list-style-type: none"> - Permitir la visualización del estado de la factura. - Cambiar el estado a “cobrado”. - Pedir confirmación del cambio de estado. - Guardar los cambios hechos. <p>Puntos de Esfuerzo: 2</p> |
| HU024 | <p>Nombre: Asignar rol al usuario</p> <p>Descripción: Yo, como Administrador, quiero asignar el rol a cada nuevo usuario para identificar qué permisos y acceso darle en la base de datos.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Permitir registro de un nuevo usuario - Permitir la estipulación de rol según los permisos que tendrá el usuario en la BD. - Pedir confirmación de rol. - Guardar información del rol asignado. <p>Puntos de Esfuerzo: 2</p> |
| HU025 | <p>Nombre: Consultar cliente por ID</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero buscar y consultar a un cliente en específico dado su documento para acceder de manera veloz a su información registrada.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Tener filtro de búsqueda por ID - Validar ID proporcionado. - Permitir visualización de datos del cliente - Dar sugerencias de ID al momento de escribirlo. - Mostrar aviso de error si no se encuentra al cliente. <p>Puntos de Esfuerzo: 1</p> |

| | |
|--------------|--|
| HU026 | <p>Nombre: Notificar desabastecimientos</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero recibir alertas o notificaciones cuando la cantidad de un producto se acerca a su umbral mínimo para saber el informar de cuando surtir.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Tener interfaz de usuario apta - Verificar que la cantidad del producto esté cerca de su umbral mínimo. - Generar una notificación si no hay la cantidad mínima estipulada del producto. - Tener apartado de notificaciones en la interfaz. - Mostrar al usuario la alerta y el producto de la notificación. <p>Puntos de Esfuerzo: 6</p> |
| HU027 | <p>Nombre: Consultar factura por estado</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero consultar buscar las facturas según su estado de pago para saber cuántas siguen pendientes.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Tener filtro de búsqueda de facturas por su estado - Mostrar lista de facturas con el estado escogido. - Tener un organizador de facturas por fecha. - Permitir organizar las facturas de cierto estado por fecha o por monto. - Permitir la visualización del estado. <p>Puntos de Esfuerzo:</p> |
| HU028 | <p>Nombre: Editar información Usuario</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero editar la información de los</p> |

| | |
|--------------|--|
| | <p>usuarios para poder arreglar datos mal registrados o actualizar la información.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Cumplir con HU025. - Permitir cambiar el nombre del usuario - Permitir cambiar la dirección del usuario. - Permitir cambiar el o los teléfonos del usuario. - Permitir cambiar el correo del usuario. - Solicitar confirmación del cambio. - Guardar los cambios hechos <p>Puntos de Esfuerzo: 2</p> |
| HU029 | <p>Nombre: Inactivar usuario</p> <p>Descripción: Yo, como Administrador quiero inactivar usuarios específicos que ya no trabajen en la empresa para quitarles el acceso a información confidencial.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Permitir consulta de usuarios. - Solicitar el documento del usuario a inactivar. - Cambiar de estado a “inactivo”. - Solicitar una confirmación de inactivación. - Registrar motivo de inactivación. - Guardar cambio de estado. <p>Puntos de Esfuerzo: 4</p> |
| HU030 | <p>Nombre: Notificar vencimiento de factura</p> <p>Descripción: Yo, como Administrador o Vendedor, quiero recibir una notificación que indique cuando una factura está cerca de su fecha límite de cobro para recordarle al cliente que debe pagar.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Tener interfaz de usuario apta |

| | |
|-------|--|
| | <ul style="list-style-type: none"> - Verificar que la fecha de vencimiento sea cercana a la actual. - Generar una notificación si faltan pocos días para el vencimiento de la factura. - Tener apartado de notificaciones en la interfaz. - Mostrar al usuario la alerta y la factura de la notificación <p>Puntos de Esfuerzo: 4</p> |
| HU031 | <p>Nombre: Registrar productos en inventario Descripción: Yo, como administrador, necesito poder registrar los productos nuevos que hay en el inventario, con las fechas ya hechas de la entrada del producto, sus referencias y precio propuesto por los propietarios.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - RQF033 - Verificar el nivel de usuario al momento del registro. - No pueden haber dos productos iguales catalogados como diferentes productos. - Si se registra otro producto igual, añadir al stock del producto. - Notificar si se encuentra un producto ya existente. - Verificar los datos del producto antes del registro. <p>Puntos de Esfuerzo: 2</p> |
| HU032 | <p>Nombre: Generar reportes de ventas diarias. Descripción: Yo, como administrador o propietario, necesito que el sistema genere automáticamente un reporte diario de ventas con información sobre productos vendidos, cantidad, total de ingresos y vendedor responsable para evaluar el desempeño del negocio.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - El reporte debe generarse automáticamente al final del día. - Debe incluir detalles de cada venta |

| | |
|-------|--|
| | <p>(producto, cantidad, ingresos).</p> <ul style="list-style-type: none"> - Debe permitir la exportación del reporte en formato Excel. - Debe ser accesible únicamente para administradores y propietarios. - Permitir la programación manual de reportes adicionales según necesidad. <p>Puntos de Esfuerzo:6</p> |
| HU033 | <p>Nombre: Fechas de entrada del producto.</p> <p>Descripción: Yo, como administrador o vendedor, necesito registrar la fecha en que un producto ingresa al stock y la entidad que lo distribuyó, asegurando que se almacenen correctamente las referencias y características del producto.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Guardar la fecha exacta de entrada del producto al sistema. - Registrar información sobre la entidad proveedora del producto. - Permitir la consulta de este historial de entrada. - Mostrar la última fecha de entrada al visualizar un producto en el inventario. - Restringir la modificación de fechas de entrada a usuarios con permisos especiales. <p>Puntos de Esfuerzo: 2</p> |
| HU034 | <p>Nombre: Generar un historial de modificaciones en facturas</p> <p>Descripción: Yo, como administrador o propietario, necesito que el sistema registre un historial de modificaciones en cada factura para poder visualizar qué cambios se hicieron, quién los realizó y en qué fecha.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Registrar cada modificación de una factura en la base de datos. - Mostrar el usuario que realizó el cambio y la fecha. - Permitir la consulta del historial por |

| | |
|-------|---|
| | <p>factura específica.</p> <ul style="list-style-type: none"> - Indicar el tipo de modificación realizada (monto, productos, cliente, etc.). - Generar una alerta si una factura ha sido modificada más de una vez en un día. <p>Puntos de Esfuerzo:</p> |
| HU035 | <p>Nombre: Cargar productos desde archivo CSV.</p> <p>Descripción: Yo, como administrador, necesito importar productos al sistema mediante un archivo CSV para agilizar el proceso de carga masiva de inventario y reducir el tiempo de registro manual.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Permitir la importación de archivos en formato CSV. - Validar los datos del archivo antes de su importación. - Notificar al usuario sobre errores o duplicados antes de completar la carga. - Permitir la vista previa de los productos antes de confirmarlos en el sistema. - Guardar un registro de las importaciones realizadas y quién las ejecutó. <p>Puntos de Esfuerzo: 4</p> |
| HU036 | <p>Nombre: Buscar productos por características.</p> <p>Descripción: Yo, como administrador o vendedor, necesito filtrar productos por diferentes características como categoría, proveedor, precio, disponibilidad y fecha de última actualización para encontrar productos específicos con rapidez.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Permitir búsqueda por múltiples criterios de filtrado. - Mostrar resultados ordenados según |

