



Universidad  
Tecnológica  
del Perú

**Facultad de Ingeniería**

**Carrera de Ingeniería de Sistemas e Informática**

**Sistema de control de ventas de la Distribuidora Group JEM EIRL**

**ALUMNOS:**

HANSEN SEIJAS, IAN ADRIAN U20240089

PIMENTEL ABAD, GERSON PAOLO STEFANO U20311434

PORTARO HINOSTROZA, JOEL ERNESTO ENRIQUE U21319867

RODRIGUEZ BERMEJO, MARCO TULIO U19215778

ROMERO ABAD, EDUARDO FAVIO U21203585

**PROFESOR:**

ROBALINO GOMEZ, HERNAN

**SECCIÓN:**

22565

2024

## ÍNDICE

<b>CAPÍTULO 1 - ASPECTOS GENERALES</b>	<b>6</b>
1.1. Descripción del problema	6
1.2. Definición de objetivos	8
1.2.1. Objetivo general	8
1.2.2. Objetivos específicos	8
1.3. Alcances y limitaciones	10
1.3.1. Alcances	10
1.3.2. Limitaciones	17
1.4. Justificación	17
1.5. Estado del arte	18
<b>CAPÍTULO 2 - MARCO TEÓRICO</b>	<b>21</b>
2.1. Marco teórico de programación orientada a objetos	21
2.1.1. Metodología	21
2.1.2. Modelo	25
2.1.3. Patrones de GUI	28
2.1.4. Patrones de desarrollo	32
2.1.5. Lenguajes de programación / framework	36
2.1.6. Gestor de base de datos	38
2.1.7. Herramientas adicionales	40
2.1. Marco teórico de las fórmulas y/o el sistema	41
2.1.1. Descripción de fórmulas	41
2.1.2. Importancia de los indicadores	44
<b>CAPÍTULO 3 - DESARROLLO DE LA SOLUCIÓN</b>	<b>44</b>
3.1. Prototipos	44
3.2. Modelo de proceso de negocios	47
3.2.1. Propuesta de valor	47
3.2.2. Diagrama de clases	49
3.2.3. Diagrama entidad - relación	50
3.2.4. Diccionario de datos	51
<b>BIBLIOGRAFÍA</b>	<b>55</b>

## ÍNDICE DE FIGURAS

Figura 1: Análisis causa - efecto en la Distribuidora Group JEM EIRL	7
Figura 2: Interfaz de vendedor - Registro de Ventas	19
Figura 3: Interfaz de administrador - Reporte del Personal	20
Figura 4: Diagrama WBS (Work Breakdown Structure)	21
Figura 5: Prototipo de Login	44
Figura 6: Prototipo de Inicio	45
Figura 7: Prototipo de Clientes	45
Figura 8: Prototipo registro de producto	46
Figura 9: Prototipo registro de nuevo producto	46
Figura 10: Prototipo de registro de nuevo cliente	47
Figura 11: Prototipo de registro de nuevo cliente	48
Figura 12: Diagrama de clases	49
Figura 13: Diagrama entidad relación lógico	50
Figura 14: Diagrama entidad relación físico	51

## ÍNDICE DE TABLAS

Tabla 1: Requerimiento funcional gestión de productos	11
Tabla 2: Requerimiento funcional control de inventario	11
Tabla 3: Requerimiento funcional registro de ventas	11
Tabla 4: Requerimiento funcional gestión de clientes	12
Tabla 5: Requerimiento funcional gestión de reportes	12
Tabla 6: Requerimiento funcional control de precios	12
Tabla 7: Requerimiento funcional gestión de proveedores	13
Tabla 8: Requerimiento funcional gestión de usuario	13
Tabla 9: Requerimiento funcional integración de sistema de pago	13
Tabla 10: Requerimiento funcional seguridad y respaldo de datos	14
Tabla 11: Requerimiento funcional interfaz de usuario	14
Tabla 12: Requerimiento funcional soporte de múltiples idiomas	14
Tabla 13: Requerimiento no funcional requerimiento de rendimiento	15
Tabla 14: Requerimiento no funcional requerimiento de disponibilidad	15
Tabla 15: Requerimiento no funcional de seguridad	15
Tabla 16: Requerimiento no funcional de mantenibilidad	16
Tabla 17: Requerimiento no funcional de usabilidad	16
Tabla 19: Productos	52
Tabla 20: Detalle de venta	52
Tabla 21: Venta	53
Tabla 22: Detalle de pedido	53
Tabla 23: Pedidos	53
Tabla 24: Cliente	54
Tabla 25: Usuario	54

## **INTRODUCCIÓN**

La Distribuidora Group JEM EIRL es una empresa ubicada en la Avenida Belisario Suarez N°240 en el distrito de San Juan de Miraflores que se dedica a la venta al por mayor de bebidas como la cerveza en presentaciones de 355 ml y 750 ml respectivamente dirigido al público que inicia sus operaciones desde setiembre del año 2021. La mencionada distribuidora nace como un pequeño negocio para eventos pequeños lo que motivó poco a poco a una expansión logrando tener un almacén de 100 mts cuadrados que contiene múltiples cajas de bebidas alcohólicas tales como Pilsen, Cristal y Cusqueña, marcas conocidas por los consumidores.

En la actualidad, es una distribuidora conocida en el cono sur debido a su pronta atención y calidad de sus productos; asimismo, dispone una propia distribución dado que cuenta con camiones para el abastecimiento a nivel local siendo contemplada con días de anticipación para la elaboración de guías y rutas de destino en la ciudad.

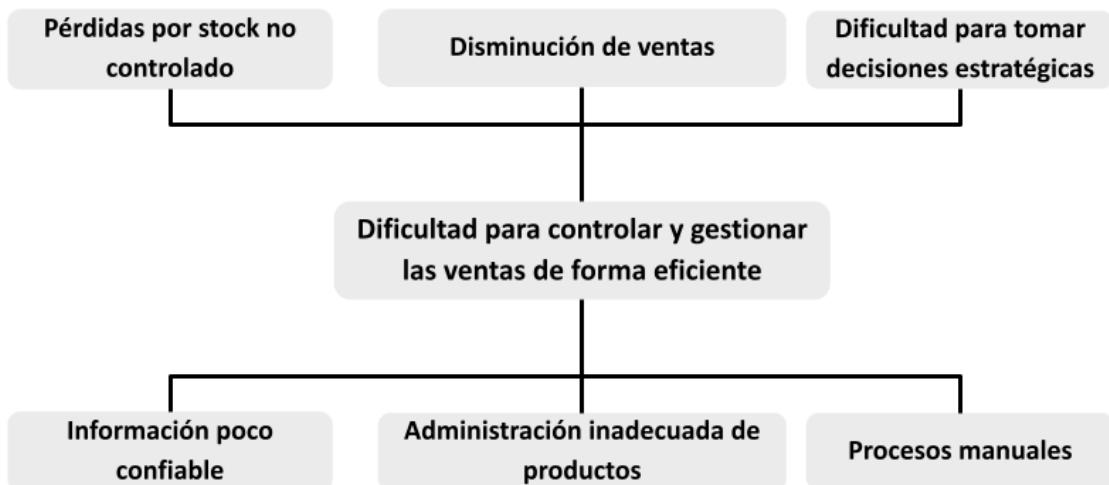
## CAPÍTULO 1 - ASPECTOS GENERALES

### 1.1. Descripción del problema

En la actualidad nos vemos frente a muchos cambios tecnológicos que sin duda, nos permiten optimizar tiempos como por ejemplo buscar algún producto solicitado para la realización de una venta; siendo más sencillo buscar por nombre o código de producto sea a través de un sistema, software o en la web considerando las necesidades que requiera el cliente antes de la ejecución de una transacción final.

La Distribuidora Group JEM EIRL se dedica a la venta de cervezas directo a los consumidores finales que como es una pequeña empresa no cuenta con algún mecanismo que permita al vendedor poder saber de forma rápida si cuenta con el stock necesario de un producto requerido por un usuario final. Cabe señalar además que, no sólo es una persona encargada de la tienda sino que, existen familiares que en su afán de ayudar en la atención no toman nota ni registro sobre las ventas y/o pedidos.

Como parte del proceso de pedidos, sólo se concreta dicha actividad más no un registro y seguimiento oportuno de los productos tanto al ingreso como de la salida; lo que lleva a que puede llegar un consumidor y no tener una atención rápida por no saber con qué se cuenta en la Distribuidora Group JEM EIRL.



**Figura 1: Análisis causa - efecto en la Distribuidora Group JEM EIRL**

Validando y verificando cuál es la situación de la Distribuidora Group JEM EIRL se logran identificar diversos obstáculos que hemos graficado en la figura 1 bajo el título de “*Análisis causa - efecto en la Distribuidora JEM EIRL*”. Cabe mencionar que, la problemática se centraliza de la siguiente manera: (i) la ausencia de un sistema informático integrado dado que, no se cuenta con un sistema que ayude en la operatividad diaria, (ii) administración inadecuada de productos dónde los colaboradores desconocen cómo poder mejorar las actividades y (iii) los procesos manuales que, en definitiva no permite mantener un orden y por ende nos brinda una información poco confiable.

De las causas antes mencionadas y teniendo idea de qué es lo sucede en la Distribuidora Group JEM EIRL tiene fuerte repercusión en cuanto al sistema de control de ventas, lo que, nos indica que es oportuno poder implementar un sistema informático que permite poder controlar el stock o la cantidad de productos que venden así como poder tomar decisiones adecuadas en cuanto a qué productos se debe adquirir para la continuación de las operaciones.

## **1.2. Definición de objetivos**

### **1.2.1. Objetivo general**

OG: Mejorar el control y gestión de ventas a través del desarrollo de un sistema informático para la Distribuidora Group JEM EIRL.

### **1.2.2. Objetivos específicos**

OE1: Automatizar el registro de ventas y pedidos.

OE2: Implementar un gestor de base de datos para los productos.

OE3: Implementar una funcionalidad que genere reportes y análisis detallados de ventas.

### **Indicadores de Gestión**

- Automatizar el registro de ventas y pedidos.

