



**Universidad
Tecnológica
del Perú**

Facultad de Ingeniería

Carreras de Ingeniería de Redes, Seguridad y Auditoría Informática, Sistemas y Software

Sistema de Ventas

Integrantes

Calla Quispe, Jimmy Roy

Hernández Mendoza, Walter Manuel

Montes Zarria, Martín André

Rivera Berrios, José Alonso

Curso: Programación Orientada a Objetos

Docente: Waldyr Fredy Cerrón

Lima - Perú

2023

Índice

<i>Aspectos Generales</i>	3
1.1. Definición del problema.....	3
1.1.1. Descripción del problema.....	3
1.2. Definición de objetivos.....	3
1.2.1. Objetivo general.....	3
1.2.2. Objetivos específicos.....	4
1.2.3. Alcances y limitaciones.....	4
Marco Teórico.....	6
2.1. Antecedentes.....	6
2.2. Fundamento teórico.....	6
Desarrollo de la Solución.....	7
3.1. Lista de requerimientos funcionales.....	7
3.2. Lista de requerimientos no funcionales.....	8
3.3. Casos de uso.....	9
3.3.1. Diagrama Casos de uso.....	9
3.3.2 Matriz de casos de uso y requerimientos.....	9
3.4. Especificación de los casos de uso.....	10
3.5. Diagrama de clases.....	16
3.6. Presentación del prototipo.....	16
3.7. Códigos de los formularios y pantallas de ejecución del programa informático.....	17
Conclusiones.....	20

Aspectos Generales

1.1. Definición del problema

1.1.1. Descripción del problema

Estar en desventaja frente a la competencia que sí usa tecnología para optimizar sus procesos puede llevar a perder clientes por la lentitud y la mala experiencia de compra. Además, puede haber dificultades para obtener información sobre el inventario, las ventas, los proveedores, etc. Esto puede resultar en pérdida de tiempo y dinero en tareas manuales y propensas a errores, y también puede impedir aprovechar las oportunidades de venta online y el comercio local. Sin embargo, también es importante tener en cuenta que el uso de tecnología en los negocios también puede tener desventajas, como los costos de implementación y actualización, la dependencia de la tecnología, el desplazamiento del factor humano en las organizaciones y la limitación en las relaciones interpersonales.

1.2. Definición de objetivos

1.2.1. Objetivo general

Aumentar las ventas: Un sistema de ventas puede ayudar a aumentar las ventas al mejorar la eficiencia en los procesos de venta y al proporcionar información valiosa sobre el comportamiento de los clientes y las tendencias del mercado.

Mejorar la eficiencia en los procesos de venta: Un sistema de ventas puede ayudar a mejorar la eficiencia en los procesos de venta al automatizar tareas repetitivas y al proporcionar herramientas para el seguimiento y la gestión de las ventas.

Optimizar los recursos: Un sistema de ventas puede ayudar a optimizar los recursos al proporcionar información valiosa sobre el inventario, las ventas y los proveedores, lo que permite tomar decisiones informadas sobre la gestión de los recursos.

1.2.2. Objetivos específicos

Mejorar el seguimiento y la gestión del inventario: Un sistema de ventas puede ayudar a mejorar el seguimiento y la gestión del inventario al proporcionar información actualizada sobre el stock de productos y al permitir la gestión eficiente de las órdenes de compra y las devoluciones.

Mejorar la experiencia del cliente: Un sistema de ventas puede ayudar a mejorar la experiencia del cliente al permitir un proceso de compra más rápido y eficiente y al proporcionar herramientas para el seguimiento y la gestión de las interacciones con los clientes.

Aumentar las ventas: Un sistema de ventas puede ayudar a aumentar las ventas al proporcionar información valiosa sobre el comportamiento de los clientes y las tendencias del mercado, lo que permite diseñar estrategias de venta efectivas.

Mejorar la toma de decisiones: Un sistema de ventas puede ayudar a mejorar la toma de decisiones al proporcionar información valiosa sobre el rendimiento de las ventas, el inventario y los proveedores, lo que permite tomar decisiones informadas sobre la gestión del negocio.

Reducir costos y aumentar la eficiencia: Un sistema de ventas puede ayudar a reducir costos y aumentar la eficiencia al automatizar tareas repetitivas y al proporcionar herramientas para la gestión eficiente de los procesos de venta.

1.2.3. Alcances y limitaciones

Alcances:

- El programa puede vender productos, agregar y eliminar proveedores, aumentar y disminuir la cantidad de productos y generar boletas y facturas.
- El programa está diseñado para ser utilizado por una tienda de ropa específica y está optimizado para funcionar en un entorno de venta al por menor.
- El programa puede manejar una cantidad significativa de datos y transacciones y puede proporcionar información valiosa sobre el comportamiento de los clientes y las tendencias del mercado.

Limitaciones:

- El programa puede tener una capacidad limitada para manejar grandes cantidades de datos y transacciones.
- El programa requiere personal capacitado para utilizarlo y mantenerlo actualizado.
- El programa depende de una conexión a Internet estable para acceder a la base de datos.
- Puede haber limitaciones en cuanto a la capacidad del programa para integrarse con otros sistemas o aplicaciones utilizadas por la tienda de ropa. Por ejemplo, el programa puede no ser compatible con otros sistemas de gestión de inventario o contabilidad utilizados por la tienda, lo que puede dificultar la integración de datos y la automatización de procesos.

Marco Teórico

2.1. Antecedentes

Un estudio podría haber encontrado que el uso de un sistema de ventas sofisticado puede aumentar la productividad de los representantes de ventas y maximizar los ingresos al proporcionar herramientas para el seguimiento y la gestión de las ventas, automatizar tareas repetitivas y proporcionar información valiosa sobre el comportamiento de los clientes y las tendencias del mercado.

2.2. Fundamento teórico

Se podría discutir cómo un sistema de ventas conectado a una base de datos puede mejorar el seguimiento y la gestión del inventario al proporcionar información actualizada sobre el stock de productos y al permitir la gestión eficiente de las órdenes de compra y las devoluciones. También se podría discutir cómo un sistema de ventas puede mejorar la experiencia del cliente al permitir un proceso de compra más rápido y eficiente y al proporcionar herramientas para el seguimiento y la gestión de las interacciones con los clientes. Además, se podría discutir cómo un sistema de ventas puede ayudar a aumentar las ventas al proporcionar información valiosa sobre el comportamiento de los clientes y las tendencias del mercado, lo que permite diseñar estrategias de venta efectivas.

Desarrollo de la Solución

3.1. Lista de requerimientos funcionales

El programa permitirá al usuario cargar la base de datos con un usuario ya existente en la misma. De lo contrario, no podrá acceder al programa

El programa debe tener acceso a los siguientes elementos: empleados, productos, proveedores, clientes (personas o empresas), ventas (boletas o facturas) y locales.

El programa permitirá al usuario cargar la base de datos con un usuario ya existente en la misma. De lo contrario, no podrá acceder al programa

El programa debe permitir agregar un nuevo elemento a una tabla.

El programa debe permitir editar un elemento existente de una tabla.

El programa debe permitir eliminar un elemento de una tabla.

El programa debe permitir ver con detalle un elemento de una tabla.

El programa no debe permitir la alteración del código primario de cualquier elemento tras su creación.

El programa debe autogenerar el código primario para los locales, proveedores, boletas, facturas y productos.

El programa debe simular una compra al agregar una nueva boleta o factura.

El programa debe calcular el subtotal, impuesto, descuento e importe final de cada compra automáticamente.

El programa considera si es una empresa en una factura o una persona en una boleta para calcular el impuesto y el descuento.

El programa calcula el descuento basado en la membresía de la persona (si es una boleta) y el descuento del proveedor del producto.

El programa permite añadir nuevos productos y cambiar su cantidad mientras se está creando la venta.

El programa solo permite agregar los productos que se encuentran en el local seleccionado y ser atendidos por empleados que también están asignados a dicho local.

El programa permite añadir un comentario al final de la venta que puede editarse después.

El programa no permite la alteración de los demás campos de la venta tras su creación, solo su visualización.

3.2. Lista de requerimientos no funcionales

El programa no debe tener datos sensibles sobre la cuenta de base de datos del usuario codificados estáticamente para mantener la seguridad.
El programa debe tener una interfaz con controles intuitivos y fáciles de usar.
El programa debe manejar de forma eficiente un gran número de ventas (boletas y facturas) y productos para garantizar tiempos de respuesta rápidos.
El programa debe permitir la visualización de los datos en una tabla tras su edición para confirmar que se guardó el cambio.
El programa debe mantener los datos en la base de datos y en la memoria local sincronizados y no permitir incongruencias.
El programa debe advertir al usuario antes de eliminar un elemento de cualquier tabla.
El programa debe mantener la integridad de los datos al no permitir entradas imposibles a la hora de crear elementos, como un DNI con letras o un número de 3 dígitos.
Las clases, frames y paneles del programa deben estar organizados para facilidad de acceso.

3.3. Casos de uso

3.3.1. Diagrama Casos de uso



3.3.2 Matriz de casos de uso y requerimientos

Casos de Uso	Req. funcional	Req. No funcional
Logearse al programa	El programa permitirá al usuario cargar la base de datos con un usuario ya existente en la misma. De lo contrario, no podrá acceder al programa.	El programa no debe tener datos sensibles sobre la cuenta de base de datos del usuario codificados estáticamente para mantener la seguridad.
Interactuar con las tablas del programa	<ul style="list-style-type: none"> - El programa debe tener acceso a los siguientes elementos: empleados, productos, proveedores, clientes (personas o empresas), ventas (boletas o facturas) y locales. - El programa debe permitir las funciones de agregar, editar, eliminar y ver con detalle cada elemento. 	<ul style="list-style-type: none"> - El programa debe tener una interfaz con controles intuitivos y fáciles de usar. - El programa debe manejar de forma eficiente un gran número de ventas (boletas y facturas) y productos para garantizar tiempos de respuesta rápidos. - El programa debe mantener los datos en la base de datos y en la memoria local sincronizados y no permitir incongruencias. - Las clases, frames y paneles del programa deben estar organizados para

		facilidad de acceso.
Agregar un elemento	<ul style="list-style-type: none"> - El programa debe permitir agregar un elemento a una tabla. - El programa debe autogenerar el código primario para los locales, proveedores, boletas, facturas y productos. 	El programa debe mantener la integridad de los datos al no permitir entradas imposibles a la hora de crear elementos, como un DNI con letras o un número de 3 dígitos.
Editar un elemento	<ul style="list-style-type: none"> - El programa debe permitir editar un elemento existente de una tabla. - El programa no debe permitir la alteración del código primario de cualquier elemento tras su creación. 	El programa debe mantener la integridad de los datos al no permitir entradas imposibles a la hora de crear elementos, como un DNI con letras o un número de 3 dígitos.
Eliminar un elemento	El programa debe permitir eliminar un elemento de una tabla.	<ul style="list-style-type: none"> - El programa debe permitir la visualización de los datos en una tabla tras su edición para confirmar que se guardó el cambio. - El programa debe advertir al usuario antes de eliminar un elemento de cualquier tabla.
Ver detalles de un elemento	El programa debe permitir ver con detalle un elemento de una tabla.	
Guardar los cambios de una tabla	El programa debe ser persistente al guardar todos los cambios sobre cualquier elemento en una base de datos.	<ul style="list-style-type: none"> - El programa debe permitir la visualización de los datos en una tabla tras su edición para confirmar que se guardó el cambio. - El programa debe mantener los datos en la base de datos y en la memoria local sincronizados y no permitir incongruencias.
Agregar una venta (boleta o factura)	<ul style="list-style-type: none"> - El programa debe simular una compra al agregar una nueva boleta o factura. - El programa debe calcular el subtotal, impuesto, descuento e importe final de cada compra automáticamente. - El programa considera si es una empresa en una factura o una persona en una boleta para calcular el impuesto y el descuento. - El programa calcula el descuento basado en la membresía de la persona (si es una boleta) y el descuento del proveedor del producto. 	
Agregar ropa a la venta	<ul style="list-style-type: none"> - El programa permite añadir nuevos productos y cambiar su cantidad mientras se está creando la venta. - El programa solo permite agregar los productos que se encuentran en el local seleccionado y ser atendidos por empleados que también están asignados a dicho local 	
Comentar sobre la venta	El programa permite añadir un comentario al final de la venta que puede editarse después.	
Editar y ver detalles de la venta	El programa no permite la alteración de los demás campos de la venta tras su creación, solo su visualización.	

3.4. Especificación de los casos de uso

Caso de uso: Logearse al programa

Descripción: Permite al usuario cargar la base de datos del programa con un usuario existente. De lo contrario, no podrá logearse.

Eventos:

- El usuario inicia la aplicación.
- El sistema muestra una pantalla de inicio de sesión.
- El usuario inicia su usuario y contraseña.
- El sistema verifica la información.
- Si los datos son válidos, el usuario accede al programa.
- Si los datos son inválidos, el sistema deniega el acceso y pide ingresar credenciales correctas.

Caso de uso: Interactuar con las tablas del programa

Descripción: Permite al usuario realizar operaciones de agregar, editar, eliminar y ver con detalle cada elemento en las diferentes tablas del programa.

Eventos:

- El usuario va a la tabla que desea utilizar.
- El sistema muestra todos los elementos de la tabla.
- El usuario puede ver las opciones de operación (agregar, editar, eliminar o ver detalles).
- El sistema continúa el siguiente paso de la operación elegida.
- El usuario puede cancelar la operación antes de completarla.

Caso de uso: Agregar un elemento

Descripción: Permite al usuario agregar un nuevo elemento a una tabla específica del programa.

Eventos:

- El usuario presiona un botón de agregar en la tabla seleccionada.
- El sistema muestra un formulario de creación de elemento.
- El usuario ingresa los datos correspondientes en los campos del nuevo elemento.
- El sistema valida la información.
- Si los datos son válidos, el sistema agrega el elemento a la tabla y lo sincroniza con la base de datos.
- Si los datos son inválidos, el formulario no se cierra y se muestra un mensaje de error en los campos inválidos.

Caso de uso: Editar un elemento

Descripción: Permite al usuario editar un elemento existente en una tabla específica del programa.

Eventos:

- El usuario presiona un botón de editar en el elemento de la tabla seleccionada.
- El sistema muestra un formulario de edición de elemento.
- El usuario modifica los campos deseados del elemento.
- El sistema valida la información modificada.
- Si los datos son válidos, el sistema ejecuta los cambios del elemento a la tabla y lo sincroniza con la base de datos.
- Si los datos son inválidos, el formulario no se cierra y se muestra un mensaje de error en los campos inválidos.

Caso de uso: Eliminar un elemento

Descripción: Permite al usuario eliminar el elemento existente de una tabla específica del programa.

Eventos:

- El usuario presiona un botón de eliminar en el elemento de la tabla seleccionada.
- El sistema muestra una advertencia de confirmación de eliminación del elemento.
- Si se confirma, el sistema elimina el elemento de la tabla y de la base de datos.
- Si se cancela, el sistema no realiza cambio alguno.

Caso de uso: Ver detalles de un elemento

Descripción: Permite al usuario ver los detalles completos de un elemento en una tabla específica del programa.

Eventos:

- El usuario presiona un botón de ver detalles en el elemento de la tabla seleccionada.
- El sistema muestra un formulario con detalles más específicos del elemento.

Caso de uso: Guardar los cambios de una tabla

Descripción: Permite al usuario guardar los cambios realizados en una tabla específica del programa.

Eventos:

- El usuario realiza modificaciones en los elementos de una tabla.
- El sistema valida los cambios realizados.
- Si los datos son válidos, el sistema ejecuta los cambios del elemento a la tabla y lo sincroniza con la base de datos.
- Si los datos son inválidos, el formulario no se cierra y se muestra un mensaje de error en los campos inválidos.

Caso de uso: Agregar una venta (boleta o factura)

Descripción: Permite al usuario agregar una nueva venta (boleta o factura) al programa simulando una venta.

Eventos:

- El usuario elige la tabla que quiere ver, boletas o facturas.
- El usuario presiona el botón de agregar en la tabla de ventas.
- El sistema muestra el formulario de creación de venta.
- El usuario selecciona los datos modificables de la venta (cliente, productos, empleado y local).
- El sistema calcula automáticamente el subtotal, impuesto, descuento e importe final de la venta.
- El sistema considera si la venta es a una empresa o una persona para calcular el impuesto y el descuento correspondientes.
- El sistema muestra una vista previa de la venta con los cálculos realizados.
- El usuario confirma la venta y el sistema la guarda en la base de datos.

Caso de uso: Agregar ropa a la venta

Descripción: Permite al usuario agregar productos de ropa a una venta en curso.

Eventos:

- Durante la venta, el usuario puede agregar productos disponibles en el local.
- El usuario selecciona los productos de ropa que desea agregar.
- El usuario puede cambiar la cantidad del producto.
- El sistema actualiza los cálculos del subtotal y otros mientras cambia la cantidad y los productos.

Caso de uso: Comentar sobre la venta

Descripción: Permite al usuario agregar un comentario durante y después de la venta.

Eventos:

- Durante y después de la venta, el usuario puede poner un comentario personalizado.
- El sistema guarda el comentario y actualiza la tabla local y la base de datos.

Caso de uso: Editar y ver detalles de la venta

Descripción: Permite al usuario editar el comentario y ver los detalles completos de una venta.

Eventos:

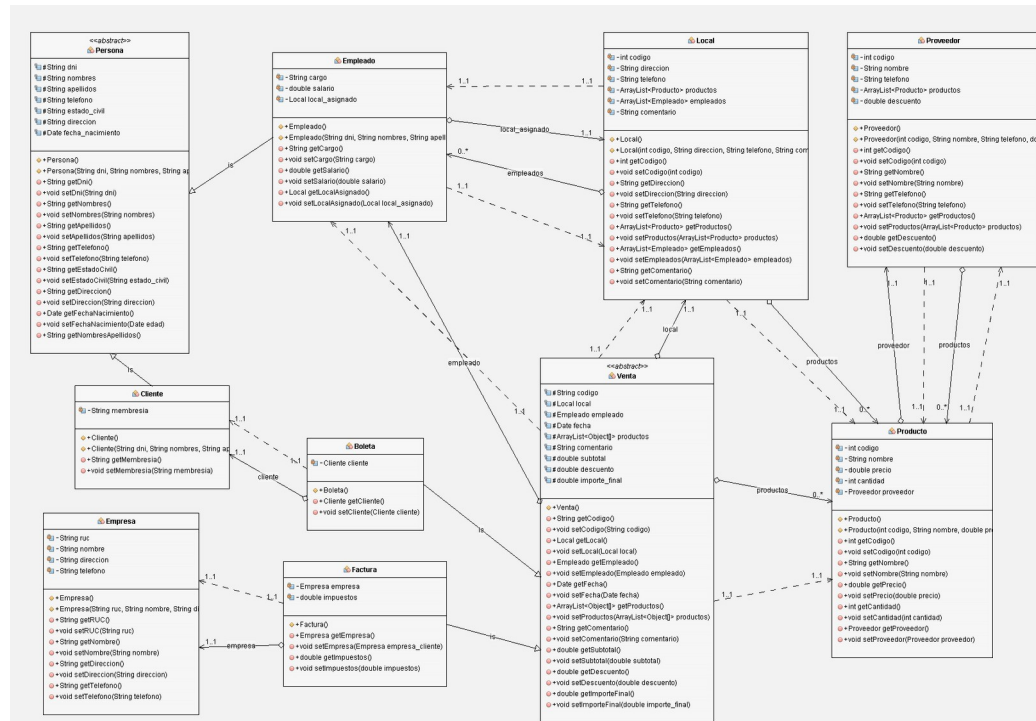
- El usuario presiona el botón de editar o ver detalle de una venta.
- El sistema muestra los detalles de la venta.
- Si está editando, el usuario puede alterar el comentario de la venta.
- El usuario puede guardar los cambios si modificó el comentario.

Código	Acción:	
	Consultar Stock	
Área	Inventario	
Precondición	Que exista el prodcto	
Descripción	Ver prodctos con sus atributos	
Secuencia normal	Paso	Acción
	1	Logearse al sistema
	2	Ir a la seccion Productos
	3	Dar click en la caja de busqueda
	4	Ingresar el nombre del producto
	5	Dar click en el sibolo de lupa para ver detalles del producto
Postcondición		
Excepciones	Paso	Acción
	1	El prodcto no exista
	2	Que no se repita los atribustos y alla conflicto con la existencia de otro producto
	3	
	4	
Comentarios		

Código	Acción:	
	Añadir nuevos proveedores	
Área	Gerencia	
Precondición		
Descripción	Añadir un nuevo proveedor para agregar productos, nueva mercancía además, se pueden o no tener un descuento por marca en las ventas	
Secuencia normal	Paso	Acción
	1	Logearse al sistema
	2	Ir a la sección Proveedores
	3	Dar clic en el símbolo más
	4	Ingresar sus atributos(nombre, teléfono, descuento)
	5	Dar clic en el botón guardar y se agregará el nuevo proveedor
Postcondición		
Excepciones	Paso	Acción
	1	Que no se repita los atributos y no haya conflicto con la existencia de otro proveedor
	2	Que el producto no exista
	3	
	4	
Comentarios		

Código	Acción:	
	Generar Venta	
Área	Ventas	
Precondición	Que los productos estén en el local y tengan stock	
Descripción	Poder generar una venta agregando productos, dependiendo si hay stock, además que la cantidad requerida no exceda el propio stock	
Secuencia normal	Paso	Acción
	1	Logearse al sistema
	2	Ir a la sección ventas
	3	Dar clic en el símbolo más
	4	Seleccionar el local de venta
	5	Seleccionar el cliente(De lo contrario, que el cliente no esté en el sistema, lo agregamos desde el símbolo más al lado de los nombres de los clientes)
	6	Seleccionamos el producto(También podemos agregar un nuevo producto)
	7	Ponemos la cantidad del producto o productos
	8	Damos clic en el botón Agregar
	9	
Postcondición		
Excepciones	Paso	Acción
	1	Agregar un producto inexistente
	2	Exceder el stock
	3	
	4	
Comentarios		

3.5. Diagrama de clases



3.6. *Presentación del prototipo*

3.7. Códigos de los formularios y pantallas de ejecución del programa informático



Usuario

Contraseña

Login

Salir



Sistema Ventas

Bienvenido

Tel: +51 958390645
Av. Huaylas 123, Chorrillos, Lima

- Bienvenido
- Empleados
- Productos
- Proveedores
- Cientes
- Ventas
- Locales

Logout

Salir

Empleado del Mes	Producto más popular	Proveedores de hoy
Martha Isabel Torres Díaz	Pantalones 23 vendidos	Index Gotcha

Cientes Plata:	Cientes Oro:	Cientes Diamante:	N° Ventas:	Ganancias:	Locales Populares
3	3	2	27	S/ 6955.23	Calle Las Begonias 456, Miraflo... Av. Javier Prado Este 123, San I...


```

CREATE PROCEDURE getBoletas
AS
BEGIN
    SELECT * FROM Boletas;
END;

CREATE PROCEDURE getProductosBoleta
@boleta_id VARCHAR(36)
AS
BEGIN
    SELECT producto_id, cantidad FROM ProductosBoleta
    WHERE boleta_id = @boleta_id;
END;

CREATE PROCEDURE addBoleta
@boletaId VARCHAR(36),
@localId INT,
@empleadoDni VARCHAR(8),
@fecha DATE,
@comentario VARCHAR(255),
@subtotal DECIMAL(10, 2),
@descuento DECIMAL(15, 5),
@importeFinal DECIMAL(10, 2),
@clienteDni VARCHAR(8)
AS
BEGIN
    INSERT INTO Boletas (id, local_id, empleado_dni, fecha, comentario, subtotal, descuento, importe_final, cliente_dni)
    VALUES (@boletaId, @localId, @empleadoDni, @fecha, @comentario, @subtotal, @descuento, @importeFinal, @clienteDni);
END;

CREATE PROCEDURE addProductosBoleta
@boleta_id VARCHAR(36),
@producto_id INT,
@cantidad INT
AS
BEGIN
    UPDATE Productos
    SET cantidad = cantidad - @cantidad
    WHERE id = @producto_id;

    INSERT INTO ProductosBoleta (boleta_id, producto_id, cantidad)
    VALUES (@boleta_id, @producto_id, @cantidad);
END;

```

Conclusiones

Es importante reconocer la necesidad de implementar un sistema de ventas en un negocio para mejorar la eficiencia en los procesos de venta, optimizar los recursos y aumentar las ventas. Esto permitirá competir de manera más efectiva en el mercado y brindar una mejor experiencia al cliente.

El uso de tecnología en los negocios presenta ventajas significativas, como la automatización de tareas repetitivas, la obtención de información valiosa sobre el comportamiento de los clientes y las tendencias del mercado, y la optimización de la gestión de inventario, proveedores y ventas.

Sin embargo, es importante tener en cuenta las limitaciones y desafíos que pueden surgir al implementar un sistema de ventas. Estos incluyen los costos de implementación y actualización, la dependencia de la tecnología, la necesidad de personal capacitado y una conexión a Internet estable, y la posible dificultad para integrar el sistema con otros sistemas o aplicaciones utilizados por el negocio.

La especificación de los requerimientos funcionales y no funcionales del sistema de ventas es fundamental para garantizar su correcto funcionamiento, seguridad y usabilidad. Esto incluye la capacidad de agregar, editar y eliminar elementos de las tablas, calcular automáticamente los montos de las ventas, mantener la integridad de los datos, proporcionar una interfaz intuitiva y manejar eficientemente un gran volumen de transacciones.

Los casos de uso definidos permiten comprender cómo interactúa el usuario con el sistema de ventas y qué acciones puede realizar. Estos casos de uso incluyen el inicio de sesión, la interacción con las tablas, la creación, edición y eliminación de elementos, la visualización de detalles de los elementos, y la gestión de ventas y comentarios.

En general, implementar un sistema de ventas adecuado puede brindar importantes beneficios para un negocio, mejorando la eficiencia operativa, la toma de decisiones y la experiencia del cliente. Sin embargo, es esencial considerar las necesidades y limitaciones específicas del negocio al seleccionar e implementar el sistema, y asegurarse de contar con los recursos y capacitación necesarios para su correcto funcionamiento y mantenimiento.

Bibliografía

Da Silva, D. (2022). *Base de datos de un sistema de ventas: ¿qué tipo de información contiene?* Zendesk. <https://www.zendesk.com.mx/blog/base-de-datos-de-un-sistema-de-ventas/>

Rodrigues, N. (2019). *Qué es un sistema de ventas: definición, características y ejemplos.* Hubspot. <https://blog.hubspot.es/sales/sistema-de-ventas>