| | |
|-------|---|
| | <p>relevancia o categoría.</p> <ul style="list-style-type: none"> - Actualizar los resultados en tiempo real conforme se aplican filtros. - Posibilitar la combinación de filtros para búsquedas más precisas. - Permitir exportar los resultados de búsqueda en un archivo Excel o PDF. <p>Puntos de Esfuerzo: 2</p> |
| HU037 | <p>Nombre: Agregar usuario nuevo.</p> <p>Descripción: Yo, como administrador o propietario, necesito poder agregar nuevos usuarios al sistema y asignarles un rol, documento, estado y permisos para controlar su acceso a las funciones del sistema.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Permitir la creación de nuevos usuarios con datos obligatorios. - Asignar roles y permisos según el tipo de usuario. - Restringir la creación de usuarios con información duplicada. - Enviar una notificación por correo al usuario con sus credenciales de acceso. - Permitir la edición de los permisos de un usuario después de su registro. <p>Puntos de Esfuerzo: 2</p> |
| HU038 | <p>Nombre: Consultar información de pagos</p> <p>Descripción: Yo, como administrador, necesito poder consultar el estado de los pagos, incluyendo la hora, monto total, método de pago y si el cliente ha pagado totalmente o tiene un saldo pendiente para llevar un control financiero preciso.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Mostrar información completa de cada pago realizado. - Indicar si el pago fue completo o si hay un saldo pendiente. - Permitir la consulta de pagos por rango de fechas o cliente. - Mostrar el método de pago utilizado |

| | |
|--------------|--|
| | <ul style="list-style-type: none"> - en cada transacción. - Permitir exportar el historial de pagos en formato Excel o PDF. <p>Puntos de Esfuerzo: 4</p> |
| HU039 | <p>Nombre: Actualización de inventario.</p> <p>Descripción: Yo, como administrador o vendedor, necesito que el sistema actualice automáticamente el stock de productos en la base de datos cuando se realice una venta, reflejando el cambio en tiempo real en todos los dispositivos conectados.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Reducir el stock automáticamente al registrar una venta. - Mostrar el stock actualizado en tiempo real. - Indicar cuando un producto llegue a un nivel mínimo de stock. - Enviar una notificación cuando un producto esté por agotarse. - Permitir la configuración de alertas para diferentes niveles de stock. <p>Puntos de Esfuerzo: 2</p> |
| HU040 | <p>Nombre: Filtrar los productos más vendidos.</p> <p>Descripción: Yo, como administrador o vendedor, necesito que el sistema filtre y muestre los productos más vendidos en el inventario, permitiendo conocer cuáles son los de mayor demanda para mejorar la estrategia de ventas.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - Calcular los productos más vendidos comparando ventas y stock. - Mostrar los productos más vendidos en el apartado de inventario. - Permitir ordenar los productos por popularidad en diferentes períodos. - Incluir un gráfico con tendencias de ventas por producto. - Exportar la lista de productos más vendidos en un archivo Excel o PDF. <p>Puntos de Esfuerzo: 2</p> |

Mapa de StakeHolders

Después de haber realizado el levantamiento de información y de haber determinado los objetivos y el alcance del proyecto, era necesario establecer cuales son los stakeholder del proyecto. “Los stakeholders del proyecto son las partes interesadas que pueden influir o verse afectadas por el proyecto en el que estás trabajando. Los participantes pueden provenir de todos los niveles de la organización, desde colaboradores individuales hasta ejecutivos” (Martins, 2025). Ya con una definición adecuada, se llegó al siguiente mapa de Stakeholders:



Figura 8: mapa de stakeHolders (interés - influencia)

El mapa clasifica a los actores según su influencia y nivel de interés. De esta escala se derivan cuatro grupos:

- Mantener satisfecho: Aquellos stakeholder los cuales tienen un nivel alto de interés pero bajo de influencia, es decir, aquellas personas o instituciones las cuales se preocupan por el proyecto pero no influyen de manera directa sobre su funcionamiento.
- Monitoreo Participantes menos interesados en el proyecto, por lo que solo es necesario mantener un monitoreo ocasional para asegurarse de que no haya problemas
- Gestión: Son los actores más importantes del proyecto, como el grupo de desarrollo y los Jefes administrativos. Deben de mantener una relación estrecha con el proyecto, y tomarlos en cuenta para las decisiones más importantes

- Mantener informado: Aquellos Stakeholders los cuales tienen un grado de influencia en el proyecto, y hay que mantenerlos informados para evitar posibles inconvenientes.

Modelamiento de base de datos SQL

Diagrama de clases:

A continuación, se presenta el diagrama de clases para el proyecto final. Su interpretación es la siguiente:

La clase Usuario, con los atributos docUsuario, nombreUsu, correoUsu, estadoUsu, claveUsu, tipoUsu, telefonoUsu, direccionUsu, nitUsu; donde su atributo clave es docUsuario. Cuenta con las operaciones registrarUsu, consultarUsu, inactivarUsu, cambiarClaveUsu, consultarTipoUsu, modificarDatoUsu, consultarTipoUsu, modificarDatoUsu, consultarUsuNit, iniciarSesion, y asignarTipo.

La clase Cliente, con los atributos docCliente, nombreCliente, correoCliente, nitCliente, estadoCliente, telefonoCliente, ciudadCliente, localidadCliente; donde el atributo clave es docCliente. También, cuenta con los métodos de registrarCliente, consultarClientePorCiudad, modificarDatosClienteNit, consultarClienteID.

La clase Venta contiene los atributos codigoVenta, codigoFact, direccionEnvio, fechaEntrega, fechaReparto, productosEnviados, estadoVenta, precioEnv, y codigoProducto; donde el atributo clave es el codigoVenta. Sus métodos son consultarVentaporFEntrega, consultarFechaReparto, inactivarVenta, consultarProductEnviados, cancelarVenta, modificarVenta, eliminarVenta.

Se relacionan mediante asociación donde 1 cliente puede tener 1 o más ventas.

usuario se relaciona con Venta donde un usuario puede gestionar 0 o más ventas

La clase FacturaVenta con los atributos codigoFact, cantidadProdu, docCliente, precioUnitario, docEmpleado, totalFact, fechaEmision, fechaVencimiento, estadoFact, estadoPago, emailCliente, idImpuesto, cantidadImpuestos, valorImpuestos; donde el atributo clave es codigoFact. Sus operaciones son crearFact, calcularTotal, agregarProducto, eliminarProducto, modificarEstPago, inactivarFact, consultarFactEmail, calcularSubTotal, consultarFacturaFEmision, notificarVenciFact, consultarFactEst.

Se relacionan mediante asociación donde 1 venta puede tener múltiples facturas.