Indicador	Fórmula	Meta
<b>% de ventas registradas automáticamente</b>	(Ventas registradas automáticamente / Ventas totales) * 100	Al menos el 95% de las ventas deben registrarse de manera automatizada
<b>Tiempo promedio de registro de una venta</b>	(Tiempo total para registrar ventas / Número total de ventas)	Reducir el tiempo promedio en un 50% después de la implementación del sistema
<b>% de reducción en errores de registro de ventas</b>	(Errores antes - Errores después) / Errores antes) * 100	Reducir los errores en un 80%

- Implementar un gestor de base de datos para los productos.

Indicador	Fórmula	Meta
<b>% de productos registrados en la base de datos</b>	(Productos registrados / Total de productos en inventario) * 100	100% de los productos deben estar registrados en la base de datos
<b>Número de consultas de stock realizadas por los usuarios</b>	Número total de consultas correctas / Número total de consultas realizadas * 100	Al menos 90% de las consultas deben dar información correcta y actualizada
<b>Tiempo promedio para localizar un producto en el sistema</b>	(Tiempo total de consulta de productos / Número total de consultas)	El tiempo de consulta debe reducirse en un 70%

- Implementar una funcionalidad que genere reportes y análisis detallados de ventas

Indicador	Fórmula	Meta
<b>% de reportes generados automáticamente</b>	(Número de reportes generados automáticamente / Total de reportes) * 100	Al menos el 95% de los reportes deben generarse automáticamente
<b>Frecuencia de generación de reportes</b>	Número de reportes generados por período (diario, semanal, mensual)	Generar al menos un reporte diario, uno semanal y uno mensual de forma regular
<b>Nivel de detalle de los reportes</b>	(Reportes detallados / Total de reportes) * 100	Al menos el 90% de los reportes deben contener análisis detallado de las ventas

### **1.3. Alcances y limitaciones**

#### **1.3.1. Alcances**

El sistema de control de ventas de la Distribuidora Group JEM EIRL para el área de licores ofrece una gestión detallada del inventario. Permite monitorear en tiempo real las existencias de cada tipo de licor, actualizando automáticamente los niveles conforme se realizan ventas. También incluye alertas para reabastecimiento cuando los niveles de stock de algún licor caen por debajo de un umbral predefinido, ayudando a evitar quiebras de stock.

En cuanto al registro y facturación de ventas, el sistema facilita la emisión de facturas electrónicas o impresas para cada transacción. Esto asegura que todas las ventas de licores estén bien documentadas y cumplan con las regulaciones fiscales. También maneja descuentos y promociones específicas para licores, facilitando la gestión de campañas promocionales. El sistema proporciona herramientas para generar informes detallados sobre las ventas de licores. Estos informes pueden ser diarios, semanales o mensuales, y permiten analizar las ventas por tipo de licor, por cliente y tendencias estacionales. Esta capacidad de análisis ayuda a identificar los productos más populares y las épocas de mayor demanda. Además, el sistema almacena información sobre los clientes que compran licores, lo que facilita la personalización de ofertas y la implementación de programas de fidelización. Se pueden registrar las preferencias de los clientes y sus historiales de compras, lo que mejora la experiencia del cliente y permite ofrecer promociones específicas.

Por último, el sistema ayuda a asegurar el cumplimiento de las regulaciones relacionadas con la venta de licores, como la emisión de facturas con la información fiscal adecuada y el seguimiento de las ventas conforme a las normativas locales y nacionales.

## Requerimientos Funcionales

Requerimiento funcional N°	Nombre
RQF 01	Gestión de Productos
Tipo	Prioridad:
Gerente	Alta
Descripción	
El sistema debe permitir la gestión de los productos de la distribuidora, incluyendo alta, baja y modificaciones.	

**Tabla 1:** Requerimiento funcional gestión de productos

Requerimiento funcional N°	Nombre
RQF 02	Control de inventario
Tipo	Prioridad:
Gerente / Vendedor	Alta
Descripción	
El sistema debe llevar un control del inventario en tiempo real, actualizando las cantidades después de cada venta o reposición.	

**Tabla 2:** Requerimiento funcional control de inventario

Requerimiento funcional N°	Nombre
RQF 03	Registro de Ventas
Tipo	Prioridad:
Vendedor	Alta
Descripción	
El sistema debe permitir registrar las ventas realizadas, incluyendo la fecha, productos vendidos, cantidades y precios.	

**Tabla 3:** Requerimiento funcional registro de ventas

Requerimiento funcional N°	Nombre
RQF04	Gestión de Clientes
Tipo	Prioridad:
Gerente	Media
Descripción	
El sistema debe permitir registrar información de clientes, incluyendo datos de contacto e historial de compras.	

**Tabla 4:** Requerimiento funcional gestión de clientes

Requerimiento funcional N°	Nombre
RQF 05	Gestión de reportes
Tipo	Prioridad:
Gerente	Alta
Descripción	
El sistema debe generar reportes sobre ventas, inventario y desempeño general del área de ventas de la distribuidora.	

**Tabla 5:** Requerimiento funcional gestión de reportes

Requerimiento funcional N°	Nombre
RQF 06	Control de precios
Tipo	Prioridad:
Gerente	Media
Descripción	
El sistema debe permitir la actualización de precios de productos y reflejar estos cambios en tiempo real.	

**Tabla 6:** Requerimiento funcional control de precios

<b>Requerimiento funcional N°</b>	<b>Nombre</b>
RQF 07	Gestión de Proveedores
<b>Tipo</b>	<b>Prioridad:</b>
Gerente	Media
<b>Descripción</b>	
El sistema debe registrar y gestionar información de proveedores, incluyendo datos de contacto y productos suministrados.	

**Tabla 7: Requerimiento funcional gestión de proveedores**

<b>Requerimiento funcional N°</b>	<b>Nombre</b>
RQF 08	Gestión de Usuario
<b>Tipo</b>	<b>Prioridad:</b>
Gerente	Alta
<b>Descripción</b>	
El sistema debe permitir la creación y gestión de cuentas de usuario con diferentes niveles de acceso y permisos.	

**Tabla 8: Requerimiento funcional gestión de usuario**

<b>Requerimiento funcional N°</b>	<b>Nombre</b>
RQF 09	Integración del Sistema de Pago
<b>Tipo</b>	<b>Prioridad:</b>
Vendedor	Alta
<b>Descripción</b>	
El sistema debe integrarse con sistemas de pago para procesar transacciones con tarjeta de crédito/débito.	

**Tabla 9: Requerimiento funcional integración de sistema de pago**

Requerimiento funcional N°	Nombre
RQF 10	Seguridad y Respaldo de Datos
Tipo	Prioridad:
Gerente	Alta
Descripción	
El sistema debe garantizar la seguridad de los datos mediante autenticación y autorización, además de realizar respaldos periódicos.	

**Tabla 10:** Requerimiento funcional seguridad y respaldo de datos

Requerimiento funcional N°	Nombre
RQF 11	Interfaz de Usuario
Tipo	Prioridad:
Gerente/Vendedor	Alta
Descripción	
El sistema debe tener una interfaz de usuario amigable y fácil de usar para el personal de la bodega.	

**Tabla 11:** Requerimiento funcional interfaz de usuario

Requerimiento funcional N°	Nombre
RQF 12	Soporte para Múltiples Idiomas
Tipo	Prioridad:
Gerente	Baja
Descripción	
El sistema debe soportar múltiples idiomas, si es necesario, para facilitar al personal que hable otro idioma.	

**Tabla 12:** Requerimiento funcional soporte de múltiples idiomas

## Requerimientos No Funcionales

Requerimiento no funcional N°	Nombre
RNF 01	Rendimiento
Tipo	Prioridad:
Gerente/Vendedor	Baja
Descripción	
El sistema debe procesar las transacciones en menos de 2 segundos.	

**Tabla 13:** Requerimiento no funcional requerimiento de rendimiento

Requerimiento no funcional N°	Nombre
RNF 02	Disponibilidad
Tipo	Prioridad:
Gerente/Vendedor	Alta
Descripción	
El sistema debe contar con autenticación de roles.	

**Tabla 14:** Requerimiento no funcional requerimiento de disponibilidad

Requerimiento no funcional N°	Nombre
RNF 03	Seguridad
Tipo	Prioridad:
Gerente	Alta
Descripción	
El sistema debe tener una disponibilidad del 99.9%.	

**Tabla 15:** Requerimiento no funcional de seguridad

Requerimiento no funcional N°	Nombre
RNF 04	Mantenibilidad
Tipo	Prioridad:
Gerente	Media
Descripción	
El sistema debe permitir actualizaciones sin afectar la operación.	

**Tabla 16:** Requerimiento no funcional de mantenibilidad

Requerimiento no funcional N°	Nombre
RNF 05	Usabilidad
Tipo	Prioridad:
Vendedor	Alta
Descripción	
La interfaz debe ser intuitiva y fácil de usar para vendedores no técnicos.	

**Tabla 17:** Requerimiento no funcional de usabilidad

#### Leyenda:

- Descripción: Detalle específico de lo que se espera que el sistema cumpla en términos de rendimiento, seguridad, disponibilidad, etc.
- Prioridad: Nivel de importancia del requerimiento, que puede ser Alta, Media, o Baja. Un requerimiento con prioridad alta debe implementarse antes de otros con menor prioridad.
- Tipo: Hace referencia a los roles de los usuarios del sistema para los que aplica el requerimiento. Los tipos incluyen:

Gerente: Requerimiento aplicable a los usuarios con rol de gerente.

Vendedor: Requerimiento aplicable a los usuarios con rol de vendedor.

Gerente/Vendedor: Requerimiento aplicable a ambos roles (gerentes y vendedores).

### **1.3.2. Limitaciones**

El sistema enfrenta varias limitaciones. La dependencia de la tecnología ante cualquier falla en el software o en el hardware puede interrumpir las operaciones de venta, lo que puede afectar la eficiencia del negocio. La implementación del sistema también puede implicar una inversión inicial significativa en software y hardware, así como costos continuos de mantenimiento y actualizaciones, que pueden ser elevados.

La capacitación del personal es otro desafío, ya que se requiere formación para utilizar el sistema de manera efectiva. La falta de capacitación adecuada puede llevar a errores en el manejo de ventas e inventario, afectando la precisión del sistema. La seguridad de los datos es crucial, ya que el sistema debe proteger la información de ventas y de clientes contra accesos no autorizados y pérdidas. Esto puede requerir inversiones adicionales en medidas de seguridad cibernética. El sistema también debe ser adaptable a cambios en las regulaciones o en las tendencias del mercado. Si no está diseñado para ajustarse a estos cambios, puede convertirse en una limitación. Por ejemplo, ajustes en las leyes sobre la venta de alcohol o en las preferencias de los consumidores podrían requerir modificaciones en el sistema.

Por último, la dependencia del proveedor para soporte técnico y actualizaciones puede limitar la flexibilidad del sistema. Si el proveedor experimenta problemas o cambia sus políticas, esto puede impactar el funcionamiento del sistema de control.

### **1.4. Justificación**

El actual informe tiene como objetivo implementar un sistema de control de ventas en la Distribuidora Group JEM EIRL para registrar y rastrear el movimiento de productos, mejorando la gestión de inventarios. Esto solucionará la falta de registro y seguimiento de ventas, que actualmente causa demoras en el servicio y pérdida de oportunidades. El sistema mejorará la eficiencia, facilitará a los vendedores la verificación de disponibilidad de productos, reducirá errores y pérdidas, y aumentará la competitividad de la Distribuidora Group JEM EIRL.

La Distribuidora Group JEM EIRL enfrenta actualmente deficiencias en su sistema de gestión de sus ventas, lo que dificulta el control adecuado de sus operaciones diarias. La ausencia de un registro automatizado de ventas genera retrasos en la atención al cliente, pérdidas de estos, traspapeleo en el registro de ventas y errores en la disponibilidad de productos. Estos problemas se agravan por la falta de un sistema centralizado que permita gestionar eficientemente el inventario y acceder a información clave en tiempo real.

El proyecto busca desarrollar un sistema informático que automatice el registro de ventas y pedidos, permitiendo un proceso más ágil y preciso. La integración de un gestor de base de datos permitirá mantener un control actualizado del inventario, facilitando a los vendedores la verificación inmediata de la disponibilidad de productos y reduciendo errores por falta de stock o sobreabastecimiento. Además, el sistema generará reportes detallados de ventas, lo que brindará a la gerencia herramientas analíticas para identificar patrones de consumo, optimizar la toma de decisiones y mejorar la planificación de ventas.

Este sistema no solo mejorará la eficiencia operativa, sino que también permitirá a la empresa aprovechar mejor sus recursos, aumentando su competitividad y capacidad de respuesta en el mercado.

### **1.5. Estado del arte**

#### **DESARROLLO DE SISTEMA DE GESTIÓN DE VENTA DE LICORES PARA LA LICORERÍA "LA NENITA"**

Este sistema de ventas fue desarrollado porque la licorería "La Nenita", ubicado en Santa Rosa de Cuzubamba, enfrenta problemas similares a los de la Distribuidora Group JEM EIRL incluyendo la falta de registro sistemático de las transacciones, la dificultad para realizar un seguimiento preciso de entrada y salida de los productos y el riesgo de pérdidas y duplicaciones de información por las anotaciones manuales lo que resultaba en una operación ineficiente en el establecimiento. Por esto un estudiante de una escuela técnica de Quito, desarrolló un backend para la gestión de venta, algunas de las implementaciones que se

rescata del informe para nuestro proyecto fueron los endpoints para gestionar categorías, subcategorías y productos que nos ayudarán a ordenar, eliminar y distribuir el listado de productos añadidos al sistema.

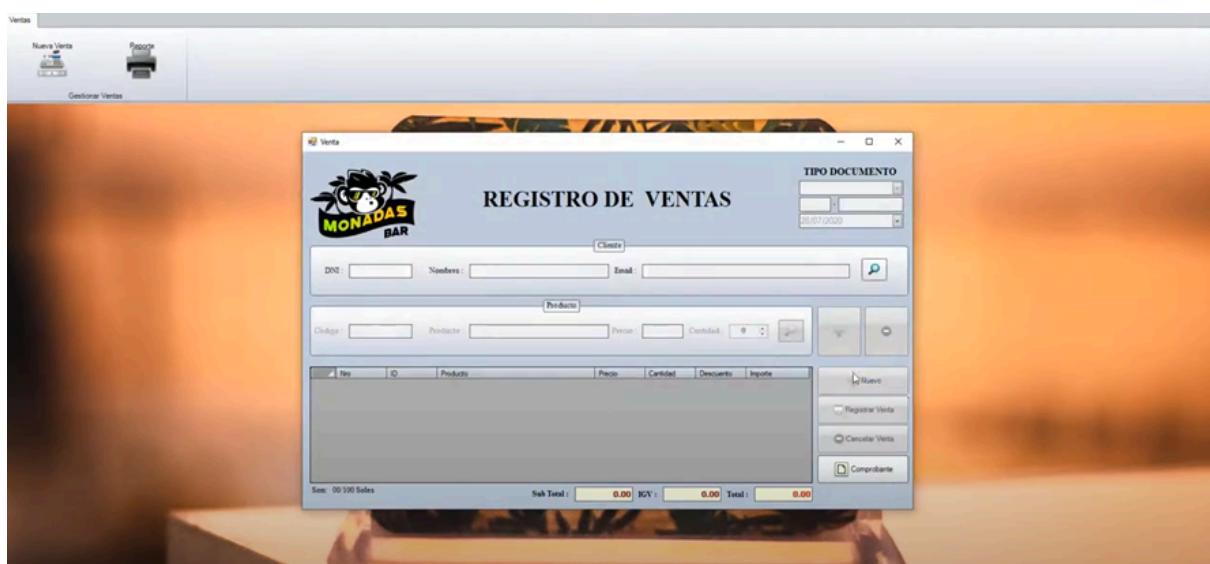
## Sistema de Ventas para Licorería

Este sistema fue desarrollado para una empresa enfocada en la venta de licores en la ciudad de Huaura, en las cuales utiliza los siguientes programas: Visual Studio 2019 y MySQL. Este sistema se enfoca en optimizar el proceso de ventas, gestionar usuarios y generar reportes administrativos. Se divide en tres interfaces:

**Vendedor:** Dedicada al registro de ventas y generación de reportes de compras realizadas.

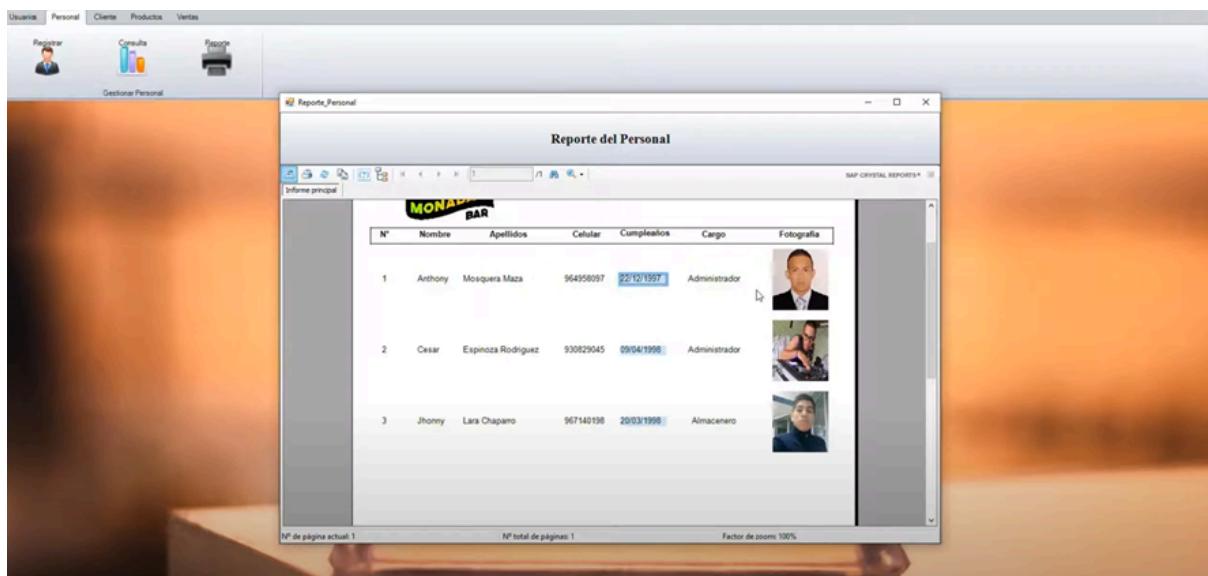
**Administrador:** Responsable de la gestión de usuarios, específicamente del personal administrativo, así como de la generación de reportes relacionados con la administración.

**Almacenero:** Aunque no es parte del enfoque principal del proyecto en sí, en sistemas similares, esta interfaz se encarga de la gestión de inventario y control de existencias.



**Figura 2: Interfaz de vendedor - Registro de Ventas**

(Auromecar,2019)



**Figura 3: Interfaz de administrador - Reporte del Personal**

(Auromecar,2019)

### Sistema informático para venta de licores

En este sistema de ventas de licores, utilizaron el método de cascada en la cual tuvo como fin dividir en diferentes fases, siempre y cuando cada una tenga que ser finalizada en su totalidad antes de comenzar las siguientes etapas del proyecto. Primero, en la fase de requisitos, se definieron los objetivos y expectativas del software junto con los requisitos que se debe implementar. La segunda etapa, el diseño, en donde se presenta las diferentes interfaces, tipografía y ubicación de los elementos que se añadirán, después de esta etapa a continuación, la implementación, en la cual se realiza el desarrollo del software de acuerdo a la arquitectura que se dio anteriormente, así mismo se pasaría a la siguiente etapa, la verificación, en esta fase se hace pruebas del software con el fin que de obtener resultados satisfactorias de acuerdo a las especificaciones y objetivos establecidos previamente, dando así paso a la última etapa, mantenimiento, en esta fase se realiza constante observación una vez se haya lanzado para su implementación con el fin de encontrar errores en la que sea necesario una actualización.

Esta metodología será fundamental para proporcionar una estructura clara y ordenada para el desarrollo del sistema de control de ventas en la Distribuidora Group JEM EIRL asegurando que cada aspecto del sistema sea cuidadosamente planificado, implementado y probado antes de su lanzamiento.

