



Gestión de **NEGOCIOS**

Dulce Ambrosio (231143)

Daniel Chet (231177)

Javier Linares (231135)

Cristian Túnchez (231359)

Resumen

Este proyecto propone el desarrollo de un sistema integral de gestión de operaciones diseñado específicamente para pequeñas y medianas empresas de los sectores comercial y de servicios, como cafeterías, restaurantes y comercios minoristas. La necesidad de esta solución radica en la falta de integración entre los sistemas de ventas, inventario y administración, lo que genera ineficiencias operativas, incrementa los costos y obstaculiza la toma de decisiones estratégicas.

En la actualidad, muchos negocios dependen de software obsoleto con interfaces poco amigables, carencia de automatización en procesos críticos y limitaciones para sincronizar datos en tiempo real. Estas deficiencias obligan a los empleados a realizar tareas manuales repetitivas, como conteos físicos de inventario, y aumentan la probabilidad de errores en procesos clave, como el cierre de caja o la gestión de existencias.

El objetivo principal de este proyecto es crear una plataforma moderna, intuitiva y escalable que centralice la gestión de ventas, inventario y facturación en un único sistema. La solución busca optimizar las operaciones mediante la automatización de tareas repetitivas, la integración con hardware de punto de venta y la implementación de herramientas de análisis predictivo para mejorar la planificación de inventarios y la toma de decisiones.

Introducción

La solución propuesta está dirigida a pequeñas y medianas empresas del sector comercial y de servicios, especialmente cafeterías, restaurantes y comercios minoristas. Estas empresas dependen de sistemas de gestión de operaciones para optimizar la toma de pedidos, cobros, control de inventario y otras funciones administrativas. Sin embargo, muchas de ellas utilizan software que no es intuitivo, tiene diseños básicos, presenta problemas de rendimiento o carece de integración entre las funcionalidades de cajero y administración. Por lo que nuestra solución estará enfocada en mejorar la eficiencia operativa de estos negocios a través de una plataforma moderna y adaptable a sus necesidades.

La propuesta consiste en desarrollar un software de gestión de operaciones para negocios que optimice las funciones de los cajeros, administradores y gerentes. Actualmente, muchos sistemas en el mercado presentan interfaces poco intuitivas lo que dificulta la adaptación de nuevos empleados y afecta la eficiencia operativa. Además, en muchos casos, las funcionalidades administrativas, como la gestión de inventario y la revisión de facturas, se encuentran en plataformas separadas, lo que genera retrasos y errores en la operación diaria.

El objetivo general de este informe es: describir y fundamentar el desarrollo de un software de gestión de operaciones para negocios, identificando la problemática, los usuarios objetivo y la propuesta de solución, estableciendo los criterios de diseño y desarrollo del proyecto.

Como objetivos específicos están: analizar la problemática existente en la gestión operativa de negocios y la necesidad de un software más eficiente; identificar las características clave que debe tener la solución para diferenciarse de los sistemas existentes en el mercado; establecer los requisitos funcionales y no funcionales del software propuesto; plantear una metodología de desarrollo basada en *Design Thinking* para asegurar una solución centrada en el usuario; y documentar el proceso de investigación y análisis previo que respalde la viabilidad del proyecto.

Design Thinking

Prototipos

Prototipos en Bruto

En base a las ideas o propuestas de mayor utilidad se realizó un prototipo en bruto sobre el software que se estará realizando.

El usuario busca en el inventario y no hay existencia en el inventario por lo que se ofrecen opciones...

Q Buscar en el Inventario...

No hay en existencia.

Productos Sugeridos:

Producto	=====
Producto	=====
Producto	=====

Otras Sucursales:

Sucursal	Ubicación: _____
	Cont. dispo: 0
	Telefono: _____

Ubicación: _____

Cont. dispo: 4

Telefono: _____

Pedir

El usuario realiza la factura y esta es enviada al correo del cliente y se imprime...

Q Buscar en el Inventario...

Facturación:

Nombre: _____

NIT: _____

Correo: _____

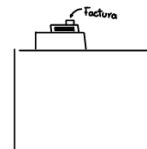
Telefono: _____

Enviar

Compra

Producto	=====	Q. _____
Producto	=====	Q. _____
Producto	=====	Q. _____

Total Q. _____



Los bodegueros reciben una notificación cuando el encargado lo asigna a una tarea...

Bodeguero/Encargado | Cajero

Ordenes Activas

Producto	=====
Producto	=====
Producto	=====

Asignación

Asignar	si	No	Q
Asignar	si	No	O
Asignar	si	No	O

Enviar Not. Q

10:00
20 Feb 2025
Tarea Asig.

El inventario es actualizado en tiempo real y hay alertas cuando un producto tiene stock minimo o ya no hay ninguna unidad...

Q Buscar en el Inventario...

Producto	=====	Cont. 3 ⁻¹
Producto	=====	Cont. 2 ⁻¹
Producto	=====	Cont. 4 ⁻¹

Alerta

Stock Min:

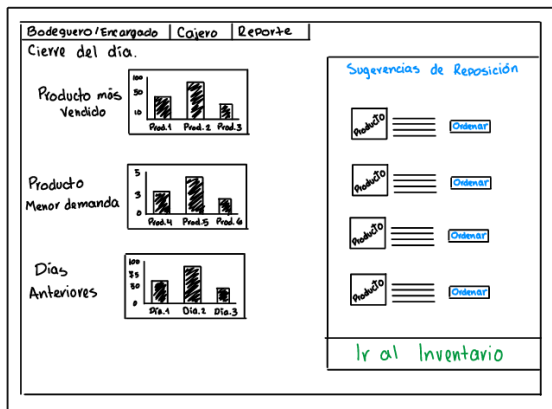
Producto	=====	Cont. 1
----------	-------	---------

Ordenar

Prod. Inexistentes

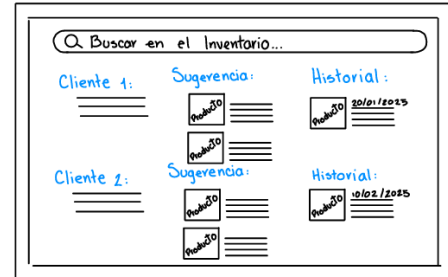
Producto	=====	Cont. 0
----------	-------	---------

Ordenar



Se da un reporte del producto más vendido y el menos vendido, también una comparación de la ventas de los días anteriores y sugerencias para reposición del producto...

Se guarda un historial de los clientes y sus compras para realizar sugerencias de otros productos...



Primer Testeo de los Prototipos

Como parte del proceso de testeo, se presentó el prototipo en bruto del sistema a un usuario potencial con el objetivo de evaluar su funcionalidad, usabilidad e identificar posibles mejoras. Durante la prueba, el usuario brindó retroalimentación y comentarios sobre la interfaz. A continuación, se recopilan algunos de los comentarios obtenidos:

- "Sería bueno que hubiera una opción para apartar un producto y que el cliente lo pueda pagar antes a veces vienen a preguntar por algo y cuando regresan ya no hay o está en otra tienda."
- "El sistema de prioridad está bueno, pero ¿cómo se asignan las tareas? Si los empleados pueden elegir la que ellos quieran, habría gente que pueden seguir agarrando solo las más fáciles. Tal vez debería haber un sistema automático que las asigne según disponibilidad."

- "El programa de puntos es buena idea, pero ¿cómo sabe la gente cuántos puntos lleva? Estaría bueno que al pagar le aparezca en la pantalla su saldo de puntos y qué puede hacer con ellos."

Con base a estos comentarios, se realizaron ajustes y se refinó el prototipo para mejorar la experiencia de usuario y garantizar una interacción más intuitiva con el sistema.

Prototipos Refinados 1

El usuario busca en el inventario y no hay existencia en el inventario
Por lo que se ofrecen opciones...

Q. Buscar en el inventario...

No hay en existencia.

Productos Sugeridos:

Otras sucursales:

Ubicación: Cont. dispo: 0 Telefono: Pedir

Ubicación: Cont. dispo: 4 Telefono: Pedir

Permitir a los clientes pagar para reservar el producto y así recibirlo cuando llegue de otra sucursal.

Q. Buscar en el inventario...

No hay en existencia.

Otras sucursales:

Ubicación: Cont. dispo: 4 Telefono: Pedir

Ubicación: Cont. dispo: 0 Telefono: Pedir

Pedido

Tiempo estimado: 15 minutos

Reservado por Cliente 1

El usuario realiza la factura y esta es enviada al correo del cliente y se imprime...

Q. Buscar en el inventario...

Facturación:

Nombre: NIT: Correo: Telefono:

Compra

Total

Enviar

Los bodegueros reciben una notificación cuando el encargado le asigna a una tarea

Bodeguero/Encargado Cojero

Ordenes Activas

Asignación

Tareas Comp.

Asignar Si No (15)

Asignar Si No (10)

Asignar Si No (5)

Enviar Not.

10:00

20 Feb 2025

Tarea Asig.

El inventario es actualizado en tiempo real y hay alertas cuando un producto tiene stock mínimo o ya no hay ninguna unidad...

Se da un reporte del producto más vendido y el menos vendido, también una comparación de las ventas de los días anteriores y sugerencias para reposición del producto...

Se guarda un historial de los clientes y sus compras para realizar sugerencias de otros productos...

Segundo Testeo de los Prototipos

Como parte del proceso de testeo, se presentó el prototipo refinado 1 del sistema a un usuario potencial con el objetivo de identificar posibles mejoras. Durante la prueba, el usuario brindó retroalimentación y comentarios sobre la interfaz. A continuación, se recopilan algunos de los comentarios obtenidos:

- "Las alertas de cambios en tendencias están buenas, pero debería haber una opción para ver más detalles. Por ejemplo, si un producto bajó sus ventas, me gustaría ver desde cuándo y en qué sucursales pasó."

- "Además de avisar sobre productos que venden menos, ¿también hay alertas para productos que empiezan a venderse más? Si algo se empieza a vender mucho, sería bueno que el sistema sugiera hacer un pedido antes de que se agote."
- "Los descuentos personalizados están buenos, pero ¿cómo se aplican? ¿El sistema los asigna solo o el cliente los puede ver y elegir cuándo usarlos? Sería bueno que el que atiende le comenté al cliente si quiere usarlo o no."

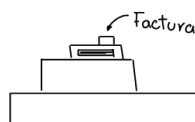
Con base a estos comentarios, se realizaron ajustes y se refinó el prototipo refinado 1.

Prototipos Refinados 2

El usuario busca en el inventario y no hay existencia en el inventario por lo que se ofrecen opciones...

Permitir a los clientes pagar para reservar el producto y así recibirlo cuando llegue de otra sucursal, el cliente puede realizar un pago parcial...

El usuario realiza la factura y esta es enviada al correo del cliente y se imprime...



Los bodegueros reciben una notificación cuando el encargado le asigna a una tarea y se agrego un chat interno para comentarios...

El inventario es actualizado en tiempo real y hay alertas cuando un producto tiene stock mínimo o ya no hay ninguna unidad...

Q. Buscar en el inventario...

Producto	Cant. 5 ⁻¹	<p>Alerta ⚠</p> <p>Stock Min: <input type="text"/> Cant. 1 <input type="button" value="Ordenar"/></p> <p>Prod. Insuficientes <input type="text"/> Cant. 0 <input type="button" value="Ordenar"/></p>
Producto	Cant. 2 ⁻¹	
Producto	Cant. 4 ⁻¹	

Se da un reporte del producto más vendido y el menos vendido, también una comparación de las ventas de los días anteriores y sugerencias para reposición del producto; también se hace comparación entre sucursales...

Boqueño/Encargado | Cajero | Reporte

Cierre del día.

Producto más Vendido

Producto Menor demanda

Días Anteriores

Comparación Sucursal

Sugerencias de Reposición

Se guarda un historial de los clientes y sus compras para realizar sugerencias de otros productos...

Q. Buscar en el inventario...

Ciente 1: Sugerencia: Historial: Puntos: 1022

Ciente 2: Sugerencia: Historial: Puntos: 2022

Boqueño/Encargado | Cajero | Reporte

Cierre del día.

Producto más Vendido

Producto Menor demanda

Días Anteriores

Comparación Sucursal

Sugerencias de Reposición

Alerta ⚠

Cambios en Tendencia

El usuario recibe alertas sobre el cambio de tendencias para identificar caídas o picos inusuales en la venta de productos...

Q. Buscar en el inventario...

Ciente 1: Historial: 20/01/2023

Ciente 2: Historial: 10/02/2023

Cientes %

Ciente 1: 50 %

Ciente 2: 25 %

Ciente 3: 30 %

Dependiendo la cantidad de productos que el cliente haya comprado se le sugiere al vendedor realizar un descuento...

Tercer Testeo de los Prototipos

- "¿Cualquiera puede entrar al sistema sin necesidad de identificarse? Tal vez sería mejor que cada usuario tenga su cuenta para que haya más control sobre quién hace qué dentro del sistema."
- "Visualmente ya no se ve tan suelto, pero algunos elementos siguen sintiéndose un poco dispersos. Tal vez con más separación entre secciones o algún fondo se vería más ordenado."
- "El mensaje general sobre productos con poco stock está bien, pero no muestra detalles de cuáles son los más urgentes. Tal vez debería destacar los productos críticos en la alerta para priorizarlos."

Prototipos Refinados 3

A hand-drawn sketch of a login form. At the top center is a circular icon representing a user. Below it is the word "Login". To the right of the "Login" text is a dropdown menu with four options: "Gerente General", "Gerente de Turno", "Cajero", and "Bodeguero". Below the "Login" text are two input fields. The first is labeled "Usuario" and has a small checkmark icon to its right. The second is labeled "Contraseña" and has a small eye icon to its right. A red arrow points from the dropdown menu to the "Usuario" input field.

El usuario busca en el inventario y no hay existencia en el inventario por lo que se ofrecen opciones...

Q. Buscar en el inventario...

No hay en existencia.

Productos Sugeridos:

Otras sucursales:

Ubicación: <input type="text"/>	Cont. disp.: <input type="text"/>	Ubicación: <input type="text"/>	Cont. disp.: <input type="text"/>
Cont. disp.: <input type="text"/>	Ubicación: <input type="text"/>	Cont. disp.: <input type="text"/>	Ubicación: <input type="text"/>
Cont. disp.: <input type="text"/>	Ubicación: <input type="text"/>	Cont. disp.: <input type="text"/>	Ubicación: <input type="text"/>

Permitir a los clientes pagar para reservar el producto y así recibirlo cuando llegue de otra sucursal, el cliente puede realizar un pago parcial...

Q. Buscar en el inventario...

No hay en existencia.

Otras sucursales:

Ubicación: Cont. disp.: Ubicación: Cont. disp.:

Telefono: Telefono:

Tiempo estimado: 15 minutos

Reservado por Cliente 1

Pago Anticipado:

Pago Total: Q. 100.00

Pago Parcial: Q. 50.00

Pago Restante: Q. 50.00

Los bodegueros reciben una notificación cuando el encargado le asigna a una tarea y se notifica

Bodeguero/Encargado | Cajero

Ordenes Activas

	Urgente!!		Asignado
	Importante		Asignado
	Normal		Asignado

Enviar Not.

10:00
20 Feb 2020
Tarea Asig.

El inventario es actualizado en tiempo real y hay alertas cuando un producto tiene stock mínimo o ya no hay ninguna unidad...

Q. Buscar en el inventario...

	Cont. 5 ⁺
	Cont. 2 ⁺
	Cont. 4 ⁺

Alerta

Los siguientes productos están casi o sin stock.

	Cont. 3	<input type="button" value="Ordenar"/>
	Cont. 1	<input type="button" value="Ordenar"/>
	Cont. 0	<input type="button" value="Ordenar"/>

Se da un reporte del producto más vendido y el menos vendido, también una comparación de las ventas de los días anteriores y sugerencias para reposición del producto; también se hace comparación entre sucursales...

Bodeguero/Encargado | Cajero | Reporte

Cierre del día

Producto más Vendido

Prod. 1	Prod. 2	Prod. 3
---------	---------	---------

Producto Menor demanda

Prod. 4	Prod. 5	Prod. 6
---------	---------	---------

Días Anteriores

Día 1	Día 2	Día 3
-------	-------	-------

Comparación Sucursal

Suc. 1	Suc. 2	Suc. 3
--------	--------	--------

Sugerencias de Reposición

Se guarda un historial de órdenes para realizar sugerencias de productos...

Q. Buscar en el inventario...

Orden 1:

Historial:

Sugerencia:

Orden 2:

Historial:

Sugerencia:

Facturación:

Nombre:

N.I.T.:

Correo:

Telefono:

Compra

Total Q. --

Análisis

Requisitos Funcionales

Gestión de Ventas y Punto de Venta (POS)

- RF01: Permitir al cajero acceder rápidamente a la pantalla de ventas para iniciar una nueva transacción.
 - Historia de usuario: "Como cajero, quiero acceder rápidamente a la pantalla de ventas para comenzar una nueva transacción sin retrasos."
- RF02: Permitir buscar y agregar productos al pedido de forma eficiente.
 - Historia de usuario: "Como cajero, quiero buscar y agregar productos fácilmente al pedido para procesar ventas de manera eficiente."
- RF03: Permitir editar pedidos en tiempo real (agregar o eliminar productos).
 - Historia de usuario: "Como cajero, quiero poder editar pedidos en tiempo real para corregir errores o agregar productos adicionales."
- RF04: Ofrecer múltiples métodos de pago (efectivo, tarjeta, etc.).
 - Historia de usuario: "Como cajero, quiero ofrecer diferentes métodos de pago para que un cliente complete su compra de manera conveniente."
- RF05: Generar recibos digitales o físicos tras procesar el pago.
 - Historia de usuario: "Como cajero, quiero procesar el pago rápidamente y generar un recibo digital o físico para finalizar la venta."
- RF06: Permitir aplicar descuentos y promociones en tiempo real.

- Historia de usuario: "Como cajero, quiero aplicar descuentos y promociones en tiempo real para ofrecer mejores precios a los clientes."
- RF07: Permitir dividir un pago entre varias tarjetas o métodos.
 - Historia de usuario: "Como cajero, quiero poder dividir el pago entre varias tarjetas o métodos de pago para brindar mayor flexibilidad a los clientes al momento de pagar."
- RF08: Permitir agregar notas personalizadas a una venta.
 - Historia de usuario: "Como cajero, quiero poder agregar notas a la venta (por ejemplo, solicitudes especiales del cliente) para mejorar la personalización del servicio."

Gestión de Inventario

- RF09: Actualización del inventario en tiempo real.
 - Historia de usuario: "Como gerente de turno, quiero que el inventario se actualice de manera automática y en tiempo real para dedicar más tiempo a otras áreas del negocio sin preocuparme de esta."
- RF10: Permitir actualizar manualmente el stock del inventario.
 - Historia de usuario: "Como bodeguero, quiero actualizar la cantidad de productos en stock para reflejar cambios en el inventario."
- RF11: Mostrar alertas automáticas cuando un producto esté por expirar.

- Historia de usuario: "Como gerente de turno, quiero poder estar al tanto de la fecha de expiración de mis productos en stock para poder hacer cambios en el inventario cuando sea necesario."
- RF12: Mostrar alertas automáticas cuando un producto esté por agotarse.
 - Historia de usuario: "Como gerente de turno, quiero recibir notificaciones cuando un producto esté por agotarse para tomar acción con antelación."
- RF13: Ofrecer sugerencias de reabastecimiento basadas en patrones de compra.
 - Historia de usuario: "Como gerente de turno, quiero recibir sugerencias de reabastecimiento basadas en patrones de compra para evitar quiebres de stock."

Gestión Financiera y Reportes

- RF14: Permitir generar reportes personalizados de ventas y gastos.
 - Historia de usuario: "Como gerente, quiero generar reportes personalizados de ventas y gastos para tomar decisiones informadas."
- RF15: Permitir descargar reportes en formatos como Excel o PDF.
 - Historia de usuario: "Como gerente, quiero descargar reportes en Excel o PDF para compartirlos con el equipo de administración."
- RF16: Ofrecer un tablero con métricas clave para visualizar la salud financiera del negocio.

- Historia de usuario: "Como gerente, quiero visualizar un tablero con métricas clave para analizar la salud financiera del negocio."
- RF17: Establecer parámetros para comparar el desempeño financiero del negocio.
 - Historia de usuario: "Como gerente, quiero comparar el desempeño financiero entre distintos períodos de tiempo, sucursales, productos, etc., para tomar mejores decisiones estratégicas."
- RF18: Recibir reportes automáticos de ventas diarias y tendencias por correo.
 - Historia de usuario: "Como gerente, quiero recibir reportes automatizados en mi correo con un resumen de ventas diarias y tendencias."

Gestión de Empleados y Turnos

- RF20: Permitir el check-in (registro) y check-out (cierre de sesión) de los empleados para registrar horas de trabajo.
 - Historia de usuario: "Como cajero/gerente, quiero hacer check-in y check-out en mi turno para registrar mis horas de trabajo correctamente."
- RF21: Asignar y modificar turnos de trabajo.
 - Historia de usuario: "Como gerente, quiero asignar y modificar turnos fácilmente para mantener una correcta operación del negocio."
- RF22: Permitir la visualización de la disponibilidad del personal.

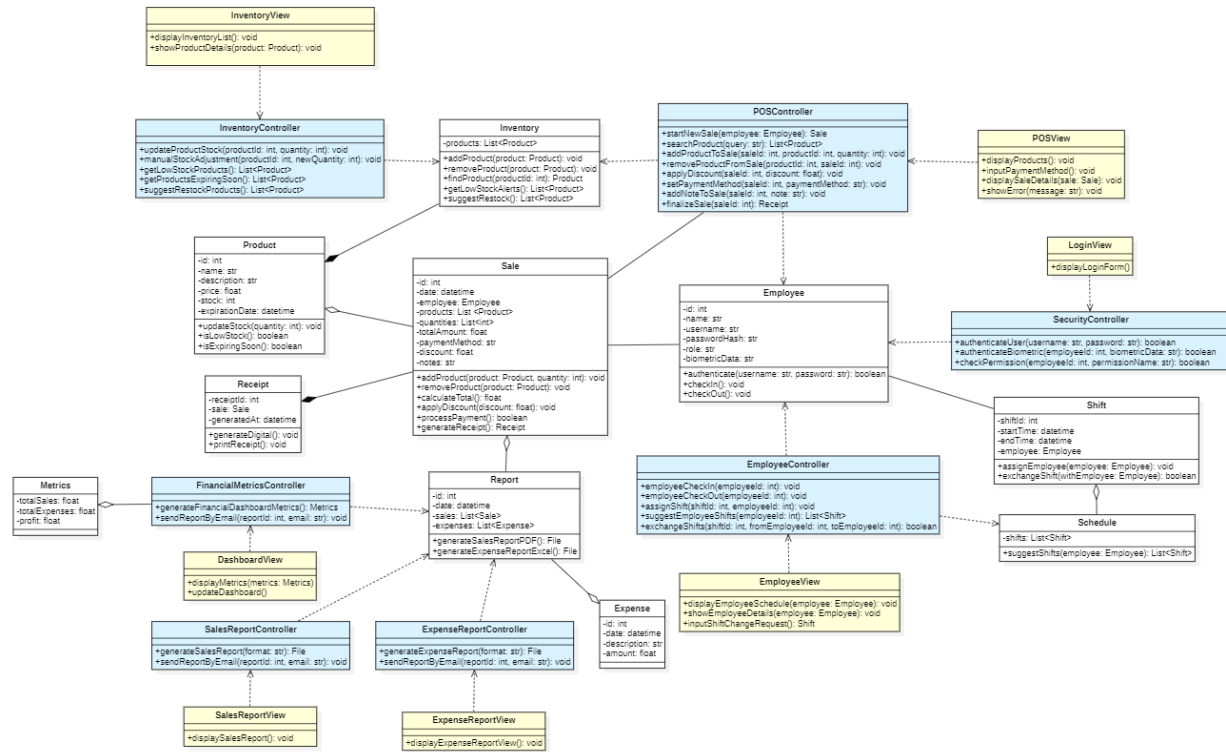
- Historia de usuario: "Como gerente, quiero ver la disponibilidad del personal para organizar los turnos de manera eficiente."
- RF23: Ofrecer sugerencias automáticas de asignación de turnos basadas en disponibilidad, demanda y desempeño.
 - Historia de usuario: "Como gerente, quiero recibir recomendaciones automáticas de asignación de turnos en función de la disponibilidad, demanda y desempeño del personal."
- RF24: Permitir el intercambio de turnos con aprobación del gerente.
 - Historia de usuario: "Como gerente, quiero permitir que los empleados intercambien turnos entre ellos con mi aprobación para mayor flexibilidad."

Seguridad y Control de Acceso

- RF25: Gestionar los permisos de acceso según el rol del usuario.
 - Historia de usuario: "Como gerente, quiero definir los permisos de acceso de cada empleado para proteger la información del negocio."
- RF26: Permitir la autenticación biométrica para el acceso al sistema.
 - Historia de usuario: "Como gerente, quiero habilitar autenticación biométrica para mejorar la seguridad de accesos en el sistema."
- RF27: Enviar alertas de actividades sospechosas dentro del sistema.
 - Historia de usuario: "Como gerente, quiero recibir alertas en caso de actividades sospechosas dentro del sistema para evitar fraudes."

Clases Preliminares

Figura 1. Diagrama de Clases UML



Modelos

Estas clases encapsulan la lógica de negocio y los datos del sistema, proporcionando estructuras para operaciones y análisis.

- **Product:** Esta clase almacena y gestiona información detallada de cada producto en inventario, incluyendo identificación, nombre, precio, cantidad en stock, fecha prevista para reposición y estado de expiración.

- Sale: Registra cada transacción de venta realizada, detallando los productos vendidos, el empleado responsable, el total de la venta, los métodos de pago utilizados y cualquier descuento aplicado.
- Receipt: Responsable de la generación de recibos que documentan cada venta, almacenando información como los detalles de la transacción, fecha, y total.
- Employee: Mantiene un registro de los detalles del personal, incluidos identificación, nombres de usuario, contraseñas, roles de seguridad, y datos biométricos.
- Shift: Gestiona la información de los turnos laborales, especificando los horarios de inicio y fin de cada turno. Es fundamental para la organización de la jornada laboral del personal, ayudando a optimizar la asignación de recursos.
- Expense: Lleva un control detallado de los gastos operativos y otros desembolsos del negocio, permitiendo un análisis financiero y facilitando un manejo efectivo de los costos.
- Report: Encapsula funcionalidades para crear informes detallados que abarcan varios aspectos del negocio, como ventas, gastos, y otras métricas operativas. Los informes pueden ser configurados para generar análisis periódicos o específicos.
- Metrics: Provee análisis de métricas claves como ventas totales, gastos y rentabilidad. Esta clase es fundamental para la toma de decisiones y la evaluación del rendimiento del negocio.

- **Schedule:** Administra los horarios de trabajo, coordinando la asignación de turnos y asegurando la cobertura adecuada según las necesidades del negocio.

Vistas

Estas clases interactúan directamente con los usuarios, proporcionando interfaces gráficas para facilitar la operación del sistema.

- **InventoryView:** Ofrece una interfaz gráfica que permite a los usuarios gestionar y visualizar el inventario, incluyendo la búsqueda de productos, la actualización de información y la revisión de niveles de stock.
- **POSView:** Sirve como el punto de interacción para las transacciones en el punto de venta, permitiendo a los empleados procesar ventas, aplicar descuentos, y gestionar pagos de manera eficiente y efectiva.
- **EmployeeView:** Proporciona una interfaz para la visualización y gestión de información relacionada con los empleados, incluyendo programación de turnos, visualización de horarios, y gestión de datos personales.
- **LoginView:** Presenta la interfaz de inicio de sesión, asegurando que solo los usuarios autenticados puedan acceder a las funciones protegidas del sistema.
- **ReportsView:** Facilita la selección y visualización de diferentes tipos de informes financieros y operativos, permitiendo a los usuarios generar y consultar datos críticos para la toma de decisiones.

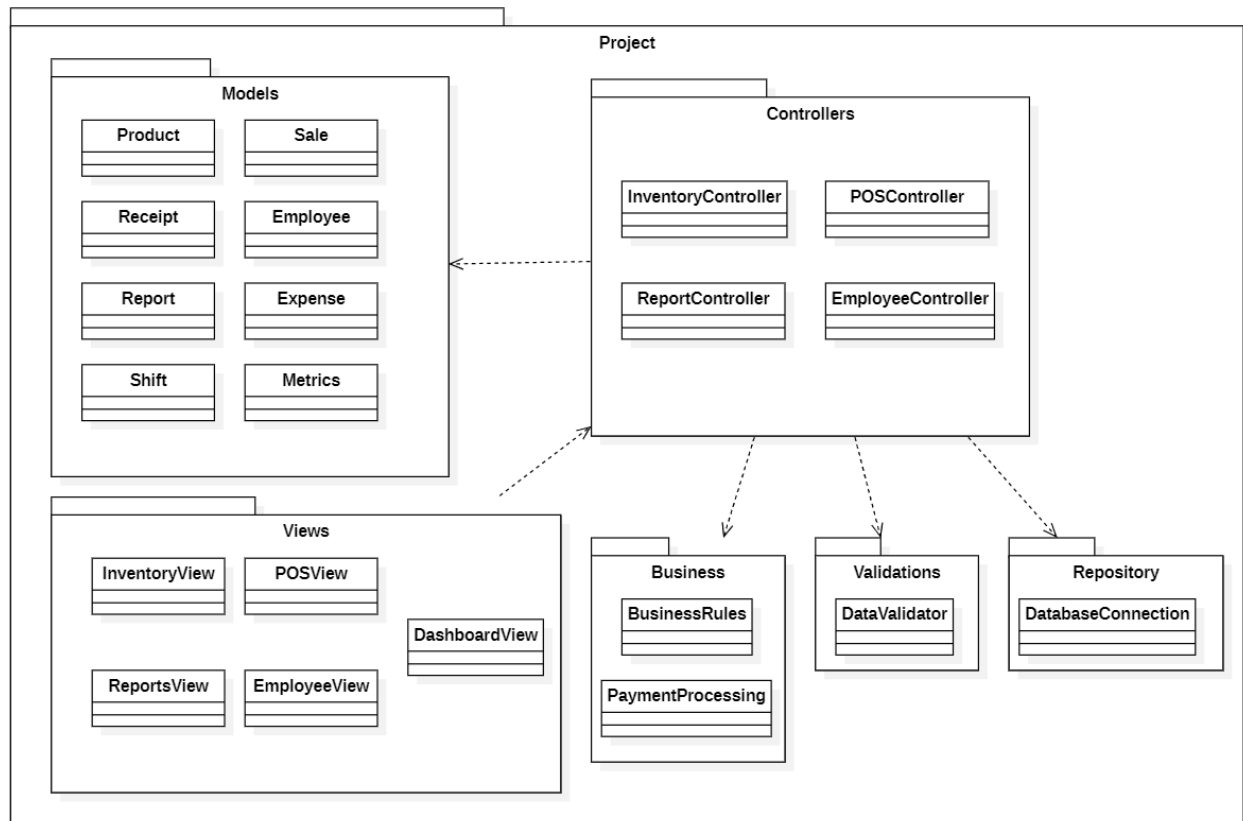
- **DashboardView:** Muestra un resumen visual de las métricas clave y el rendimiento del negocio, proporcionando una vista rápida y comprensible de la situación actual y facilitando el monitoreo continuo de la salud empresarial.

Controladores

Estas clases manejan la lógica de la interacción entre el modelo y la vista, facilitando la respuesta a acciones del usuario y el flujo de datos.

- **InventoryController:** Coordina las operaciones relacionadas con el inventario, asegurando que las actualizaciones, las alertas de stock bajo y las sugerencias de reabastecimiento sean gestionadas de manera precisa y oportuna.
- **POSController:** Controla las operaciones en el punto de venta, desde la iniciación de transacciones hasta la finalización de ventas, incluyendo el manejo de descuentos y la generación de recibos.
- **EmployeeController:** Regula las interacciones relacionadas con el personal, incluyendo el registro de horas, la asignación de turnos, y la gestión de datos personales.
- **ReportController:** Facilita la creación y personalización de informes, asegurando que los datos sean precisos y estén disponibles en formatos adecuados para análisis y revisión.
- **SecurityController:** Gestiona la autenticación y autorización de usuarios, controlando el acceso a funcionalidades del sistema basadas en roles y permisos.

Figura 2. Diagrama de Paquetes



Models

Este paquete define la estructura de datos fundamental del sistema. No solo representan entidades del negocio como productos o empleados, sino que también establecen las relaciones entre estos datos, facilitando así las operaciones en la aplicación.

Componentes:

- **Product:** Almacena datos como nombre, precio, stock y fecha de caducidad. Es esencial para operaciones de inventario y ventas.

- Sale: Central para el proceso de ventas, registra cada transacción, incluyendo detalles como fecha, productos vendidos, cantidades y precios finales.
- Receipt: Documenta el recibo emitido a los clientes tras una venta, incluyendo detalles completos de los productos comprados y los precios.
- Employee: Contiene información del personal, como identificación, roles, datos de contacto y horarios, clave para la gestión de recursos humanos.
- Report: Facilita la creación de informes basados en datos acumulados de ventas, inventario y gastos, soportando decisiones de negocio.
- Expense: Registra los gastos operacionales, proporcionando un seguimiento detallado para la contabilidad y la planificación financiera.
- Shift: Gestiona los turnos de trabajo de los empleados, asegurando que el personal esté adecuadamente asignado según las necesidades operacionales.
- Metrics: Proporciona indicadores de desempeño, esenciales para el análisis y la mejora continua de las operaciones.

Views

Este paquete maneja todas las interacciones con el usuario final, presentando datos de manera legible y recogiendo entradas del usuario para diversas operaciones. Es crucial para la experiencia del usuario y la eficiencia operativa.

Componentes:

- InventoryView: Permite a los usuarios visualizar y gestionar el stock actual, y realizar ajustes según sea necesario.
- POSView: Interfaz utilizada por el personal para ingresar ventas, aplicar descuentos y procesar pagos, fundamental en el punto de venta.
- ReportsView: Ofrece funcionalidades para seleccionar, configurar y visualizar varios tipos de informes financieros y operativos.
- EmployeeView: Utilizada por los administradores para gestionar información relacionada con los empleados, desde contrataciones hasta asignaciones de turnos.
- DashboardView: Proporciona un resumen visual de indicadores clave como ventas diarias, tendencias de inventario y desempeño general.

Controllers

Sirven como intermediarios entre las Views y el Models, gestionando la lógica de negocio necesaria para responder a las acciones del usuario y actualizar las vistas apropiadamente.

Componentes:

- InventoryController: Maneja solicitudes relacionadas con el inventario, como añadir o eliminar productos.
- POSController: Controla transacciones de ventas, desde la iniciación hasta la finalización y generación de recibos.

- ReportController: Administra la generación y recuperación de informes personalizados y predefinidos.
- EmployeeController: Se encarga de todas las operaciones relacionadas con la administración de empleados.

Business

Encapsula la lógica de negocio central, separando estas operaciones complejas de las otras preocupaciones del sistema. Esto incluye reglas de negocio, cálculos y decisiones que no son directamente parte de la presentación de datos o de la gestión de la base de datos.

Detallar los componentes específicos dependería de las necesidades del negocio, pero típicamente incluirían gestores o servicios especializados en financiamiento, operaciones, análisis de datos, etc.

Repository

Gestiona todas las interacciones con la base de datos o cualquier otro almacenamiento de datos, abstrayendo la complejidad de las operaciones de persistencia de datos.

Detallar los componentes específicos dependería de la estructura de la base de datos y los requerimientos del sistema, pero típicamente incluirían clases encargadas de manejar la persistencia de datos para las distintas entidades del modelo, proporcionando métodos

para realizar operaciones CRUD. Estas clases actuarían como una capa intermedia entre la lógica de negocio y la base de datos, asegurando que los datos sean accedidos y manipulados de manera estructurada y eficiente.

Validations

Asegura que los datos ingresados en el sistema cumplan con todos los estándares y reglas antes de ser procesados o almacenados, crucial para mantener la integridad y la validez de los datos.

Detallar los componentes específicos dependería de las reglas de negocio y los requisitos de integridad de datos del sistema, pero típicamente incluirían clases o servicios dedicados a la validación de datos de entrada, asegurando que cumplan con los formatos esperados y las restricciones definidas. Estas validaciones podrían abarcar desde comprobaciones básicas (como formatos de números y fechas) hasta reglas de negocio más complejas, como la verificación de disponibilidad de stock antes de completar una venta o la validación de permisos de empleados para ejecutar ciertas acciones.

Persistencia de Datos

Figura 3. Diagrama de Clases Persistentes

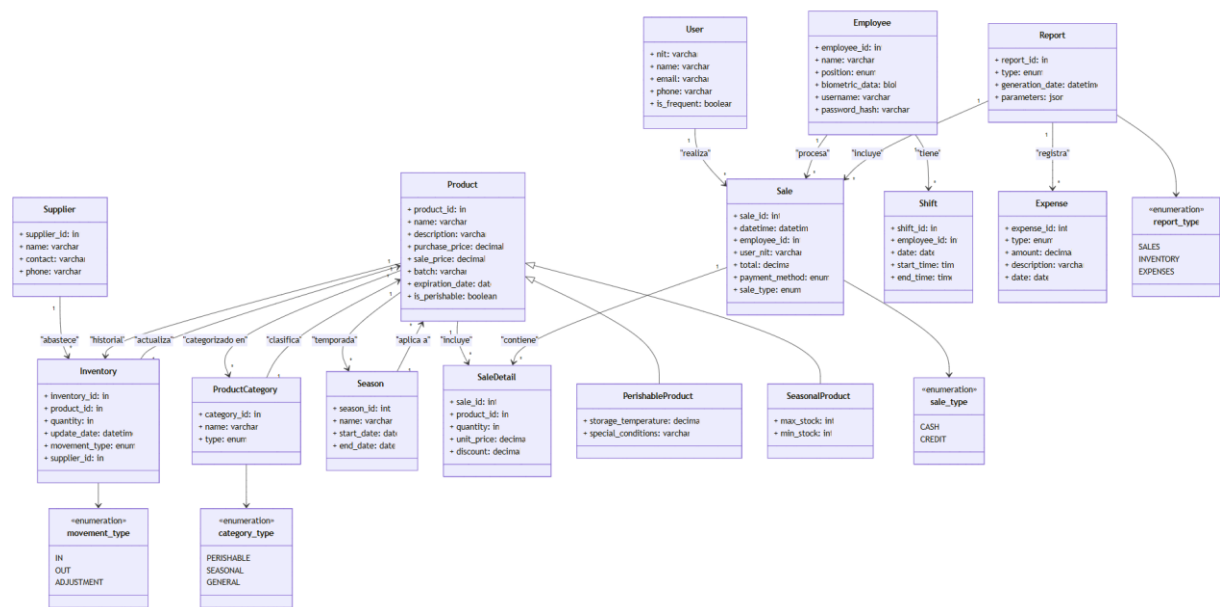
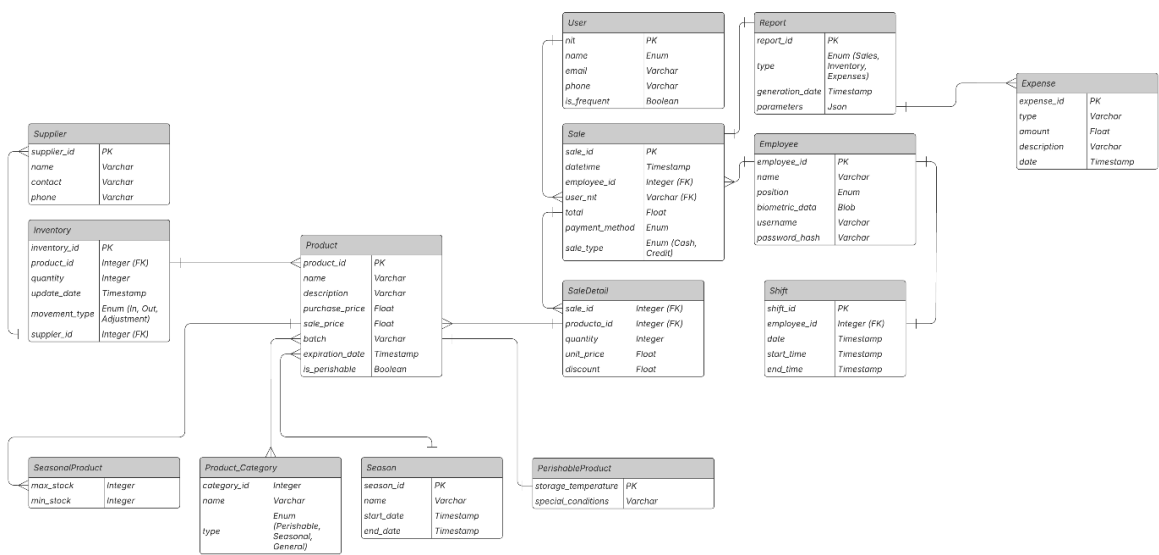


Figura 4. Diagrama Entidad-Relación



Las figuras 3 y 4 muestran la estructura fundamental para el almacenamiento y la gestión de datos, garantizando el cumplimiento de los requisitos funcionales y operativos del negocio que servirá de base para la futura implementación. Permitiendo la actualización automática del inventario y la generación de alertas sin intervención manual (RF09-RF13), facilitando el seguimiento detallado de las transacciones en Sale y los movimientos en Inventory y su estructura modular permite la integración de nuevos módulos y funcionalidades, adaptándose a las necesidades de crecimiento del negocio.

Diseño

Dimensiones del Sistema

1. Cantidad de usuarios probables que pudiera llegar a tener el sistema

- Pequeñas empresas: 5 - 50 usuarios (tiendas locales, cafeterías, restaurantes pequeños).
- Medianas empresas: 50 - 500 usuarios (franquicias, supermercados, cadenas de restaurantes).
- Grandes empresas: 500 - 5,000+ usuarios (corporaciones multinacionales, retail a gran escala).

Estimación global: El sistema debería escalar para soportar 10,000 – 50,000 usuarios activos registrados si se considera una adopción masiva en diferentes sectores.

2. Tiempo promedio de respuesta del sistema

- Operaciones críticas (punto de venta, búsqueda de productos).
 - Tiempo objetivo: menos de 2 segundos (para asegurar una experiencia fluida en la caja).
 - Pico máximo aceptable: menos de 5 segundos en situaciones de alta demanda.
- Operaciones complejas (generación de reportes, análisis financiero)
 - Tiempo objetivo: menos de 10 segundos para consultas medianas (mes de datos).

- Procesos históricos (años de registros): 30 - 60 segundos con optimización por lotes.

Meta de rendimiento: Mantener el 99% de las solicitudes en menos de 2 segundos en operaciones regulares.

3. Cantidad de usuarios promedio trabajando simultáneamente

- Pequeñas empresas: 5 - 20 usuarios concurrentes.
- Medianas empresas: 50 - 200 usuarios concurrentes.
- Grandes empresas: 500 - 5,000 usuarios concurrentes en picos.

Escenario de carga máxima: Soportar 10,000 - 25,000 conexiones simultáneas en situaciones de alto tráfico (Black Friday, fin de mes).

4. Escalabilidad del sistema

A medida que el negocio crece, el sistema debe gestionar el tamaño de datos almacenados. Considerando registros de ventas, inventario y clientes, una estimación por negocio es:

- Pequeñas empresas: 1 GB - 10 GB/año (registros básicos).
- Medianas empresas: 50 GB - 500 GB/año (información detallada de productos, ventas y empleados).
- Grandes empresas: 1 TB - 5 TB/año (alta frecuencia de transacciones, auditorías completas).

Estimación general: En 5 años, el sistema podría almacenar entre 50 GB (pequeños negocios) y 50 TB (grandes cadenas).

5. Disponibilidad de la información

- Objetivo de Disponibilidad (SLA): 99.99% (equivalente a menos de 5 minutos de inactividad al mes).
- Respaldos y replicación
 - Backups incrementales cada 5 minutos para transacciones críticas.
 - Respaldo completo al menos una vez por día.
- Acceso 24/7: Los usuarios deben acceder al sistema en todo momento, desde múltiples dispositivos.

Meta: Garantizar acceso a los datos en tiempo real, con replicación automática en múltiples zonas geográficas.

6. Confiabilidad, tolerancia a fallos masivos, replicación de la información

- Replicación de base de datos
 - Primario-Secundario (Master-Slave) para alta disponibilidad.
 - Multi-región en la nube para protegerse de fallos catastróficos.
- Estrategias de tolerancia a fallos
 - Balanceo de carga para distribuir el tráfico entre múltiples servidores.

- Monitoreo activo (alertas en caso de anomalías o caídas).
- Failover automático: Si un nodo falla, otro debe asumir en < 30 segundos.

Meta de confiabilidad:

- RTO (Recovery Time Objective): < 1 minuto en caso de fallo.
- RPO (Recovery Point Objective): < 5 minutos (pérdida máxima de datos en un incidente).

Actualización de Requisitos No Funcionales

Requisito No Funcional	Categoría	Forma en que se medirá su cumplimiento
Interfaz intuitiva y accesible	Usabilidad	Se realizará una prueba con usuarios nuevos, y al menos el 80% deben completar una tarea básica en menos de 90 segundos sin ayuda.
Capacidad de expansión	Escalabilidad	El tiempo de respuesta (Segundos) no debe aumentar más del 10% al agregar nuevos módulos.
Registro de errores y monitoreo	Mantenibilidad	Se analizarán logs de errores durante 30 días y al menos el 98% de los fallos críticos deben registrarse y notificarse.

Modularidad del sistema	Mantenibilidad	Se medirá la cantidad de errores derivados de actualizaciones. Menos del 5% de los cambios en un módulo deben generar fallos en otros módulos no relacionados.
Cifrado de datos sensibles	Usabilidad	Se realizarán 100 intentos de acceso no autorizado simulados y el 100% deben ser bloqueados.
Soporte técnico y documentación	Soporte	Se registrarán los tiempos de respuesta a tickets de soporte. El 90% deben resolverse en menos de 12 horas.
Respaldo y recuperación de datos	Confiabilidad	Se ejecutarán pruebas de recuperación de datos y el sistema debe restaurar la información al 100% haciendo uso de backups.
Cumplimiento de normativas legales (Facturación)	Legales/Estándar	Se auditarán 200 facturas al mes y el 100% deben cumplir con normativas fiscales.
Escalabilidad de usuarios concurrentes	Escalabilidad	El sistema debe soportar hasta 10 usuarios concurrentes sin afectar el rendimiento.

Disponibilidad del sistema	Disponibilidad	El sistema debe estar disponible el 99.9% del tiempo mensualmente.
----------------------------	----------------	--

Investigación de Tecnologías (frameworks, patrones, lenguaje, etc.)

Cuadro 1. Cuadro comparativo de tecnologías frontend

Tecnología	Lenguaje	Ventajas	Desventajas
React.js	JavaScript	<ul style="list-style-type: none"> - Amplio ecosistema de librerías y herramientas. - Flexible y adaptable a diferentes arquitecturas. - Soporte de renderizado del lado del servidor (SSR). 	<ul style="list-style-type: none"> - JSX y manejo de estados que requieren un mayor dominio. - Puede ser complejo para principiantes. - Actualizaciones pueden generar problemas de compatibilidad

Vue.js	JavaScript	<ul style="list-style-type: none"> - Fácil de aprender. - Sintaxis más simple que React. - Reactividad incorporada facilita la gestión de estados simples 	<ul style="list-style-type: none"> - Menos opciones de librerías y herramientas en comparación con React. - Puede ser menos adecuado para aplicaciones muy grandes y complejas.
Angular	TypeScript	<ul style="list-style-type: none"> - Arquitectura bien definida y escalable. - Soporte oficial de Google. - Buen manejo de grandes aplicaciones. 	<ul style="list-style-type: none"> - Mayor rigidez en la estructura puede limitar la flexibilidad. - Tamaño del bundle puede ser mayor en comparación con otros frameworks. - Proceso de actualización puede ser complejo.

Cuadro 2. Cuadro comparativo de tecnologías para backend

Tecnología	Lenguaje	Ventajas	Desventajas
Spring Boot	Java	<ul style="list-style-type: none"> - Gran ecosistema y comunidad amplia. - Soporte nativo para multi-tenant con Hibernate. - Seguridad robusta con Spring Security. - Escalabilidad horizontal con microservicios. 	<ul style="list-style-type: none"> - Curva de aprendizaje pronunciada. - Configuración compleja para aplicaciones grandes. - Más consumo de recursos que opciones más livianas.
Django + Django Tenants	Python	<ul style="list-style-type: none"> - Rápido desarrollo con enfoque DRY. - Módulo "Django Tenants" para manejar bases de datos compartidas o separadas. - Admin integrado para gestión de inquilinos. 	<ul style="list-style-type: none"> - Menos eficiente para cargas extremadamente altas. - No es ideal para arquitecturas de microservicios complejos.

Node.js + NestJS	JavaScript/TypeScript	<ul style="list-style-type: none"> - Rendimiento asíncrono para alta concurrencia. - NestJS ofrece estructura modular similar a Spring. - Flexible para distintos patrones multi-tenant (DB compartida o aislada). 	<ul style="list-style-type: none"> - Mayor complejidad al gestionar grandes bases de datos. - Más susceptible a errores de ejecución por su naturaleza asíncrona.
Laravel	PHP	<ul style="list-style-type: none"> - Framework experimentado y bien documentado. - Soporte para multi-tenancy con paquetes como "Tenancy for Laravel". - Fácil despliegue en infraestructuras tradicionales. 	<ul style="list-style-type: none"> - Rendimiento comparativamente más lento que otros Frameworks. - Dependencia de demasiadas librerías externas.

Ruby on Rails + Apartment	Ruby	<ul style="list-style-type: none"> - Rápido desarrollo con convenciones claras. - Gem "Apartment" para manejar multi-tenancy. - Comunidad sólida y bien documentada. 	<ul style="list-style-type: none"> - Rendimiento menor que opciones como Java o C#. - Dificultad para escalar horizontalmente.
---------------------------	------	---	--

Para el desarrollo del proyecto se buscaron diversas tecnologías con el objetivo de garantizar que estas garantizaran escalabilidad, eficiencia, y seguridad para los negocios a los que el software este dirigido. Como grupo luego de una profunda investigación se optó por utilizar React como biblioteca para el desarrollo del frontend, Node.js como Framework para el Backend y de patrón de diseño MVC (Modelo-Vista-Controlador). La elección de estas tecnologías se fundamenta en la necesidad de construir una plataforma moderna e intuitiva que centralice y mejore la gestión de ventas, inventario y facturación, optimizando así las operaciones de los negocios usuarios.

Tecnología Seleccionada para Frontend: React

React es una biblioteca de JavaScript, desarrollada por Facebook (Meta) con el objetivo de la creación de interfaces de usuario interactivas y de alto rendimiento. Esta biblioteca

tiene como enfoque la creación de componentes reutilizables e independientes ya que estos facilitan la organización y mantenimiento del código, lo que permite una arquitectura modular y escalable del programa. Asimismo, su fácil integración con otras tecnologías y frameworks de backend hacen que React ofrezca una gran cantidad de flexibilidad en la arquitectura del programa, y esta cuenta con una gran comunidad de desarrolladores, documentación muy completa y disponibilidad de librerías externas, facilitando la resolución de problemas y extensión de funcionalidades.

Sin embargo, uno de los retos que se tienen al utilizar esta tecnología como Frontend es la curva de aprendizaje pronunciada que tiene esta, ya que es recomendable tener conocimientos previos de JSX (JavaScript XML) y el manejo de estados debido a que si no se implementan los patrones adecuados esta última puede llegar a volverse demasiado compleja ocasionando que se consuma más tiempo del necesario.

Tecnología Seleccionada para Backend: Node.js

Este Framework es un entorno de ejecución para JavaScript que permite el desarrollo de aplicaciones eficientes y escalables, más específicamente en sistemas basados en eventos y en tiempo real ofreciendo un alto rendimiento, ideal para aplicaciones con muchas conexiones concurrentes y es utilizada normalmente para la creación de APIs RESTful.

Otro punto a favor de este framework es el lenguaje de programación, ya que nuestro Frontend también es basado en JavaScript por lo que solo se utilizaría un solo lenguaje de código facilitando el desarrollo y la colaboración entre los integrantes del grupo.

Pero no todo es bueno, ya que si se busca el usar aplicaciones que requieran un procesamiento intensivo de la CPU este framework no sería la mejor opción debido a su naturaleza de un solo hilo, a su vez que se requieren de herramientas adicionales para un manejo correcto y eficiente de bases de datos relacionales.

Patrón de Diseño Seleccionado: Modelo-Vista-Controlador (MVC)

Este patrón de diseño organiza el código en tres componentes principales siendo estos: el modelo donde se encuentra todos los datos y la lógica del negocio, la vista donde iría toda la interfaz de usuario es decir lo que vera el usuario y el controlador el cual hará gestión de las interacciones entre el modelo y la vista. Este patrón facilita el mantenimiento del código al separar las diferentes capas que posee la aplicación, permitiéndonos trabajar en las distintas capas de la aplicación de forma independiente y permitiendo la adición y modificación de funcionalidades nuevas o ya creadas.

Sin embargo, este patrón de diseño puede generar contratiempos si no es planificado correctamente desde las etapas iniciales ya que si no se define correctamente la arquitectura puede llegar a provocar complejidades innecesarias en el proyecto.

Investigación de Bases de Datos

Para este proyecto de gestión de operaciones para el cual se trata de solucionar la falta de integración entre los sistemas de ventas, inventario y administración, tanto en empresas pequeñas como en empresas grandes, se debe de contar con una base de datos que sea segura, escalable y eficiente al momento de manejar los datos de las ventas, el inventario y el área de administración. Por lo que algunas de las tecnologías recomendadas son:

Bases de Datos Relacionales (SQL)

Es la manera de almacenar y organizar datos con relaciones definidas para tener un acceso rápido, los datos son organizados en tablas que contienen la información de las entidades y las categorías por medio de filas y columnas, esta estructura sirve para que el acceso a los datos sea más eficiente y flexible y se reduce la redundancia de datos. Las bases de datos utilizan cuatro criterios principales, atomicidad para, coherencia, aislamiento y durabilidad (ACID).

PostgreSQL

Es un sistema de base de datos que utiliza código abierto y lenguaje SQL, cuenta con varias funciones de almacenamientos y escalabilidad para el trabajo de datos más complejos. Cuenta con buena reputación por la arquitectura que utiliza que es la de

cliente – servidor, su fiabilidad, la integridad de datos y la robustez de funciones. Otra de sus cualidades es que se encuentra disponible para diferentes sistemas operativos como Linux, Windows, macOS e incluso BSD y solaris y hace uso de ACID desde el 2001. Es útil para los desarrolladores por la integridad de los datos y el entorno tolerante a fallos, también que es un sistema gratuito y a la gestión de los datos de manera independiente al tamaño y permite escribir código en diferentes lenguajes de programación sin tener que recompilar la base de datos.

- Ventajas

- Tiene un soporte avanzado para transacciones, cumpliendo ACID.
- Soporta grandes cantidades de datos sin comprometer el rendimiento lo que lo hace escalable.
- Permite el agregar funciones personalizadas y tiene soporte para JSON, XML y arrays.
- Proporciona seguridad por autenticación basada en roles y encriptación de los datos.
- Ofrece un buen rendimiento en consultas complejas.
- Es de código abierto y no tiene costos de licencia.

- Desventajas

- Se necesita un mayor conocimiento en comparación con MySQL.
- Consume más RAM y CPU que MySQL y MariaDB.

MySQL

Con este sistema se puede impulsar las aplicaciones web de comercio electrónico y procesamiento de transacciones de línea (OLTP) por lo que la hace segura para transacciones y hace uso con ACID, tiene funciones de confirmación, reversión, recuperación por fallos y bloqueo por nivel de fila. MySQL es de uso fácil, seguro, escalable y tiene el rendimiento necesario para impulsar aplicaciones web o móviles. Este sistema también ofrece un diccionario de los datos transaccionales, consultas a las tablas, reduce la complejidad del código, permite otorgar y denegar permisos a los diferentes usuarios entre otras funcionalidades más.

- Ventajas
 - Tiene buen rendimiento en consultas de lectura.
 - Es fácil de aprender y utilizar y tiene una amplia documentación.
 - Es compatible con aplicaciones web.
 - Tiene soporte de ACID.
- Desventajas
 - Es menos potente en transacciones complejas que otros sistemas como PostgreSQL o SQL Server.
 - No tiene mucho soporte para consultas analíticas.

MariaDB

Este sistema también es de código abierto, algunos de sus usuarios son Wikipedia, WordPress.com y Google, se puede utilizar para datos transaccionales de alta disponibilidad y para el análisis de datos. Tiene compatibilidad con MySQL por lo que las aplicaciones que hagan uso de MySQL no tendrán problemas con este sistema, pero se desvía del objetivo de MySQL. MariaDB tiene la capacidad de poder dividir las cargas de las bases de datos en diferentes servidores y optimizar el escalado, este sistema se encuentra en varios sistemas operativos como Linux, Windows y macOS.

- Ventajas
 - Es rápido en diferentes operaciones que MySQL y la optimización de almacenamiento.
 - Cuenta con una mejor escalabilidad.
 - Es de código abierto.
- Desventajas
 - No cuenta con mucho soporte empresarial como SQL Server o PostgreSQL.
 - Puede llegar a tener problemas de compatibilidad.

SQL Server

Es un conjunto de tablas en las cuales se almacenan los datos estructurados. Cuenta con una interfaz de usuario gráfica para poder crear las bases de datos y un editor de consultas para poder interactuar con los datos. Con este sistema se puede almacenar y recuperar los datos, también facilita funciones como la inteligencia empresarial, procesamiento de transacciones y el análisis.

- Ventajas
 - Cuenta con buen soporte técnico y actualizaciones constantes.
 - Tiene seguridad con cifrado de datos y administración de usuarios.
 - Tiene buen rendimiento en sistemas transaccionales y analíticos.
- Desventajas
 - La licencia tiene un costo y este es mayor dependiendo de la versión del sistema.
 - Consume más recursos que MySQL y MariaDB.
 - No están flexible en código abierto como PostgreSQL.

Tecnología Seleccionada para la Base de Datos (PostgreSQL)

Para la selección de la tecnología para la base de datos se tomaron en cuenta varios gestores, con el fin de poder seleccionar una tecnología que proporcione seguridad, escalabilidad y eficiencia al momento de administrar los datos. Los gestores que se tuvieron en consideración son: PostgreSQL, MySQL, MariaDB, SQL Server, de cada uno

de ellos se tomó en cuenta las funcionalidades de cada gestor y sus ventajas y desventajas. Después de tomar todos los aspectos en consideración se decidió utilizar PostgreSQL por las siguientes razones:

- Permite el manejo de grandes volúmenes de datos sin afectar el rendimiento y esto es importante para un sistema de ventas, inventario y administración de datos.
- Proporciona seguridad a los datos permitiendo la gestión de roles y permisos, también por medio del cifrado de datos, lo que protege la información del negocio.
- Es compatible con JSON por lo que permite almacenar y gestionar datos semiestructurados lo que facilita la integración de nuevas funcionalidades.
- Al ser de código abierto la licencia no cuenta con un costo sin comprometer la funcionalidad y el rendimiento de este mismo.
- Proporciona herramientas para el análisis de datos, optimización de consultas y recopilación de datos, lo que es importante para poder generar reportes financieros.

Referencias

Microsoft. (2025). *¿Qué es un sistema de administración de bases de datos relacionales?*. <https://azure.microsoft.com/es-mx/resources/cloud-computing-dictionary/what-is-a-relational-database#whatis>

PostgreSQL. (s. f.). *The PostgreSQL Global Development Group*.
<https://www.postgresql.org/about/>

MySQL Database. (s. f.). <https://www.mysql.com/products/enterprise/database/>

MariaDB. (19 de junio de 2023). *MariaDB in brief*. <https://mariadb.org/en/#scaling-partitioning>

Branchek, B., & Branchek, B. (26 de junio de 2024). What is Microsoft SQL Server used for? - ONLC. ONLC. <https://www-onlc-com.translate.goog/blog/what-is-microsoft-sql-server-used-for/? x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=rq>

Aiemwp. (2024, 16 enero). *¿Qué es React y sus características?* - Devits. Devits.
<https://devits.io/blog/que-es-react/>

De Imagina, E. (2025, 17 marzo). *Qué es Node.js y Por qué Debes Usarlo*. Imagina Formación. <https://imaginaformacion.com/tutoriales/que-es-nodejs-y-por-que-debes-usarlo>

Hernandez, R. D. (2021, 28 junio). *El patrón modelo-vista-controlador: Arquitectura y frameworks explicados*. freeCodeCamp.org.

<https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern>

Desarrolladoresweb.org. (2024, 30 noviembre). ▷*Vue.js: Una guía completa y detallada*. Web Devs. <https://desarrolladoresweb.org/vue-js/vue-js-una-guia-completa-y-detallada/>

Angular. (s. f.). Angular. <https://angular.dev/overview>

GeeksforGeeks. (2025, 2 enero). *Spring Boot Tutorial*. GeeksforGeeks. <https://www.geeksforgeeks.org/spring-boot/>

W3Schools.com. (s. f.). https://www.w3schools.com/django/django_intro.php

Informe de Gestión

El presente informe tiene como objetivo detallar la gestión del tiempo realizada por el equipo de desarrollo durante el proceso de investigación y ejecución de tareas asignadas, analizando tanto los aspectos positivos como los aspectos a mejorar en el desempeño del grupo.

Cronograma de Actividades

Tarea	Inicio	Fin	Comentarios
Diagrama de Clases	16-03-2025 19:00 hrs.	16-03-2025 23:00 hrs.	Pausa de 30 minutos para actividades diarias.
Diagrama de Paquetes	17-03-2025 06:00 hrs.	17-03-2025 08:00 hrs.	Pausa de 15 minutos para actividades diarias.
Diagrama de Clases Persistentes	18-03-2025 17:00 hrs.	18-03-2025 20:00 hrs.	Pausa de 30 minutos para despejar la mente.
Diagrama de Entidad Relación	17-03-2025 17:00 hrs.	17-03-2025 19:00 hrs.	Pausa de 25 minutos para despejar la mente.
Investigación de Tecnologías	16-03-2025 18:43 hrs.	17-03-2025 16:00 hrs.	Pausa de 2 horas para actividades diarias.

Investigación de Bases de Datos	16-03-2025 17:00 hrs.	16-03-2025 23:30 hrs.	Pausa de 2 horas para actividades diarias.
Presentación	17-03-2025 17:30 hrs.	19-03-2025 15:00 hrs.	Pausa de para actividades diarias o descanso.
Informe de gestión	19-03-2025 23:00 hrs.	19-03-2025 23:50 hrs.	Revisión detallada de tiempo. Trabajo dependiente de otras tareas.

Desglose de Tareas por Miembro

Tarea	Miembro
Diagrama de Clases	Javier Linares
Diagrama de Paquetes	Javier Linares
Diagrama de Persistencia	Daniel Chet
Diagrama de Entidad Relación	Dulce Ambrosio
Requisitos Funcionales	Cristian Túnchez
Requisitos No Funcionales	Cristian Túnchez
Investigación de Tecnologías	Daniel Chet
Investigación de Bases de Datos	Dulce Ambrosio
Presentación	Todos
LOGT	Todos
Informe de Gestión de Tiempo	Cristian Túnchez

Conclusiones (de la gestión de tiempo)

El equipo demostró un alto nivel de compromiso y colaboración, lo que se reflejó en la calidad del trabajo entregado. La comunicación constante y la revisión periódica del progreso permitieron que el equipo se mantuviera alineado con los objetivos del proyecto. Además, la utilización de herramientas como Word facilitó la colaboración y el seguimiento de los cambios realizados.

Aunque el equipo logró cumplir con los plazos establecidos, se identificó que en algunas ocasiones hubo retrasos debido a la falta de claridad en los requisitos iniciales. Esto llevó a la necesidad de realizar ajustes en etapas avanzadas del proyecto, lo que incrementó el tiempo de desarrollo. Para futuros proyectos, se recomienda dedicar más tiempo a la fase de análisis inicial para asegurar que todos los requisitos estén claramente definidos desde el principio.

En general, la gestión del tiempo fue efectiva y permitió al equipo cumplir con los objetivos del proyecto dentro del plazo establecido. El uso del formulario LOGT fue una herramienta valiosa para mantener un registro preciso del tiempo invertido y para identificar áreas de mejora.

Anexos

CRISTIAN TÚNCHEZ (231359) LOGT

FECHA	INICIO	FIN	TIEMPO INTERRUPCIÓN	DELTA TIEMPO	FASE	COMENTARIOS
16/03/2025	18:01	18:57	00:00:00	00:56:00	Requisitos Funcionales	Sin comentarios.
16/03/2025	22:48	23:50	00:00:00	01:02:00	Dimensiones del Sistema	Sin comentarios.
17/03/2025	17:30	19:00	00:00:00	01:30:00	Presentación	Sin comentarios.
19/03/2025	23:00	23:50	00:05:00	00:45:00	Informe de Gestión	Se tuvo que esperar a que se concluyeran detalles de algunas tareas.

DANIEL CHET (231177) LOGT

FECHA	INICIO	FIN	TIEMPO INTERRUPCIÓN	DELTA TIEMPO	FASE	COMENTARIOS
16/03/2025	18:43	23:11	02:00:00	02:28:00	Tecnologías de Frontend escogida	Se trabajo en la descripcion y justificacion de la tecnoloa escogida del Frontend y se tuvo un descando para cenar
17/03/2025	03:00	4:00	00:00:00	01:00:00	Tecnologías de Backend escogida	Se trabajo en la descripcion y justificacion de la tecnoloa escogida del Backend
18/03/2025	17:00	20:00	00:30:00	02:30:00	Diagrama de Persistencia	Se realizo el diagrama de Presistencia del proyecto y se tuvo una pausa para descansar
19/03/2025	12:00	15:00	00:30:00	01:30:00	Presentacion	Se trabajo en la parte de tecnologías de Frontend en la presentacion del corte 3 y se tuvo una pausa para descansar

DULCE AMBROSIO (231143) LOGT

FECHA	INICIO	FIN	TIEMPO INTERRUPCIÓN	DELTA TIEMPO	FASE	COMENTARIOS
16/03/2025	5:00	11:30	02:30:00	04:00:00	Tecnología Escogida	Se trabajo en la descripción y justificación de la tecnología escogida para la base de datos y se tomo un tiempo para descansar y cenar.
17/03/2025	5:00	7:00	00:25:00	01:35:00	Diagrama ER	Se realizo el diagrama ER y se tomó un descanso.
18/03/2025	3:00	4:00	00:00:00	00:01:00	Presentación	Se trabajo en la presentación.

JAVIER LINARES (231135) LOGT

FECHA	INICIO	FIN	TIEMPO INTERRUPCIÓN	DELTA TIEMPO	FASE	COMENTARIOS
16/03/2025	19:00	22:00	00:30:00	02:30:00	Diagrama de Clases	Hubo una pausa de 30 min para cenar.
16/03/2025	22:05	23:00	00:00:00	00:55:00	Descripción de Clases	No hubo ninguna interrupción durante la realización de la tarea.
17/03/2025	6:00	7:30	00:15:00	01:15:00	Diagrama de Paquetes	Pausa de 15 min para ir al baño.
17/03/2025	7:30	8:00	00:00:00	01:30:00	Descripción de cada paquete y componentes	No hubo ninguna interrupción durante la realización de la tarea.
17/03/2025	17:30	18:50	00:00:00	01:20:00	Presentación	No hubo ninguna interrupción durante la realización de la tarea.
18/03/2025	21:30	23:00	00:35:00	01:30:00	Correcciones sugeridas por el catredatico	Hubo una pausa de 35 mins para descansar.

Enlaces Adicionales

Documento en Línea (Word)



Repositorio del Proyecto (GitHub)