La clase detalleFactura (TABLA DEBIL) con los atributos idDetalle, codigoFacturaFK, codigoProductoFK, cantidadProd, precio

Se relacionan mediante asociación donde 1 factura puede tener múltiples detalles.

Se relacionan con Producto mediante asociación donde muchos detalles puede tener 1 producto, es decir, es una tabla débil.

La clase Producto con los atributos codigoProducto, nombreProducto, precioProducto, categoriaProdu, cantidadProducto, descripcionProducto, estadoProducto; donde el atributo principal es el codigoProducto. Tiene las operaciones de registrarProducto, consultarProducto, modificarDatosProduc, inactivarProduc.

Se relaciona mediante asociación donde una venta, puede tener de 1 a muchos productos.

La clase Stock con los atributos codigoStock, idProduct, stockPorProducto, estadoProductos, estadoStock, fechaEntradaProduc, fechaSalidaProduc, capacidadStock; con su atributo clave de codigoStock. Tiene las operaciones de consultarNumProductos, consultarStock, inactivarStock, modificarEstadoProducto, ingresarProducto, modificarCapacidad, consultarProductos, modificarStockProducto y notificarDesabastecimientos.

Se relaciona mediante asociación, donde cada stock contiene la cantidad y fechas de entrada de un producto en específico.

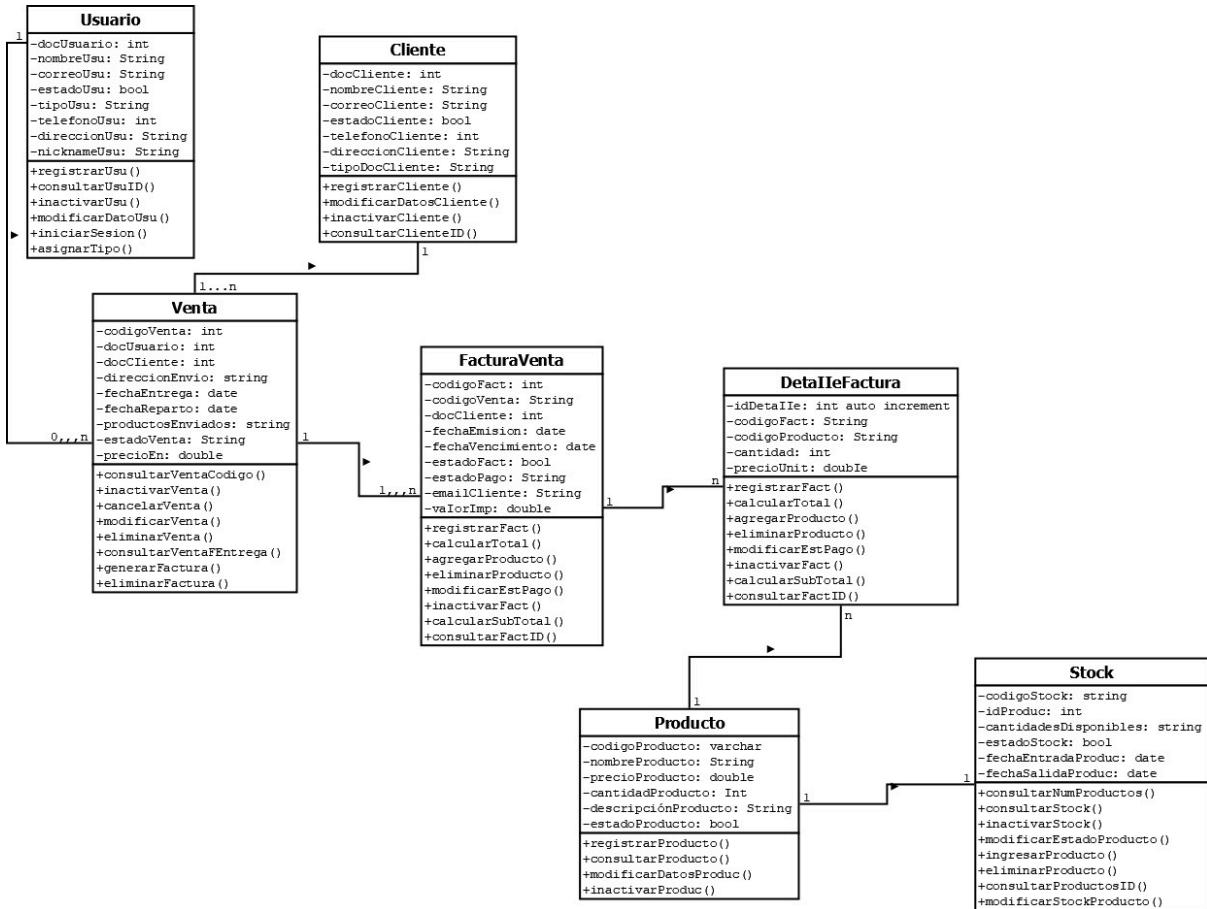


Figura 9: Diagrama de clases

Modelo Conceptual:

El siguiente Modelo de Entidad Relación muestra las relaciones que se aplican entre los módulos establecidos con anterioridad.

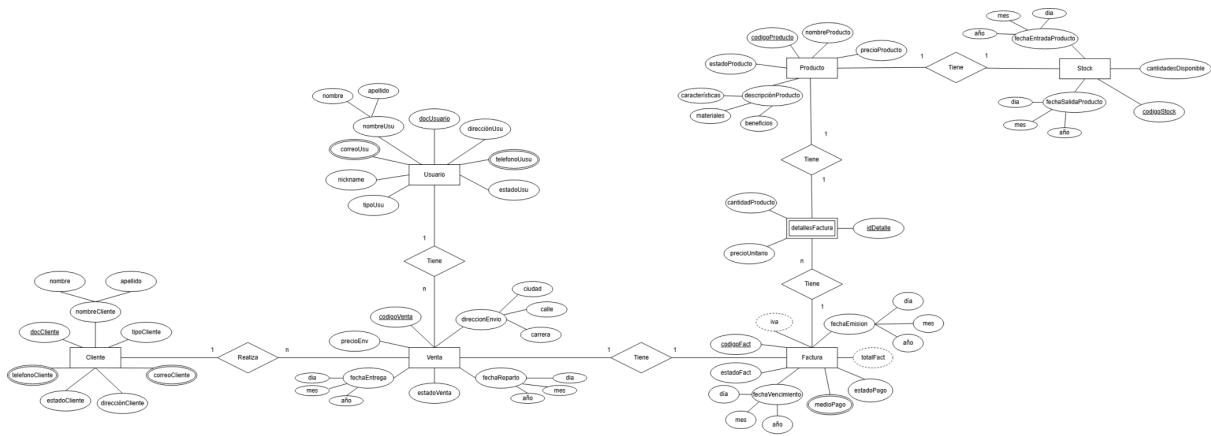


Figura 10: Modelo entidad relación (MER)

Modelo Lógico

Continuando con el desarrollo, se necesitaba avanzar con el modelamiento para poder llegar a la fase de desarrollo con el lenguaje SQL, por tanto, se realizó el siguiente modelamiento lógico para facilitar la programación del presente problema.

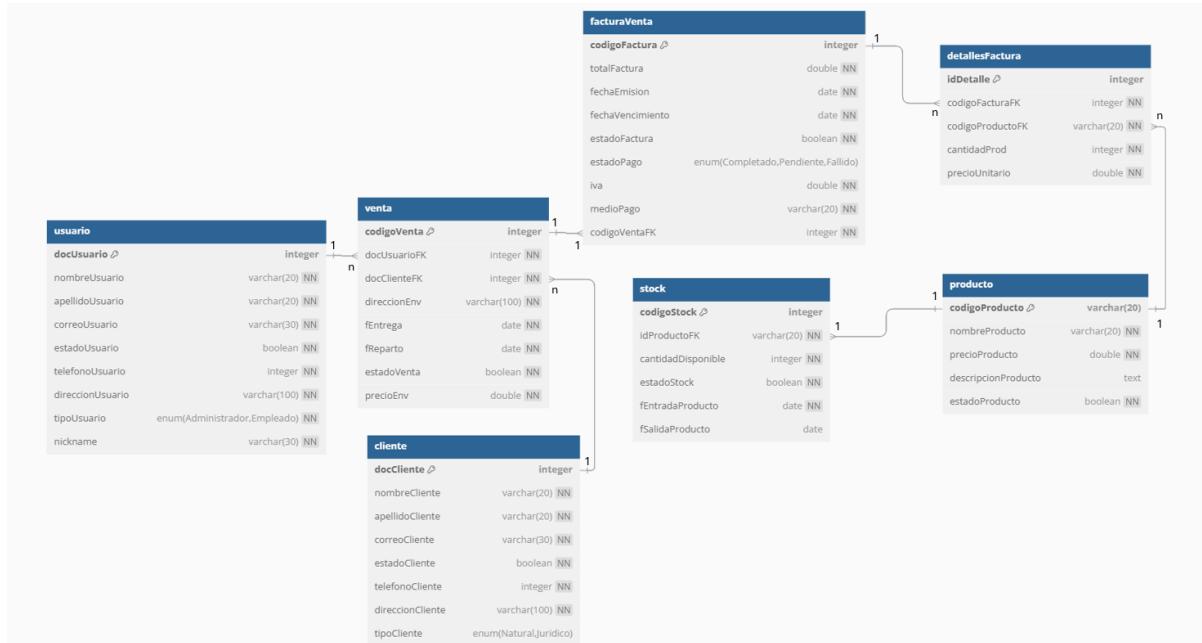


Figura 11: Modelo Lógico

Modelo Físico

Después de crear la base de datos llamada palacioDelTiempo, se procedió a crear las tablas y sus atributos anteriormente definidos en los demás modelos. Con esto, se logró obtener el siguiente modelo físico.

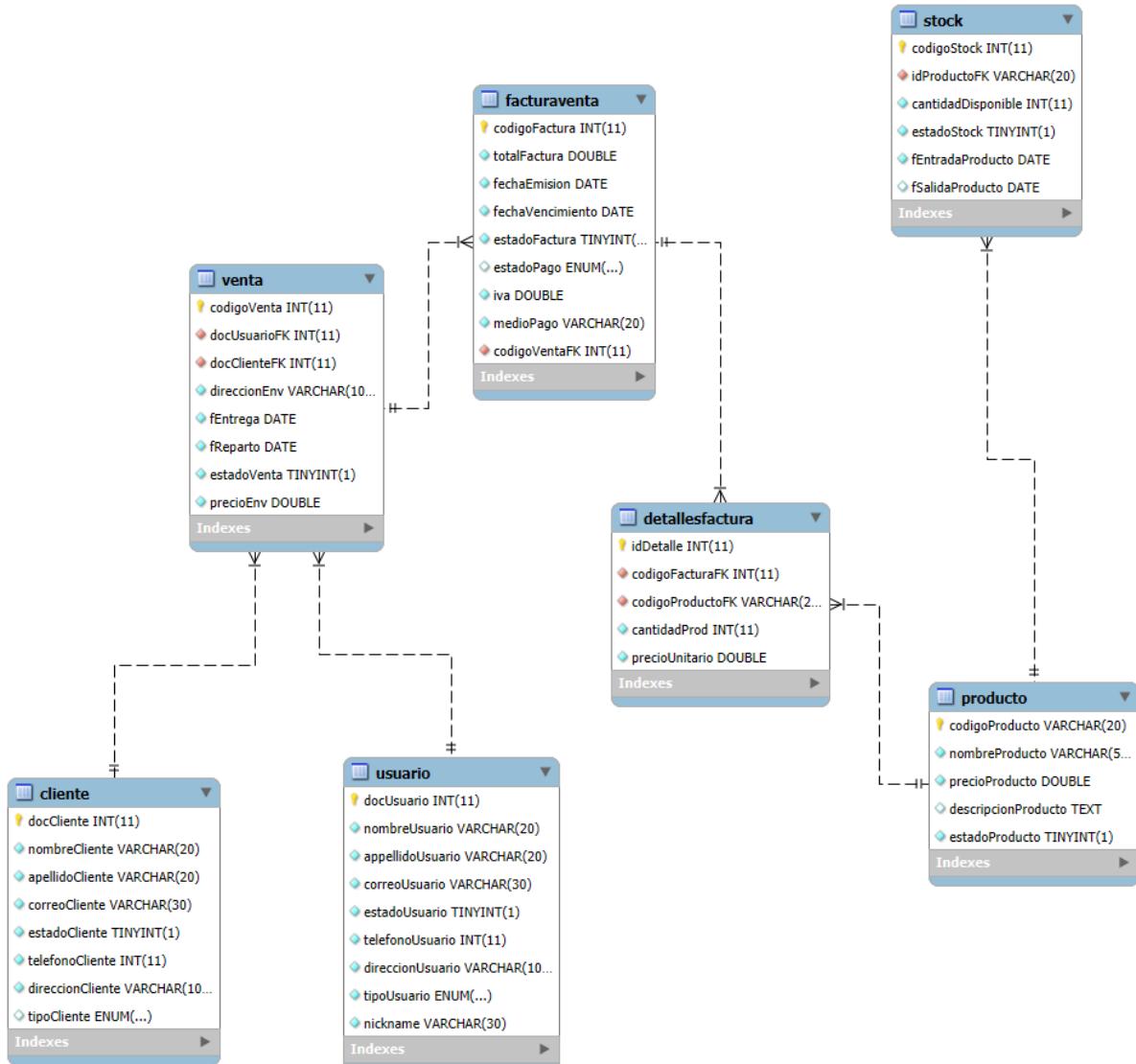


Figura 12: Modelo físico

Cómo se logra apreciar, este modelamiento coincide con el modelo planteado en el diagrama de clases por lo que, se continuó con el desarrollo.

Diccionario de Datos

Para poder realizar un modelado preciso, se tuvo que definir cuales son los datos de cada tabla, junto a sus restricciones, tipo de dato, tamaño y descripción. Toda esta información se encuentra en el siguiente diccionario de datos.

diccionario de datos

Script de procesos DDL y DML de la base de datos

Para empezar con el desarrollo se inició con sentencias DDL como las siguientes:

- `create database palacioDelTiempo;`
- `use palacioDelTiempo;`
- `create table Usuario(`
 `docUsuario int primary key,`
 `nombreUsuario varchar(20) not null,`
 `apellidoUsuario varchar(20) not null,`
 `correoUsuario varchar(30) not null,`
 `estadoUsuario bool not null,`
 `telefonoUsuario int not null,`
 `direccionUsuario varchar(100) not null,`
 `tipoUsuario enum('Administrador','Empleado') not null,`
 `nickname varchar(30) not null`
`);`

Figura 13: ejemplo sentencias DDL

Donde se procedió a crear la base de datos palacioDelTiempo junto a todas sus tablas correspondientes.

```

create table Cliente(
    docCliente int primary key,
    nombreCliente varchar(20) not null,
    apellidoCliente varchar(20) not null,
    correoCliente varchar(50) not null,
    estadoCliente bool not null,
    telefonoCliente int not null,
    direccionCliente varchar(100) not null,
    tipoCliente enum('Natural','Juridico')
);

create table Venta(
    codigoVenta int auto_increment primary key,
    docUsuarioFK int not null,
    docClienteFK int not null,
    direccionEnv varchar(100) not null,
    fEntrega date not null,
    fReparto date not null,
    estadoVenta bool not null,
    precioEnv double not null,
    foreign key (docUsuarioFK) references Usuario(docUsuario)
        on delete cascade on update cascade,
    foreign key (docClienteFK) references Cliente(docCliente)
        on delete cascade on update cascade
).

```

```

    ↵ create table Producto(
        codigoProducto varchar(20) primary key,
        nombreProducto varchar(50) not null,
        precioProducto double not null,
        descripcionProducto text null,
        estadoProducto bool not null
    );

    ↵ create table FacturaVenta(
        codigoFactura int auto_increment primary key,
        totalFactura double not null,
        fechaEmision date not null,
        fechaVencimiento date not null,
        estadoFactura bool not null,
        estadoPago enum ('Completado', 'Pendiente', 'Fallido'),
        iva double not null,
        medioPago varchar(20) not null,
        codigoVentaFK int not null,
        foreign key (codigoVentaFK) references Venta(codigoVenta)
        on delete cascade on update cascade
    );

create table DetallesFactura(
    idDetalle int auto_increment primary key,
    codigoFacturaFK int not null,
    codigoProductoFK varchar(20) not null,
    cantidadProd int not null,
    precioUnitario double not null,
    foreign key (codigoFacturaFK) references FacturaVenta(codigoFactura)
    on delete cascade on update cascade,
    foreign key (codigoProductoFK) references Producto(codigoProducto)
    on delete cascade on update cascade
);

create table Stock(
    codigoStock int auto_increment primary key,
    idProductoFK varchar(20) not null,
    cantidadDisponible int not null,
    estadoStock bool not null,
    fEntradaProducto date not null,
    fSalidaProducto date,
    foreign key (idProductoFK) references Producto(codigoProducto)
    on delete cascade on update cascade
);

```

Figura 14: ejemplo sentencias DDL

Posteriormente, se procedió a realizar las inserciones de datos a cada una de las tablas.

```
insert into Usuario values
(981234585, 'Clara Lucía', 'Morales González', 'clara.morales@hotmail.com', true, 30103456, 'Carrera 70 #45-67, Bogotá', 'Empleado', 'claralucia'),
(9801234585, 'Antonio José', 'González Díaz', 'antonio.gonzalez@hotmail.com', true, 3010012345, 'Carrera 35 #23-45, Bogotá', 'Empleado', 'antoniojose'),
(109234595, 'Martín Felipe', 'Hernández Ruiz', 'martin.hernandez@hotmail.com', true, 3011012345, 'Avenida 12 #56-78, Bogotá', 'Empleado', 'martinfelipe'),
(881234575, 'Pedro Pablo', 'Vázquez Ruiz', 'pedropablo.vazquez@gmail.com', true, 3009012345, 'Carrera 45 #67-89, Bogotá', 'Empleado', 'pedropablo'),
(931234580, 'Miguel Ángel', 'Díaz Gómez', 'miguel.diaz@gmail.com', true, 3009567890, 'Calle 70 #56-78, Bogotá', 'Empleado', 'miguelangel'),
(105234591, 'Ricardo Andrés', 'González López', 'ricardo.gonzalez@hotmail.com', true, 3010678901, 'Carrera 50 #12-34, Bogotá', 'Empleado', 'ricardoandres'),
(107234593, 'Óscar Antonio', 'Gómez Fernández', 'oscar.gomez@gmail.com', true, 3010890123, 'Calle 40 #45-67, Bogotá', 'Empleado', 'oscarantonio'),
(119234605, 'Rosa Margarita', 'Romero Silva', 'rosa.romero@gmail.com', true, 3012012345, 'Calle 56 #78-90, Bogotá', 'Empleado', 'rosamargarita'),
(114234608, 'Ramón Eduardo', 'Díaz Rodríguez', 'ramon.diaz@gmail.com', true, 3011567890, 'Calle 21 #45-67, Bogotá', 'Empleado', 'ramoneduardo'),
(921234579, 'José Antonio', 'García Martínez', 'joseantonio.garcia@gmail.com', true, 3009456789, 'Avenida 60 #23-45, Bogotá', 'Empleado', 'joseantonio'),
(116234602, 'Raúl Ernesto', 'Rojas Herrera', 'raul.rojas@gmail.com', true, 3011789012, 'Calle 90 #45-67, Bogotá', 'Empleado', 'raulernesto'),
(111234597, 'Mónica Alejandra', 'Campos Díaz', 'monica.campos@hotmail.com', true, 3011234567, 'Carrera 12 #34-56, Bogotá', 'Empleado', 'monicaalejandra'),
(861234573, 'José Luis', 'López Herrera', 'joseluis.lopez@hotmail.com', true, 30087890123, 'Carrera 20 #13-45, Bogotá', 'Empleado', 'joseluis'),
(971234584, 'Ricardo Javier', 'Álvarez Pérez', 'ricardo.alvarez@gmail.com', true, 3009901234, 'Avenida 40 #56-78, Bogotá', 'Empleado', 'ricardojavier'),
(110234596, 'Esteban Alejandro', 'Cordero Paredes', 'esteban.cordero@gmail.com', true, 3011123456, 'Calle 23 #12-34, Bogotá', 'Empleado', 'estebanalex'),
(104234590, 'Beatriz Elena', 'Serrano Ríos', 'beatriz.serrano@gmail.com', true, 3010567890, 'Calle 60 #34-56, Bogotá', 'Empleado', 'beatrizelena'),
(100123456, 'Sara Beatriz', 'Villar Ortega', 'sara.villar@hotmail.com', true, 3010901234, 'Carrera 33 #23-45, Bogotá', 'Empleado', 'sarabatriz'),
(971234583, 'María Dolores', 'Hernández Gómez', 'maria.dolores.hernandez@hotmail.com', true, 300990123, 'Calle 90 #34-56, Bogotá', 'Empleado', 'mariadolores'),
(761234567, 'Juan Carlos', 'Pérez Gómez', 'juancarlos.perez@hotmail.com', true, 3001234567, 'Calle 123 #45-67, Bogotá', 'Administrador', 'juancarlos'),
(821234570, 'Ana María', 'Rodríguez Pérez', 'anamaría.rodríguez@gmail.com', true, 3004567890, 'Avenida 60 #23-45, Bogotá', 'Administrador', 'anamaría'),
(101234587, 'Fernando Javier', 'Mora Sánchez', 'fernando.mora@gmail.com', true, 3010234567, 'Calle 88 #12-34, Bogotá', 'Empleado', 'fernandojavier'),
(113234599, 'Lucía Fernanda', 'Ramírez Ríos', 'lucia.ramirez@hotmail.com', true, 3011456789, 'Carrera 67 #23-45, Bogotá', 'Empleado', 'luciafernanda'),
(951234582, 'Santiago José', 'Torres Rodríguez', 'santiago.torres@hotmail.com', true, 3009789012, 'Calle 30 #78-90, Bogotá', 'Empleado', 'santiagojose'),
(871234574, 'Raquel María', 'Moreno Díaz', 'raquel.moreno@hotmail.com', true, 3008901234, 'Calle 10 #21-43, Bogotá', 'Empleado', 'raquelmaria'),
(981234577, 'Carlos Enrique', 'Ramírez Rodríguez', 'carlos.ramirez@hotmail.com', true, 3009234567, 'Calle 55 #12-34, Bogotá', 'Empleado', 'carlosenrique'),
(931234581, 'Laura Isabel', 'Gómez Sánchez', 'laura.gomez@gmail.com', true, 3009876543, 'Carrera 10 #20-45, Bogotá', 'Empleado', 'lauraisabel'),
(112234598, 'Carlos Alberto', 'Martínez Pérez', 'carlos.martinez@hotmail.com', true, 3012345678, 'Calle 22 #11-23, Bogotá', 'Empleado', 'carlosalberto'),
(932234580, 'Estefanía Paola', 'Ramírez Ríos', 'estefania.ramirez@gmail.com', true, 3012348765, 'Carrera 45 #65-23, Bogotá', 'Empleado', 'estefaniapaola'),
(932234586, 'Francisco Javier', 'Torres López', 'francisco.torres@gmail.com', true, 3007654321, 'Calle 14 #12-45, Bogotá', 'Empleado', 'franciscojavier'),
```

Figura 14: ejemplo inserciones

Ya con todos los datos necesarios, se realizaron diversas manipulaciones mediante sentencias DML, por ejemplo:

- select * from “tabla”

```
-- Consultas básicas
select * from Cliente;
SELECT * FROM detallesfactura;
SELECT * FROM facturaVenta;
SELECT * FROM producto;
SELECT * FROM Stock;
SELECT * FROM Usuario;
SELECT * FROM Venta;
```

| | | | |
|---|----------|---|--------------------|
| 2 | 09:28:54 | SELECT * FROM detallesfactura LIMIT 0, 1000 | 59 row(s) returned |
| 3 | 09:28:55 | SELECT * FROM facturaVenta LIMIT 0, 1000 | 70 row(s) returned |
| 4 | 09:28:55 | SELECT * FROM producto LIMIT 0, 1000 | 59 row(s) returned |
| 5 | 09:28:55 | SELECT * FROM Stock LIMIT 0, 1000 | 59 row(s) returned |
| 6 | 09:28:55 | SELECT * FROM Usuario LIMIT 0, 1000 | 53 row(s) returned |
| 7 | 09:28:55 | SELECT * FROM Venta LIMIT 0, 1000 | 55 row(s) returned |

Figura 15: ejemplo consultas básicas

- Update y Set

-- Modificaciones

```
update Producto
set estadoProducto = false
where codigoProducto = 'P28';

UPDATE Producto SET estadoProducto = false WHERE codigoProducto = 'P1';

UPDATE Stock SET cantidadDisponible = 50 WHERE idProductoFK = 'P2';

update Producto
set precioProducto = precioProducto * 1.20;

UPDATE FacturaVenta SET medioPago = 'Tarjeta' WHERE codigoFactura = 1;

update Producto
set nombreProducto = 'CORREA SILICONA ROJO'
where codigoProducto = 'P8';
```

```
13 09:35:52 update Producto set estadoProducto = false where codigoProducto = 'P28'          0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
14 09:35:52 UPDATE Producto SET estadoProducto = false WHERE codigoProducto = 'P1'          0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
15 09:35:52 UPDATE Stock SET cantidadDisponible = 50 WHERE idProductoFK = 'P2'          0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
```

Figura 16: ejemplo actualizaciones de datos

- Consultas multitable y subconsultas:

```
/** Consultas multitableas**/
-- facturas con informacion del cliente y el usuario que realizó la venta
select codigofactura, totalfactura, nombrecliente, nombreusuario from facturaventa join venta on codigoventafk = codigoventa
join cliente on venta.docclientefk = cliente.doccliente join usuario on venta.docusuariofk = usuario.docusuario;
```

| | codigofactura | totalfactura | nombredcliente | nombreusuario |
|---|---------------|--------------|----------------|----------------|
| ▶ | 1 | 320000 | PEDRO | Clara Lucía |
| | 56 | 1500000 | PEDRO | Clara Lucía |
| | 2 | 45000 | YEISON | Antonio José |
| | 57 | 42000 | YEISON | Antonio José |
| | 3 | 1500000 | YEISON | Antonio José |
| | 59 | 1550000 | YEISON | Antonio José |
| | 4 | 275000 | NELFY | Martín Felipe |
| | 60 | 410000 | NELFY | Martín Felipe |
| | 5 | 620000 | LUIGY | Pedro Pablo |
| | 58 | 89000 | LUIGY | Pedro Pablo |
| | 6 | 790000 | JHONATAN | Miguel Ángel |
| | 61 | 1300000 | JHONATAN | Miguel Ángel |
| | 7 | 138000 | YISETH | Ricardo Andrés |
| | 8 | 97000 | LUISA FERN... | Oscar Antonio |

Figura 17: ejemplo consulta multitable

```
-- productos que nunca han sido vendidos
select * from producto
    where codigoproducto not in (
        select codigoproductofk from detallesfactura
    );

```

| | codigoProducto | nombreProducto | precioProducto | descripcionProducto | estadoProducto |
|---|----------------|----------------|----------------|---------------------|----------------|
| 8 | NULL | NULL | NULL | NULL | NULL |

Figura 18: ejemplo subconsulta

como todos los producto se han vendido, el resultado de la consulta es NULL

```
-- detalles de venta incluyendo producto, cantidad y subtotal
select codigoventa, nombreproducto, cantidadprod, preciounitario, (cantidadprod * preciounitario) as 'subtotal'
from venta join facturaventa on venta.codigoventa = facturaventa.codigoventafk join detallesfactura on facturaventa.codigofactura = detallesfactura.codigofacturafk
join producto on detallesfactura.codigoproductofk = producto.codigoproducto;
```

| | codigoventa | nombreproducto | cantidadprod | preciounitario | subtotal |
|---|-------------|-------------------------|--------------|----------------|----------|
| ▶ | 1 | THOM GAFAS DE SOL迪特眼镜 | 5 | 3600 | 18000 |
| | 2 | BOLSAS RAYBAN | 12 | 3600 | 43200 |
| | 3 | RELOJ CASIO MTP-V002 | 8 | 3600 | 28800 |
| | 4 | RELOJ TIMEX WEEKENDER | 17 | 3600 | 61200 |
| | 5 | RELOJ SEIKO 5 SPORTS | 24 | 3600 | 86400 |
| | 6 | CAJA RELOJ LUJO | 9 | 3600 | 32400 |
| | 7 | CORREA CUERO NEGRO | 15 | 4120 | 61800 |
| | 8 | CORREA SILICONA AZUL | 33 | 2780 | 91740 |
| | 9 | RELOJ CITIZEN ECO-DRIVE | 11 | 3490 | 38390 |
| | 10 | RELOJ ORIENT CLASSIC | 27 | 1980 | 53460 |
| | 11 | RELOJ DIESEL DZ4323 | 18 | 4050 | 72900 |
| | 12 | RELOJ FOSSIL CH2565 | 21 | 2210 | 46410 |
| | 13 | RELOJ SKAGEN ANITA | 7 | 4870 | 34090 |
| | 14 | RELOJ MVMT CLASSIC | 29 | 1340 | 38860 |

Figura 19: ejemplo consulta multitable

- procedimiento almacenados:

```
-- Marcar factura como pagada
delimiter //
create procedure marcarfacturapagada(in p_codigofactura int)
begin
    update facturaventa
    set estadopago = 'completado',
        estadofactura = true
    where codigofactura = p_codigofactura;
end //
delimiter ;
```

Figura 20: ejemplo procedimiento almacenado

```
delimiter //
create procedure detallefacturacompleta(in p_codigofactura int)
begin
    select codigofactura, totalfactura, fechamision, estadopago,nombrecliente, nombreproducto, cantidadprod, preciounitario
    from facturaventa join venta on facturaventa.codigoventafk = venta.codigoventa
    join cliente on venta.doclientefk = cliente.docliente
    join detallesfactura on facturaventa.codigofactura = detallesfactura.codigofacturafk
    join producto on detallesfactura.codigoproductofk = producto.codigoproducto
    where facturaventa.codigofactura = p_codigofactura;
end //
delimiter ;
select*from facturaventa;
call detallefacturacompleta(2);
```

| | codigofactura | totalfactura | fechamision | estadopago | nombrecliente | nombreproducto | cantidadprod | preciounitario |
|---|---------------|--------------|-------------|------------|---------------|----------------|--------------|----------------|
| ▶ | 2 | 45000 | 2025-02-25 | Pendiente | YEISON | BOLSAS RAYBAN | 12 | 3600 |

Figura 21: ejemplo procedimiento almacenado

```
DELIMITER //
CREATE PROCEDURE cambiarDisponibilidadProducto(in p_codigoProducto varchar(20), p_estadoNuevo bool)
BEGIN

    update producto
    set estadoProducto = p_estadoNuevo
    where codigoProducto = p_codigoProducto;
END //
DELIMITER ;

call cambiarDisponibilidadProducto('P10', true);

```

Figura 22: ejemplo procedimiento almacenado

- Triggers:

```

delimiter //
create trigger trg_disminuir_stock
after insert on detallesfactura
for each row
begin
    update stock
    set cantidaddisponible = cantidaddisponible - new.cantidadprod
    where idproductofk = new.codigoproductofk;
end //
delimiter ;

3 •  insert into detallesfactura (codigofacturafk, codigoproductofk, cantidadprod, preciouunitario)
4   values (56, 'P1', 2, 10.00);
5   -- ver el stock:
5 •  select * from stock where idproductofk = 'P1';
7

```

ult Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: IA

| codigoStock | idProductoFK | cantidadDisponible | estadoStock | fEntradaProducto | fSalidaProducto |
|-------------|--------------|--------------------|-------------|------------------|-----------------|
| 1 | P1 | 48 | 1 | 2025-04-01 | 2025-12-31 |
| NULL | NULL | NULL | NULL | NULL | NULL |

Figura 23: ejemplo de triggers

```

-- Registrar cambios de estado en productos para tener un registr0 en la empresa
create table historialestadoproducto (
    idhistorial int auto_increment primary key,
    codigoproductofk varchar(20),
    estadoanterior bool,
    estadonuevo bool,
    fechacambio timestamp default current_timestamp
);

delimiter //
create trigger historial
before update on producto
for each row
begin
    if old.estadoproducto <> new.estadoproducto then
        insert into historialestadoproducto (codigoproductofk, estadoanterior, estadonuevo) values (old.codigoproducto, old.estadoproducto, new.estadoproducto);
    end if;
end //
delimiter ;
update producto set estadoproducto = false where codigoproducto = 'P1';
-- consultar el historial
select * from historialestadoproducto where codigoproductofk = 'P1';

```

53 11:03:00 update producto set estadoproducto = false where codigoproducto = 'P1' 0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0 0.000 sec
54 11:03:03 select * from historialestadoproducto where codigoproductofk = 'P1' LIMIT 0, 1000 0 row(s) returned 0.000 sec / 0.000 sec

Figura 24: ejemplo de triggers

- vistas:

```

/*VISTAS*/
-- Vista de clientes con total de compras
create view vista_totalcomprascliente as
select doccliente, nombrecliente, apellidocliente, sum(totalfactura) as 'total comprado'
from cliente join venta on cliente.doccliente = venta.docclientefk
join facturaventa on venta.codigoventa = facturaventa.codigoventafk
group by cliente.doccliente;
SELECT * FROM Vista_TotalComprasCliente;

```

| | doccliente | nombrecliente | apellidocliente | total comprado |
|---|------------|----------------|-------------------|----------------|
| ▶ | 10280550 | JOSE GERARDO | CARMONA | 138000 |
| | 10289718 | CARLOS ALBERTO | CARMONA | 790000 |
| | 15325489 | PEDRO | DIAZ TORREZ | 1820000 |
| | 1002300530 | YEISON | ESTUPIÑAN TORRES | 3694000 |
| | 1002599645 | NELFY | RODRIGUEZ | 685000 |
| | 1010171307 | LUIGY | HERNANDEZ | 709000 |
| | 1010185799 | JHONATAN | AVILA | 2090000 |
| | 1010187182 | YISETH | MAHECHA | 138000 |
| | 1010218476 | LUISA FERNANDA | MURILLO | 447000 |
| | 1010235161 | KAREN | GOMEZ | 150000 |
| | 1012383973 | RAMIREZ | JOHAN | 1232000 |
| | 1012395828 | JORGE ELIECER | BARRERA HERNAN... | 379000 |
| | 1013587671 | ANDREA | PALACIOS | 2220000 |
| | 1014260000 | AUDREY | PINTO GARCIA | 500000 |

Figura 25: ejemplo vistas

```

-- Productos con stock disponible
create view vista_productosconstock as
select codigoproducto, nombreproducto, cantidaddisponible from producto
join stock on producto.codigoproducto = stock.idproductofk where stock.cantidaddisponible > 0;
select * from vista_productosconstock;

```

| | codigoproducto | nombreproducto | cantidaddisponible |
|---|----------------|-------------------------|--------------------|
| ▶ | P1 | THOM GAFAS DE SOL迪特眼镜 | 48 |
| | P2 | BOLSAS RAYBAN | 50 |
| | P3 | RELOJ CASIO MTP-V002 | 30 |
| | P4 | RELOJ TIMEX WEEKENDER | 25 |
| | P5 | RELOJ SEIKO 5 SPORTS | 20 |
| | P7 | CORREA CUERO NEGRO | 80 |
| | P8 | CORREA SILICONA AZUL | 90 |
| | P9 | RELOJ CITIZEN ECO-DRIVE | 15 |
| | P10 | RELOJ ORIENT CLASSIC | 18 |
| | P11 | RELOJ DIESEL DZ4323 | 22 |
| | P12 | RELOJ FOSSIL CH2565 | 24 |
| | P13 | RELOJ SKAGEN ANITA | 28 |
| | P14 | RELOJ MVMT CLASSIC | 26 |
| | P15 | RELOJ TOMMY HILFIGER | 30 |

Figura 26: ejemplo vistas

```
-- Facturas pendientes de pago
create view vista_facturaspendientes as
select codigofactura, nombreciente, totalfactura, fechavencimiento from facturaventa
join venta on facturaventa.codigoventafk = venta.codigoventa join cliente on venta.doclientefk = cliente.docliente
where facturaventa.estadopago = 'pendiente';
select * from vista_facturaspendientes;
```

| | codigofactura | nombreciente | totalfactura | fechavencimiento |
|---|---------------|-----------------|--------------|------------------|
| ▶ | 2 | YEISON | 45000 | 2025-03-10 |
| | 5 | LUIGY | 620000 | 2025-03-17 |
| | 8 | LUISA FERNANDA | 97000 | 2025-03-18 |
| | 10 | RAMIREZ | 42000 | 2025-03-10 |
| | 13 | ALBA LUZ | 410000 | 2025-03-15 |
| | 16 | steven | 1190000 | 2025-03-17 |
| | 18 | AVILA | 670000 | 2025-03-10 |
| | 21 | YEISON | 490000 | 2025-03-21 |
| | 24 | JOAAN SEBASTIAN | 860000 | 2025-03-19 |
| | 26 | DIEGO FERNANDO | 999000 | 2025-03-14 |
| | 29 | CRISTHIAN | 540000 | 2025-03-15 |
| | 32 | JAZBLEYDI | 190000 | 2025-03-10 |
| | 34 | MICHAEL STEVE | 950000 | 2025-03-20 |
| | 36 | DIANA | 620000 | 2025-03-15 |

Figura 27: ejemplo vistas

Estos ejemplos son solo una parte de los ejercicios realizados, para evidenciarlos todos se recomienda revisar el repositorio de github y el script.

link:

<https://github.com/ivysrrgo/PROYECTO-ING-DATOS/blob/39e0fa48759649c59dbf24dc4e1cf8b3798990c3/PAQUETE-PROYECTO/Segundo%20Entregable/Script%20Segunda%20Entrega.sql>

Conclusiones por Fases de Desarrollo del Proyecto

- Principales Problemas:

Se destacó la importancia de realizar un modelado previo de la base de datos, ya que un modelado inadecuado puede generar un diseño complejo e incluso ineficaz. También se subrayó la relevancia de contar con un diccionario de datos al momento de crear las tablas, ya que su ausencia puede generar incoherencias entre los datos almacenados en las diferentes tablas. Además, se evidenció la necesidad de establecer relaciones claras entre las tablas para facilitar las consultas; una relación mal definida dificulta enormemente las consultas multitablas. El manejo de fechas resultó ser un desafío, especialmente en lo que respecta al formato de las mismas durante la inserción de datos. Finalmente, la verificación de las capacidades de las variables es crucial, ya que estas deben ser adecuadas al tipo y tamaño de

los datos que se almacenarán. De lo contrario, pueden surgir problemas de almacenamiento y rendimiento.

- Lecciones Aprendidas

Es fundamental realizar un modelado previo bien estructurado, claro y conciso, para que, al momento de crear la base de datos, no se presenten dificultades relacionadas con las relaciones entre las tablas. Además, debe definirse un diccionario de datos detallado, que permita asegurar la coherencia en las variables y su correspondencia con los datos que se almacenarán en cada campo. También se debe verificar que las capacidades de las variables sean apropiadas para los datos que se almacenarán, para evitar posibles problemas de desbordamiento o pérdida de información. Por último, es crucial tener muy claro el formato de los valores que se van a insertar en la base de datos para evitar inconsistencias y errores en el procesamiento de los datos.

Referencias

- Recalde, H., & Zambrano, W. (2022, Junio). Análisis, diseño, implementación y gestión de un sistema de bases de datos para automatizar los procesos de facturación de la empresa “OKI DOKI”. *Revista Nexos Científicos*, 1, 8-13. Retrieved Febrero 19, 2025, from <http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index>
- Martins, J. (2025, 18 febrero). ¿Quiénes son los stakeholders de un proyecto? [2025] • Asana. Asana. <https://asana.com/es/resources/project-stakeholder>
- Sánchez Godínez, E., & Zúñiga Segura, L. (2011, Octubre 30). La importancia de contar con información precisa, confiable y oportuna en las bases de datos. *Revista Nacional de administración*, 2, 145-154.
- Valverde, V., Portalanza, N., & Mora, P. (2019, Junio 4). *Análisis descriptivo de base de datos relacional y no relacional*. Eumed.net. Retrieved Febrero 21, 2025, from <https://www.eumed.net/rev/atlante/2019/06/base-datos-relacional.html>

Anexo

Evidencia de comunicación con el cliente: WorkShop con el Cliente



EL PALACIO DEL TIEMPO

Calle 11 No. 10-08 Local 201 - Bogotá
Tel: 2 84 03 81.

Santafé de Bogotá, Febrero 25 de 2025

**PROFESORA
TATIANA CABRERA
INGENIERIA DE DATOS
Universidad del Rosario.**

Respetada Profesora:

Por medio de la presente informo que soy propietario del negocio PALACIO DEL TIEMPO identificado con NIT 79.741.364-2 ubicado en la calle 11 # 10-08 local 201 de la ciudad de Bogotá y que autorizo a mi hijo SERGIO ALEJANDRO VELANDIA y su grupo de trabajo que relaciono a continuación para que utilicen la información de mi negocio con **fines educativos** en la asignatura que usted lidera.

DANIEL MORENO
MARIANA ALVAREZ
LINA PEÑA

Agradezco de antemano su atención, cualquier duda o inquietud no dude en contactarme.

Cordialmente,



OSWALD VELANDIA MORA
c.c. 79.741.364 de Bogotá.
CEL. 310 3294878

INFORME REUNION

 INFORME REUNIÓN CON EL CLIENTE