## Work Breakdown Structure

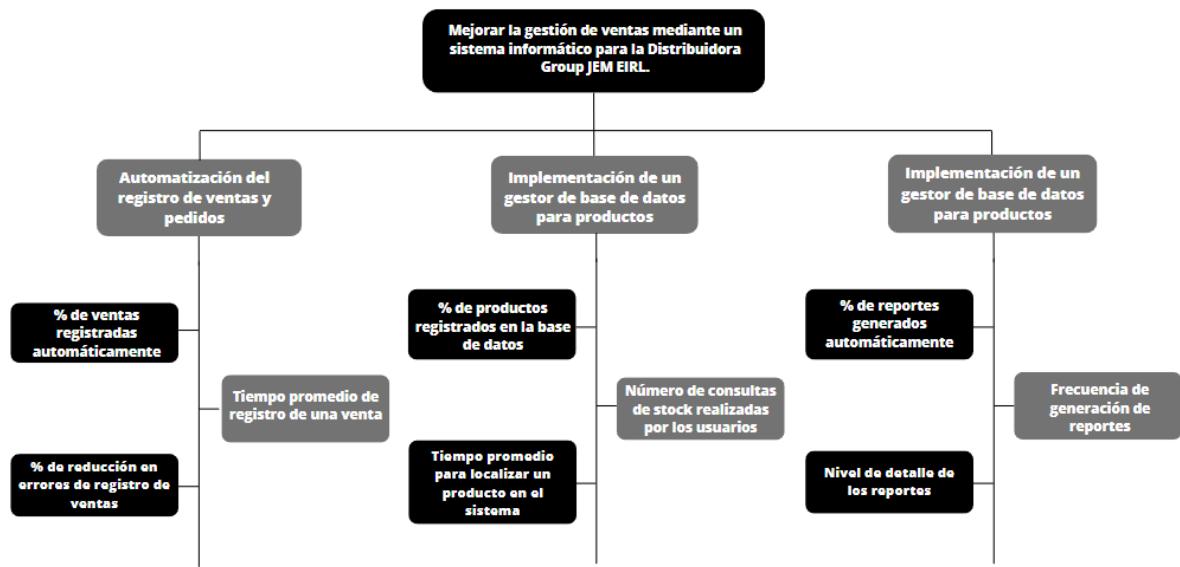


Figura 4: Diagrama WBS (Work Breakdown Structure)

## CAPÍTULO 2 - MARCO TEÓRICO

### 2.1. Marco teórico de programación orientada a objetos

#### 2.1.1. Metodología

En referencia a la metodología empleada según Sánchez *et al.* (2018) refiere que el método analítico se rige en poder estudiar por separado las variables; es por ello que, en la investigación en curso se puede revisar las variables a medida de poder descomponerlas y así poder determinar cuáles son los indicadores que formarán parte de nuestro estudio. Por ende, en el proyecto se ha empleado el método de cascada para desarrollar el sistema de control de ventas de Distribuidora Group

JEM EIRL, ya que sigue un ciclo de vida básico del desarrollo de software, utilizando principios de metodologías ágiles, lo que nos permitió trabajar de manera iterativa y flexible.

El proyecto de investigación busca diseñar un sistema que permita la interacción entre los datos ingresados y el comportamiento de elección de productos a través del vendedor y el consumidor respectivamente, dónde se considerarán las siguientes clases tales como:

- **Producto:** como el nombre del producto, precio y stock.
- **Clientes:** se tomará en cuenta nombre y apellido del cliente además de su número de DNI.
- **Forma de pago:** sea efectivo o bajo la modalidad de tarjeta sea débito o crédito.

Asimismo, de definirán los atributos y métodos como sigue:

- **Atributos:** en este punto se tendrá en cuenta la presentación del producto relacionado a los mililitros de las bebidas.
- **Métodos:** respecto a poder calcular el total del pedido, agregar otro ítem.

Es importante señalar que, se tiene contemplado la relación entre clases como son:

- **Herencia:** se contempla el acceso de un empleado con usuario administrador para otro colaborador.
- **Asociación:** mantener una lista de pedido que relacione lo que el consumidor adquiere regularmente.
- **Composición:** reconocer un pedido de ciertas especificaciones.



Método de Cascada

## Análisis

Durante la fase de análisis, identificamos los principales requerimientos del sistema a través de entrevistas con los usuarios clave y el análisis de los procesos actuales. El objetivo principal es poder automatizar el registro de ventas y mejorar el control de productos e inventarios. Nos enfocamos en requisitos funcionales, como la capacidad de registrar ventas y consultar el inventario en tiempo real, y no funcionales, como la seguridad y el rendimiento.

## Diseño

El diseño del sistema se realizó utilizando diagramas UML. Elaboramos un Diagrama de Casos de Uso para identificar las interacciones de los usuarios con el sistema, y un Diagrama de Clases para modelar las entidades clave como productos, ventas y usuarios. Además, diseñamos la interfaz gráfica con un enfoque en la simplicidad y facilidad de uso, considerando la capacitación mínima del personal.

Es importante señalar que, el sistema de ventas contempla lo siguiente:

- **Diagrama de clases:** tales como las representaciones gráficas entre las clases y relaciones mencionadas en el apartado de metodología.
- **Patrones de diseño:** Son las soluciones que permitirán solucionar problemas repetitivos a través de MySQL.
- **Base de datos:** permitiendo poder almacenar la información registrada por las operaciones con MySQL.
- **Interfaz de usuario:** se refiere a la interacción entre usuarios tales como el colaborador y el consumidor final.

## Implementación

La implementación se realizó utilizando Java para la lógica del sistema y MySQL como gestor de base de datos. Escogimos Java por su versatilidad y facilidad para integrarse con MySQL. Se adoptó una arquitectura modular para que el sistema sea escalable y fácil de mantener en el futuro.

```
Java

public class Producto {
    private String nombre;
    private double precio;
    private int stock;

    // Constructor, getters y setters

    public double calcularSubtotal(int cantidad) {
        return precio * cantidad;
    }
}
```

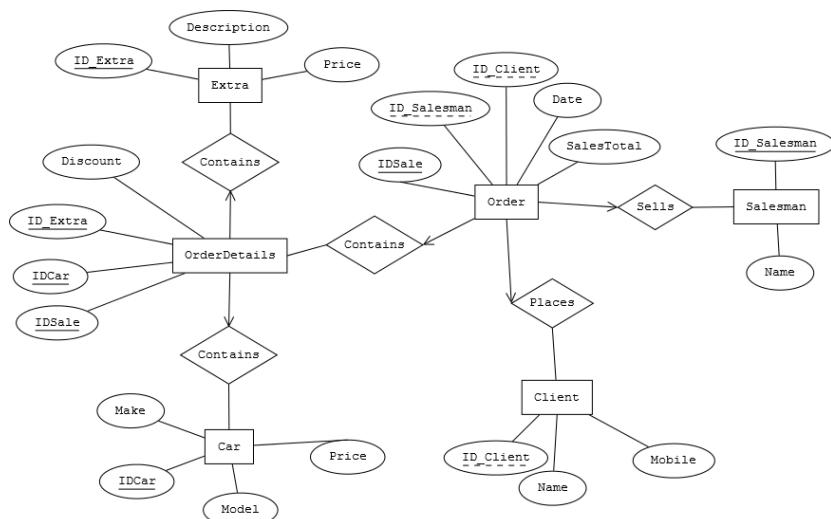
*Modelo lo lógica del sistema Java*

## Pruebas

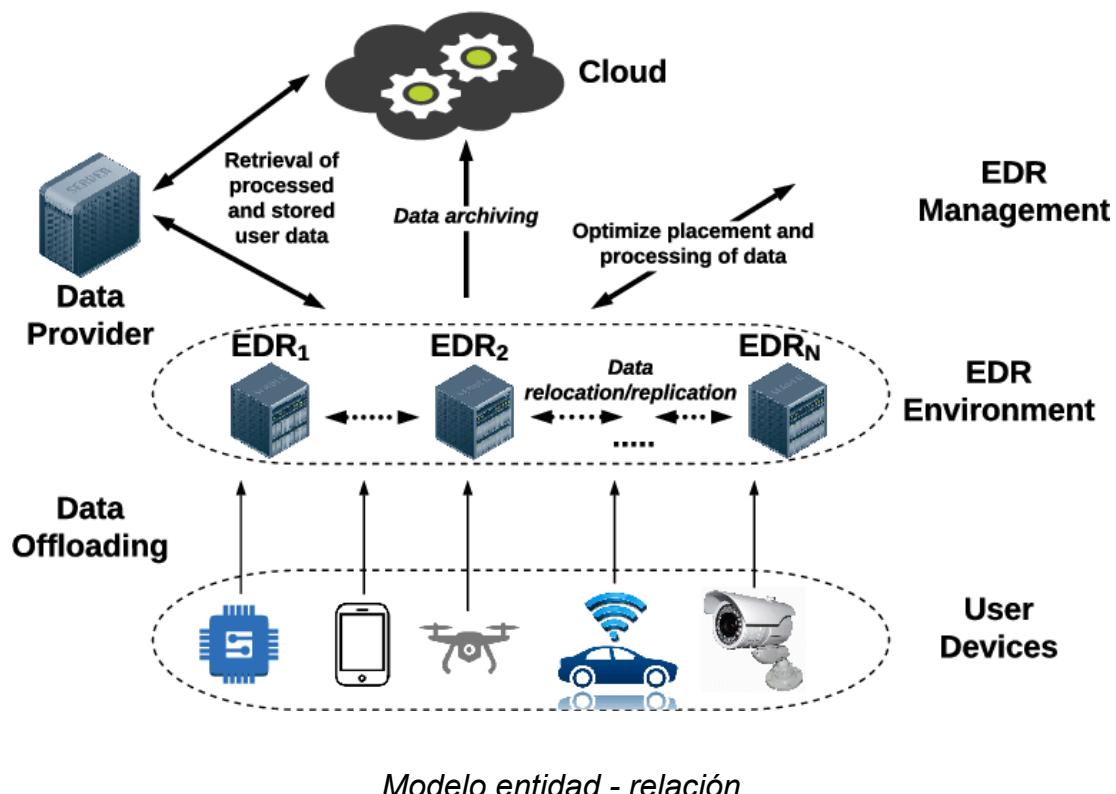
En la fase de pruebas, se realizaron pruebas unitarias y funcionales para verificar que cada componente funcione correctamente. Simulamos distintos escenarios de venta y consulta de inventario para asegurar que el sistema respondiera de manera efectiva. Además, probamos la seguridad del sistema mediante la creación de diferentes perfiles de usuario.

### 2.1.2. Modelo

**Modelo Entidad Relación (ER):** Tipo de diagrama de flujo que muestra cómo las "entidades", como personas, objetos o conceptos, están vinculadas entre sí dentro de un sistema. Estos diagramas se utilizan comúnmente en el diseño y depuración de bases de datos relacionales, especialmente en áreas como la ingeniería de software, sistemas de información empresarial, educación e investigación. Los modelos ERD o ER emplean símbolos específicos, como rectángulos, diamantes, óvalos y líneas, para representar entidades, relaciones y atributos, reflejando una estructura similar a la gramatical, donde las entidades actúan como sustantivos y las relaciones como verbos.



Modelo de entidad relación base de datos de sistema de ventas



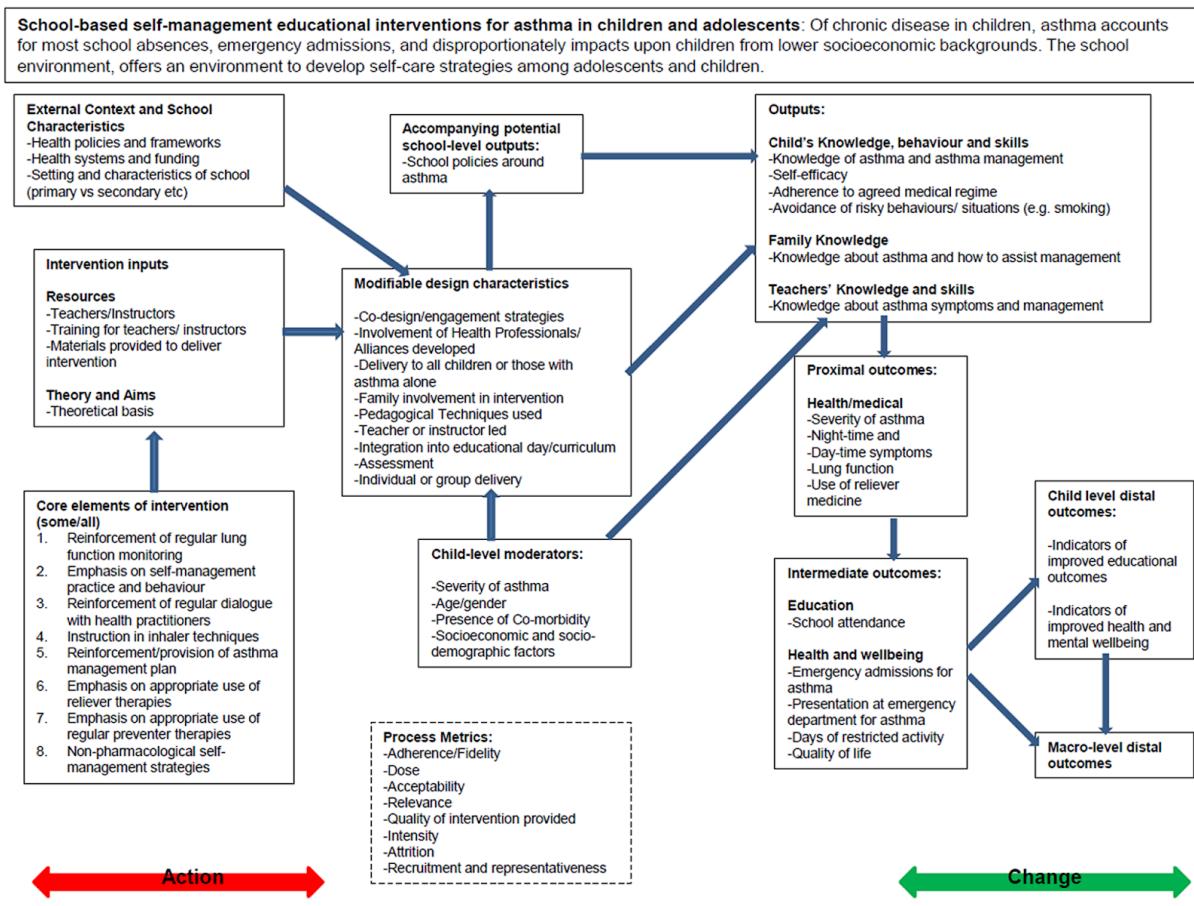
*Modelo entidad - relación*

**Modelo Lógico:** Un modelo lógico de datos es una representación abstracta que describe las necesidades de una organización para gestionar información y las relaciones entre los distintos elementos, sin estar ligado a una base de datos específica.

Este modelo abarca entidades, atributos, relaciones, identificadores únicos, subtipos, supertipos y restricciones. También puede incluir objetos de dominio o hacer referencia a modelos de dominio o glosarios. Una vez definidas las relaciones y objetos en el modelo lógico, puede transformarse en uno físico que refleje la estructura de una base de datos concreta.

Se puede crear a partir de plantillas, e importar tipos de datos desde archivos XML (.xsd) para emplearlos como tipos de dominio. El asistente facilita su configuración inicial, y a través del Explorador de proyectos es posible añadir objetos lógicos, cuyas propiedades pueden ajustarse en la vista correspondiente. Las modificaciones en claves, relaciones o atributos impactan en las claves externas vinculadas.

En proyectos colaborativos, es recomendable particionar los modelos lógicos para facilitar su manejo en equipo.

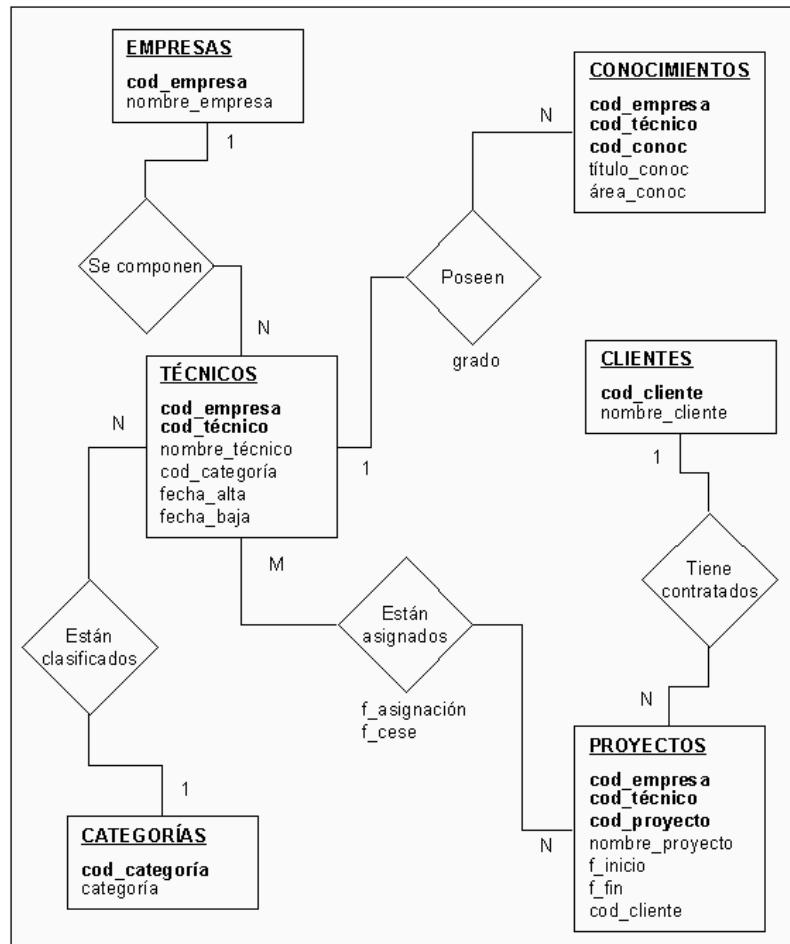


Modelo lógico

**Modelo Físico:** Un modelo de datos físico es una representación específica de una base de datos que describe objetos relacionales como tablas, columnas, claves primarias y externas, junto con sus relaciones. Este modelo se emplea para generar sentencias DDL que luego pueden ejecutarse en un servidor de base de datos.

Existen diversas formas de crear un modelo físico dentro del entorno de trabajo: se puede generar uno desde cero mediante un asistente, usar una plantilla, o realizar ingeniería inversa a partir de una base de datos o archivo DDL, arrastrando objetos desde el Explorador de orígenes de datos. También es posible importar un archivo de modelo físico o transformar un modelo lógico en uno físico.

El proceso típico con un modelo físico incluye: la creación del modelo mediante uno de los métodos mencionados, la adición de objetos como índices, vistas y elementos de almacenamiento, la generación del script DDL, y la ejecución de este para crear los objetos en el servidor de base de datos.



Modelo físico

### 2.1.3. Patrones de GUI

#### Patrón de Navegación

- Menú Principal: Un diseño que organiza las principales funciones del sistema en un menú lateral o superior, permitiendo a los usuarios acceder fácilmente a secciones como "Ventas", "Clientes", "Inventario", "Reportes" y "Configuraciones". La claridad en la nomenclatura y la jerarquía visual son clave para una navegación efectiva.

- **Breadcrumbs:** Un rastro visual que muestra la ruta del usuario dentro de la aplicación, facilitando la navegación hacia atrás. Esto es especialmente útil en aplicaciones con múltiples niveles de contenido.

**Jerárquica:** Un menú principal con submenús para organizar las funciones.

**Basada en tareas:** Organiza la navegación según las tareas del usuario (realizar una venta, gestionar productos).

**Pestañas:** Permite cambiar entre diferentes vistas dentro de una misma pantalla.

**Breadcrumb:** Muestra el camino recorrido por el usuario.

## Patrón de Formulario

- **Formulario de Venta:** Diseñado para ser simple y directo, este formulario debe incluir campos obligatorios claramente marcados y etiquetas descriptivas. La validación en tiempo real ayuda a los usuarios a corregir errores antes de enviar la información, mejorando la experiencia general.
- **Auto-sugerencias:** Al ingresar texto en campos como "Buscar producto" o "Buscar cliente", se muestran sugerencias basadas en los caracteres ingresados. Esto no solo ahorra tiempo, sino que también ayuda a prevenir errores de escritura.

**Diseño claro:** Utiliza etiquetas claras y ordena los campos de manera lógica.

**Validación en tiempo real:** Muestra errores al usuario mientras completa el formulario.

**Ayuda contextual:** Proporciona información adicional sobre cada campo.

## Patrón de Tabla

- **Lista de Ventas:** Esta tabla presenta un resumen claro de todas las ventas realizadas, permitiendo a los usuarios ordenar los datos por diferentes columnas (fecha, cliente, total, etc.) y aplicar filtros para encontrar información específica rápidamente.
- **Detalles en Línea:** Al hacer clic en una venta específica, se puede mostrar un panel lateral o un modal con más detalles, evitando que el usuario salga de la vista de lista. Esto mantiene la fluidez de la interacción.

**Cabecera clara:** Define las columnas con encabezados concisos.

**Ordenamiento:** Permite ordenar los datos por diferentes columnas.

**Filtrado:** Facilita la búsqueda de datos específicos.

**Paginación:** Divide grandes conjuntos de datos en páginas más pequeñas.

## Patrón de Notificación

- **Alertas y Mensajes:** Implementar un sistema de notificaciones que informe a los usuarios sobre el estado de las transacciones, incluyendo mensajes de éxito, advertencias o errores. Por ejemplo, un mensaje de confirmación como "Venta registrada con éxito" proporciona retroalimentación instantánea y positiva.

**Mensajes claros:** Informa al usuario sobre el resultado de una acción.

**Ubicación visible:** Las notificaciones deben aparecer en un lugar destacado.

**Duración adecuada:** Las notificaciones deben desaparecer después de un tiempo.

## Patrón de Dashboard

- **Panel de Control:** Un diseño que centraliza información clave como ventas totales, productos más vendidos y estadísticas de clientes en un solo lugar. Utiliza gráficos y visualizaciones interactivas para ayudar a los usuarios a interpretar datos rápidamente y tomar decisiones informadas.

**Visualización de datos clave:** Presenta los datos más importantes de forma gráfica.

**Personalización:** Permite al usuario personalizar los widgets.

**Interactividad:** Permite al usuario interactuar con los datos.

### Patrón de Filtros

- **Filtrar Ventas:** Proporcionar opciones de filtro permite a los usuarios especificar criterios como rangos de fechas, tipos de productos o estados de venta (completadas, pendientes, canceladas). Esto mejora la capacidad de búsqueda y hace que la gestión de ventas sea más eficiente.

### Patrón de Accesibilidad

- **Compatibilidad Móvil:** Asegurarse de que la interfaz se adapte bien a diferentes dispositivos y resoluciones es esencial, ya que muchos usuarios pueden acceder al sistema desde teléfonos y tablets.
- **Contraste y Tamaño de Fuente:** Utilizar colores contrastantes y un tamaño de fuente legible garantiza que todos los usuarios, incluidas las personas con discapacidades visuales, puedan navegar por el sistema sin dificultad.

**Contraste de colores:** Utiliza una combinación de colores que facilite la lectura.

**Tamaños de fuente:** Ofrece opciones para ajustar el tamaño de la fuente.

**Teclado:** Permite navegar y realizar todas las acciones utilizando únicamente el teclado.

### Patrón de Asistente

- **Wizard de Configuración:** Un proceso guiado que ayuda a los usuarios a configurar el sistema inicial, como agregar productos y métodos de pago. Este patrón facilita la complejidad al dividir el proceso en pasos claros y manejables, mejorando la usabilidad.

**Guianza paso a paso:** Divide un proceso complejo en pasos más sencillos.

**Información contextual:** Proporciona información relevante en cada paso.

### Patrón de Búsqueda

- Barra de Búsqueda Global: Implementar una barra de búsqueda en la parte superior de la interfaz que permita a los usuarios buscar productos, clientes o ventas rápidamente desde cualquier parte del sistema. Es fundamental que los resultados sean relevantes y se presenten de forma clara.

**Campo de búsqueda destacado:** Coloca el campo de búsqueda en un lugar visible.

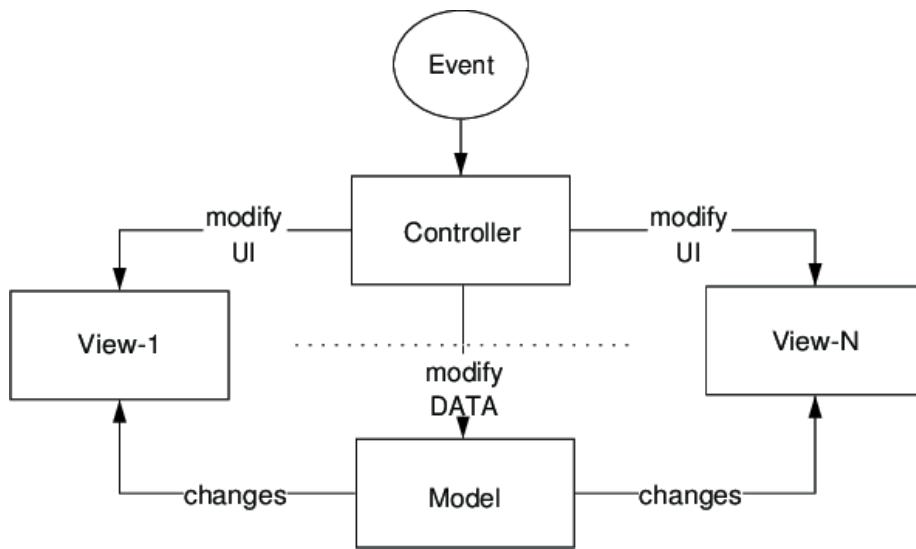
**Autocompletar:** Sugiere opciones a medida que el usuario escribe.

**Filtros:** Permite refinar los resultados de la búsqueda.

#### 2.1.4. Patrones de desarrollo

##### Patrón MVC (Modelo-Vista-Controlador): (Presentación Acceso a Datos)

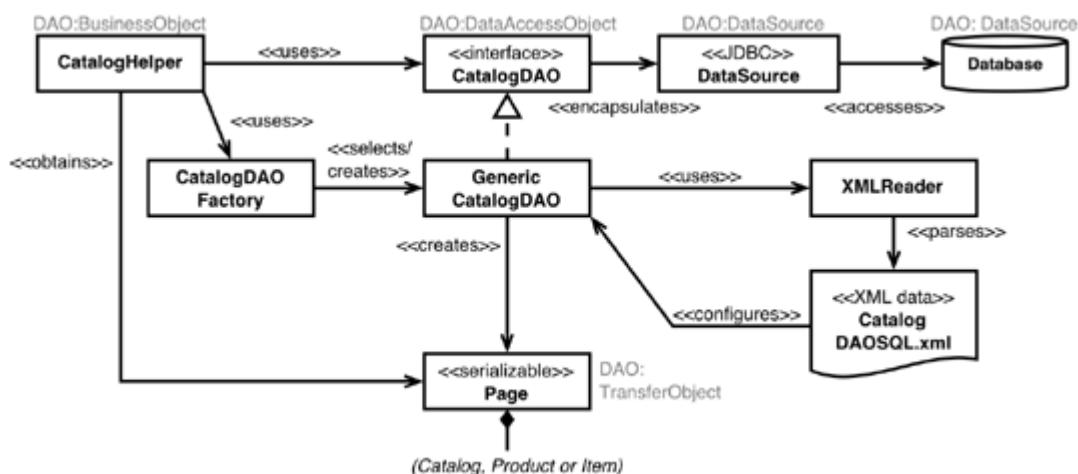
- El patrón MVC separa la aplicación en tres capas distintas, lo que permite gestionar la complejidad de sistemas de gran escala. El Modelo gestiona los datos (en este caso, utilizando SQL para las operaciones sobre la base de datos), la Vista maneja la interfaz de usuario (con JavaScript para proporcionar interactividad y dinamismo en el frontend), y el Controlador procesa las entradas de usuario y actualiza el modelo o la vista según sea necesario.
- Aplicación en el Sistema: El sistema de control de ventas puede estructurarse de manera que el Modelo sea responsable de las transacciones de ventas, inventarios, y la información de clientes; el Controlador maneja las acciones del usuario como realizar una venta o generar un reporte, y la Vista se encargará de mostrar la información actualizada al usuario final en el navegador.



*Modelo de aplicación en el sistema*

### Patrón DAO (Data Access Object):

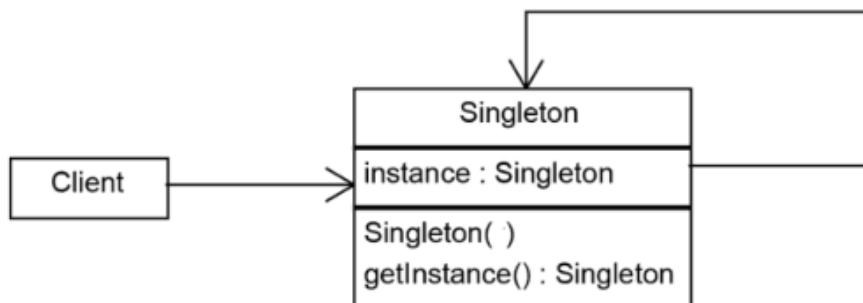
- El patrón DAO establece una capa entre la lógica del negocio y el acceso a los datos, facilitando el manejo de las operaciones CRUD (Crear, Leer, Actualizar, Borrar) en la base de datos. Esto no solo mejora el mantenimiento del sistema, sino que también permite cambiar el tipo de base de datos sin afectar el resto de la aplicación. En el contexto del sistema, se podría usar un DAO para acceder a la base de datos SQL, encapsulando todas las operaciones relacionadas con ventas, productos y clientes.
- Aplicación en el Sistema: Se implementarán clases DAO que gestionen todas las consultas SQL, asegurando una estructura más limpia y desacoplada, lo que facilita futuras modificaciones.



## Externalización de DAO SQL

### Patrón Singleton:

- Este patrón asegura que una clase tenga solo una instancia y proporciona un punto global de acceso a ella. En un sistema de ventas, sería útil para gestionar conexiones a bases de datos o controladores de recursos compartidos, como la configuración de la base de datos o la gestión de sesiones de usuario. Al asegurarse de que solo haya una instancia de conexión a la base de datos, se optimiza el uso de recursos y se evita la creación innecesaria de múltiples conexiones.
- Aplicación en el Sistema: Este patrón se aplicará para controlar el acceso a la base de datos SQL, asegurando que solo una conexión activa esté en uso durante la vida del sistema, lo que previene problemas de sobrecarga y manejo incorrecto de recursos.

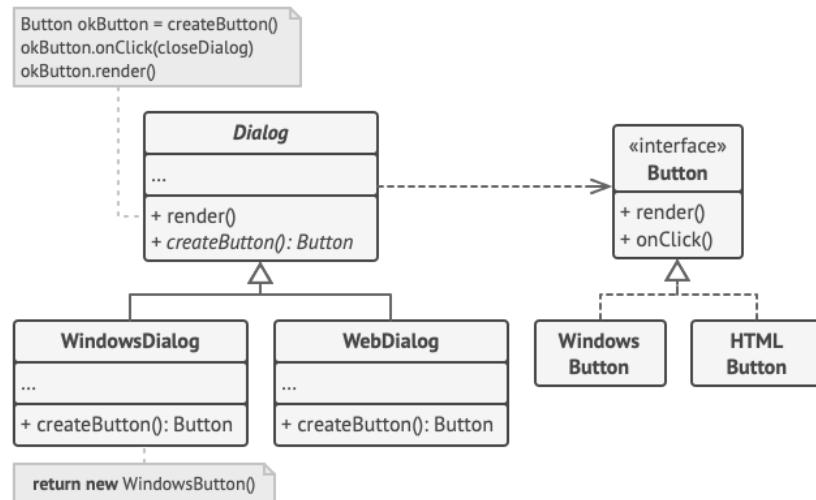


Modelo Patrón Singleton

### Patrón Factory (Fábrica):

- El patrón Factory es útil cuando se necesita crear objetos que pertenecen a una clase común, pero con ligeras diferencias en su configuración. En este sistema, puede ser útil para crear diferentes tipos de objetos de ventas o reportes en función de los datos de entrada. De esta manera, la creación de objetos es flexible y se adapta a las necesidades específicas de la aplicación.
- Aplicación en el Sistema: Se puede usar el patrón Factory para crear objetos de facturas o recibos, dependiendo de si se trata de una venta al por mayor o

al por menor, permitiendo al sistema adaptarse dinámicamente a diferentes tipos de transacciones.

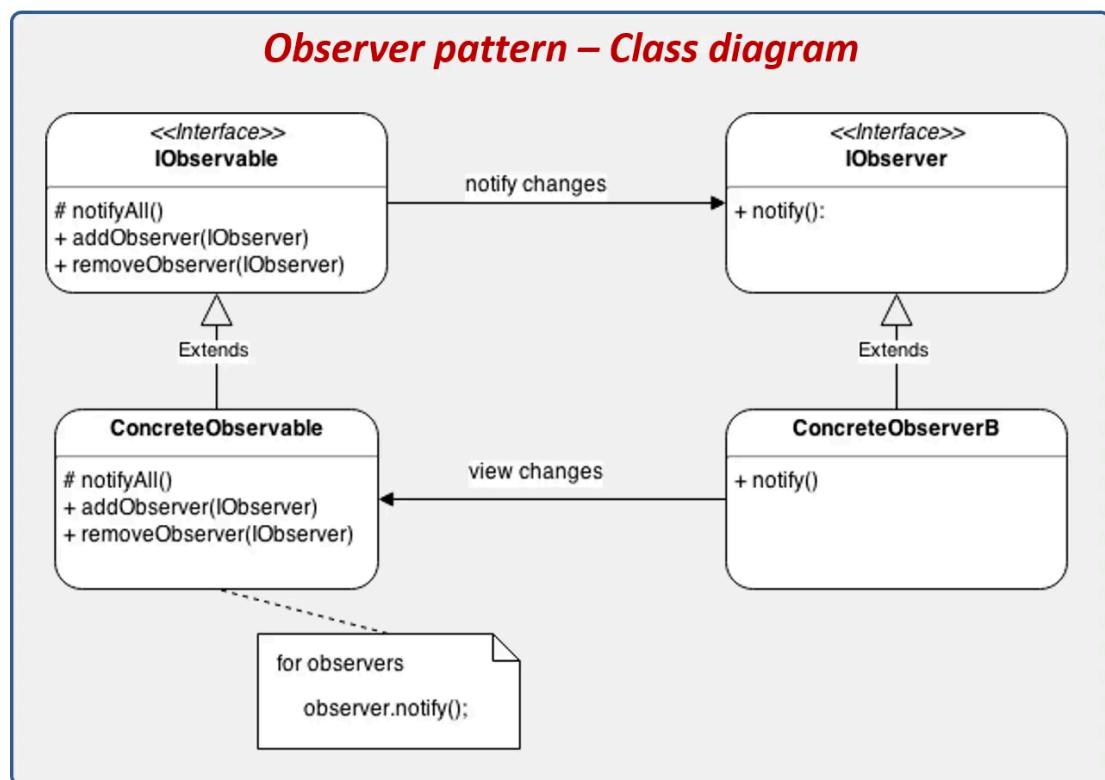


Modelo Patrón Factory

### Patrón Observer (Observador):

- El patrón Observer permite que un objeto notifique automáticamente a otros objetos cuando cambia su estado. Esto es útil en sistemas donde se necesita actualizar múltiples componentes cuando se realiza una acción. Por ejemplo, en el sistema de control de ventas, cuando se actualiza el inventario después de una venta, otros módulos del sistema (como reportes o estadísticas) pueden ser notificados automáticamente para que reflejen los nuevos datos.
- Aplicación en el Sistema: Este patrón será útil para notificar a varios componentes cuando ocurra una venta. Por ejemplo, al registrar una nueva venta, el sistema puede actualizar automáticamente el inventario y generar estadísticas de ventas sin que el usuario tenga que ejecutar varias acciones.

## **Observer pattern – Class diagram**



Modelo Patrón Observador

### **2.1.5. Lenguajes de programación / framework**

#### **Java**

La elección de Java es una buena alternativa para el desarrollo del sistema de ventas debido a su portabilidad, robustez y amplia comunidad dado que está más orientada a los objetivos, siendo independiente y sobre todo por el uso para desarrollo empresarial.

Java es un lenguaje que nos permite programar orientada a objetos con una finalidad puntual como es el caso de implementar nuestro sistema de ventas brindando mayores oportunidades a un negocio como la optimización no solo de tiempo sino de recursos.

Si hablamos de características podemos citar las siguientes:

- **Orientada a objetos:** básicamente es un modelo o prototipo que facilita los procesos a la organización, un detalle importante es que se puede reutilizar el código.
- **Portabilidad:** cabe señalar que, el código Java como tal se compila en bytecode, siendo un formato de nivel intermedio que permite ejecutarse en cualquier máquina de Java (JVM), volviéndola portátil y fácil de trasladar.
- **Seguridad:** su propio sistema permite proteger aplicaciones y datos que se vayan almacenando.
- **Robustez:** debido a que cuenta con un sistema de gestión que permite almacenar de forma automática así como cuenta con un compilador cuya función es prevenir errores comunes.
- **Concurrencia:** es porque permite la programación multihilo que ayuda a ejecutar diversas tareas simultáneamente.

Un punto muy importante en Java es mencionar sobre su versatilidad que lo vuelve ideal debido a su amplia gama de aplicaciones que son implementadas en el sector de negocios, donde podemos mencionar lo siguiente:

- **Desarrollo de aplicaciones web:** Frameworks tales como Spring y JavaServer Faces lo cuales nos facilitan la creación de aplicaciones web que sean robustas y escalables.
- **Desarrollo de aplicaciones empresariales:** este lenguaje se utiliza para construir sistemas de gestión de relaciones con clientes (CRM), sistemas de planificación de recursos empresariales (ERP) y otras aplicaciones que son críticas para el mundo empresarial.
- **Desarrollo de aplicaciones móviles:** Java se ha convertido en un lenguaje muy conocido para el desarrollo o elaboración de aplicaciones en dispositivos Android debido a su herramienta Android Studio.
- **Big data:** es usado para el procesamiento de grandes volúmenes de datos debido a sus frameworks como Hadoop y Spark.
- **Internet de las cosas (IoT):** lo que permite la interacción con dispositivos IoT.



## SQL

- SQL es un lenguaje de consulta que permite gestionar bases de datos relacionales. En este sistema, SQL se encargará de almacenar la información de ventas, productos, clientes y usuarios. Su robustez y capacidad de realizar consultas complejas lo hacen ideal para gestionar grandes volúmenes de datos en tiempo real, garantizando la consistencia y seguridad de la información.
- Aplicación en el Sistema: El sistema de ventas utilizará SQL para realizar consultas sobre el inventario, generar reportes de ventas, y mantener registros históricos de transacciones. Se implementarán procedimientos almacenados para optimizar consultas frecuentes, como la obtención de los productos más vendidos o el balance diario de ventas.



### 2.1.6. Gestor de base de datos

Para el almacenamiento y gestión de los datos del sistema, utilizamos MySQL. Este gestor de bases de datos relacional fue elegido por ser gratuito, eficiente y ampliamente utilizado en proyectos de tamaño mediano como el nuestro. MySQL soporta grandes volúmenes de datos y ofrece la seguridad necesaria para proteger la información crítica de la empresa.

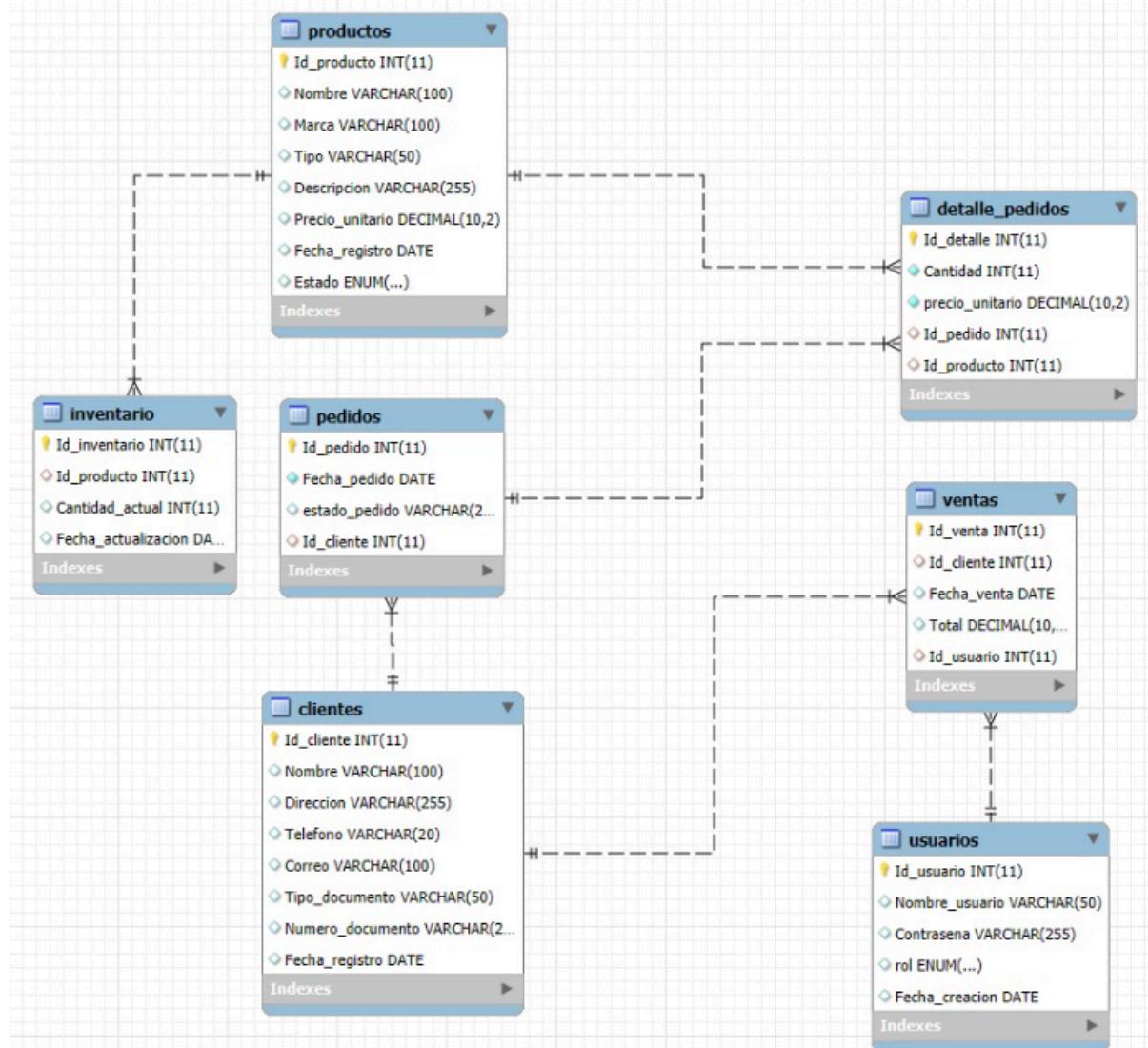
#### Características de MySQL

1. Escalabilidad: MySQL es capaz de manejar grandes cantidades de datos, lo que nos asegura que el sistema podrá crecer junto con el negocio.

2. Compatibilidad: Es compatible con múltiples plataformas y lenguajes, permitiendo que se integre fácilmente con nuestro sistema desarrollado en Java.
3. Seguridad: MySQL cuenta con herramientas para garantizar la seguridad de los datos, como autenticación de usuarios, control de acceso, seguridad de red y cifrado de conexiones.

## Modelo Relacional

En el modelo relacional de nuestra base de datos, las tablas más importantes son las de productos y ventas. Estas tablas están interconectadas mediante claves primarias y foráneas para garantizar la integridad de los datos. Cada venta se registra con referencia a un producto, lo que facilita la trazabilidad de la información.



## Modelo relacional

### 2.1.7. Herramientas adicionales

#### Figma

Figma es una extensión dentro de Figma que permite gestionar documentación relacionada con los proyectos de diseño. Facilita la integración de notas, especificaciones y descripciones directamente dentro del espacio de trabajo de diseño, mejorando la colaboración entre diseñadores, desarrolladores y otros miembros del equipo. Al funcionar dentro de Figma, evita la necesidad de utilizar herramientas externas para documentar el proceso de diseño, proporcionando un entorno de trabajo más eficiente y centralizado. (Figma, 2023)



#### NetBeans:

- NetBeans es un IDE que soporta múltiples lenguajes de programación, incluyendo JavaScript y SQL. Su principal ventaja radica en su capacidad para gestionar grandes proyectos con múltiples clases e interfaces, así como para integrar herramientas de bases de datos directamente en el entorno de desarrollo. Además, NetBeans permite la depuración en tiempo real, lo que mejora la eficiencia del desarrollo y mantenimiento del sistema (Oracle, 2020).



## **BIZAGI:**

- BIZAGI es una plataforma de gestión de procesos de negocio (BPM por sus siglas en inglés) que permite a las empresas modelar, automatizar y optimizar sus procesos ofrece herramientas para crear diagramas de flujo de trabajo diseñar procesos monitorear su ejecución en tiempo real.



### **2.1. Marco teórico de las fórmulas y/o el sistema**

#### **2.1.1. Descripción de fórmulas**

El sistema utiliza diversas fórmulas para realizar cálculos automáticos que permiten medir el rendimiento y la eficiencia del proceso de ventas.

#### **Indicadores de Gestión**

##### **- Rotación de Inventario (RI)**

Este indicador mide cuántas veces el inventario se ha renovado completamente en un periodo determinado.

$$RI = \frac{\text{Costo de Ventas}}{\text{Inventario Promedio}}$$

**Costo de Ventas:** El valor de los productos vendidos dentro de un periodo.

**Inventario Promedio:** Es el valor promedio del inventario al principio y al final del periodo.

##### **- Cobertura de Inventario (CI)**

Con esta fórmula evaluamos cuántos días el inventario actual puede satisfacer la demanda de nuestros clientes sin necesidad de reponer el inventario.

$$CI = \frac{\text{Inventario Actual}}{\text{Promedio Diario de Ventas}}$$

**Inventario Actual:** Cantidad actual de productos en inventario.

**Promedio Diario de Ventas:** Las ventas totales del periodo divididas por el número de días.

## Indicadores Financieros

### - Margen Bruto (MB)

Mide la rentabilidad bruta de las ventas, indicando el porcentaje que queda después de cubrir el costo de los productos vendidos.

$$MB = \frac{\text{Ventas Netas} - \text{Costo de Ventas}}{\text{Ventas Netas}} \times 100$$

**Ventas Netas:** Ingresos totales por ventas después de devoluciones y descuentos.

**Costo de Ventas:** Costo de los productos vendidos.

### - Margen Neta (MN)

Nos indica la rentabilidad neta de la empresa, considerando los costos operativos y financieros.

$$MN = \frac{\text{Utilidad Neta}}{\text{Ventas Netas}} \times 100$$

**Utilidad Neta:** Ganancias después de descontar todos los costos y gastos.

## Indicadores de Ventas

- **Crecimiento de Ventas (CV)**

Medimos el crecimiento de las ventas en comparación con el periodo pasado.

$$CV = \frac{\text{Ventas del periodo actual} - \text{Ventas del periodo anterior}}{\text{Ventas del periodo anterior}} \times 100$$

- **Ticket Promedio de Ventas (TPV)**

Determina el valor promedio de las ventas por transacción generadas en las operaciones de la empresa.

$$TPV = \frac{\text{Ventas Totales}}{\text{Número de Transacciones}}$$

## Indicadores de Servicio al Cliente

- **Tasa de Retención de Clientes (TRC)**

Mide el porcentaje de clientes que regresan a realizar compras después de su primera transacción.

$$TRC = \frac{\text{Clientes recurrentes}}{\text{Clientes totales}} \times 100$$

- **Nivel de Satisfacción del Cliente (NSC)**

Estima cuán satisfechos están los clientes con el servicio recibido.

$$NSC = \frac{\text{Total de puntuaciones de satisfacción}}{\text{Número de clientes encuestados}} \times 100$$

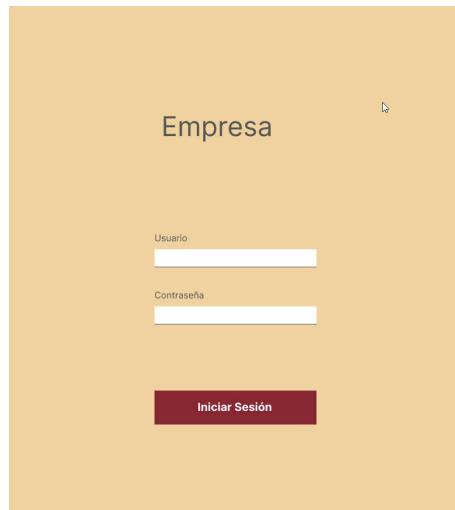
### **2.1.2. Importancia de los indicadores**

Los indicadores de rendimiento juegan un papel crucial en la evaluación del éxito del sistema implementado. A través de estos indicadores, podemos medir aspectos clave como la eficiencia, la precisión y la velocidad en la operación del sistema de ventas.

- a. **Monitoreo del desempeño:** Los indicadores permiten verificar que el sistema funcione correctamente, especialmente en lo que se refiere a la automatización y el control de inventarios.
- b. **Optimización de procesos:** Al medir el tiempo promedio de registro de ventas, podemos detectar oportunidades para mejorar el flujo de trabajo, lo que ayudará a la empresa a operar de manera más eficiente.
- c. **Toma de decisiones informadas:** