

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.

Objetivos específicos

- 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.
- Desarrollar un front intuitivo para nuestro cliente, de tal manera que se tenga facilidad en el uso de la plataforma.
- 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.

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:

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 estructurarlo 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 eliminen 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:

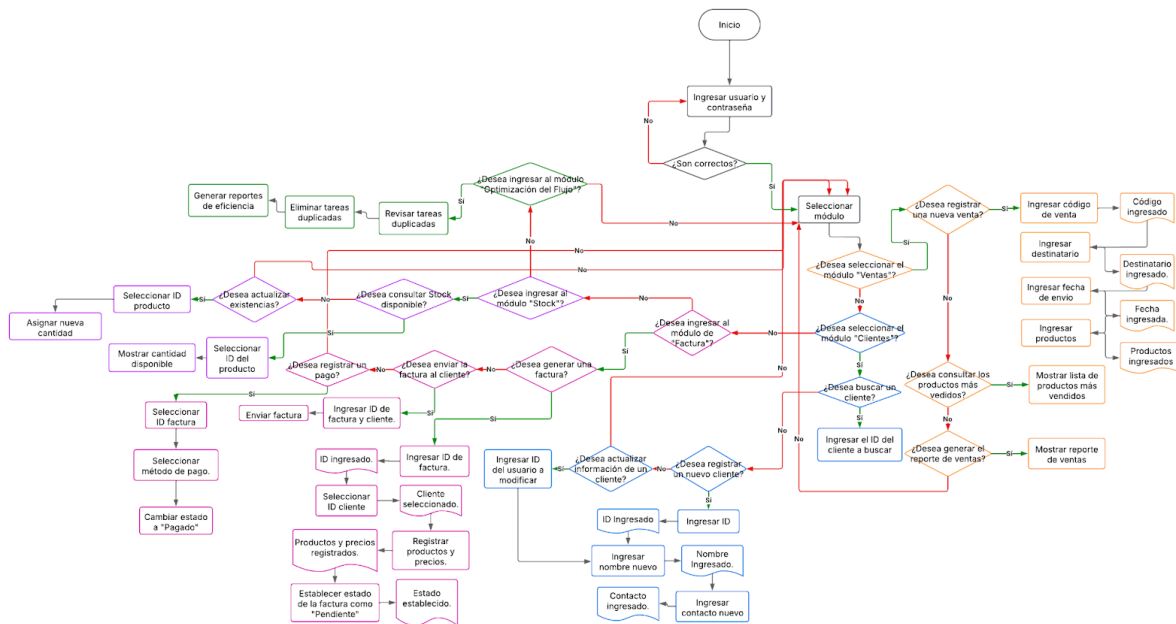


Figura 1: Diagrama de Flujo - Solución Mariana Álvarez

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:

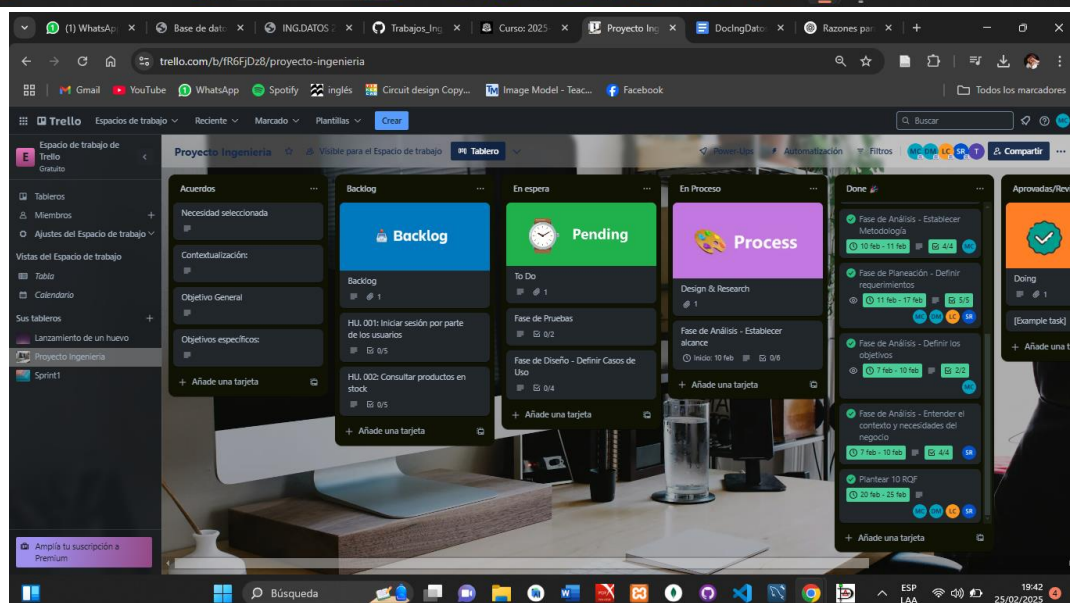
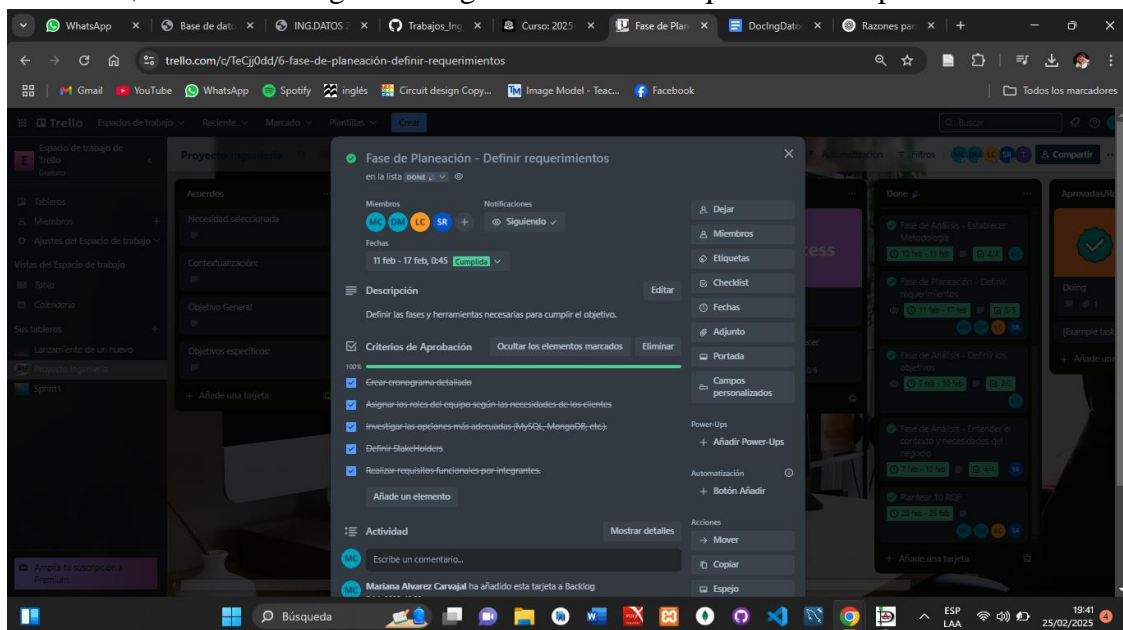
1. Usuario: Que permitirá asignar un tipo según la persona que desee entrar al sistema. Además de contener los datos personales y una clave de cada uno de los registros.
2. Cliente: Garantiza que se tenga recopilada la información de todos los clientes de la empresa con sus datos personales.
3. Venta: 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.

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.



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.

Requerimientos funcionales:

CÓDIGO	Requisitos Funcionales
RQF001	Nombre: Iniciar sesión
	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.
	Usuarios: Vendedores y administrador

CÓDIGO	Requisitos Funcionales
RQF002	Nombre: Consultar productos en stock
	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.
	Usuarios: Vendedores y administrador

CÓDIGO	Requisitos Funcionales
RQF003	Nombre: Inactivar Stock

	<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 discontinuar un producto.</p>
	<p>Usuarios: Administrador</p>

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>
	<p>Usuarios: Administrador y Vendedores</p>

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>

	Usuarios: Administrador y Vendedores
--	---

CÓDIGO	Requisitos Funcionales
RQF010	Nombre: Modificar la información de un producto
	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.
	Usuarios: Administrador

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>HU002</p>	<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>HU003</p>	<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 discontinuados.</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)

	<ul style="list-style-type: none"> • Verificar que el que haga estas acciones sea el administrador.
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.
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

	<ul style="list-style-type: none"> • Visualizar total y subtotal en la factura • Negar números negativos en el total
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
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”

	<ul style="list-style-type: none"> • 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
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.
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
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 puedan cumplir esta función. • 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.

Además, proporcioné la versión inicial del Diagrama de Clases, uniendo y complementando todas las ideas propuestas por los integrantes. Esta primera versión también está en mi rama correspondiente. A continuación, se presenta la primera versión:

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.

Se relacionan mediante asociación, donde 1 usuario puede gestionar 1 cliente.

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 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 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 1 stock puede contener múltiples productos.

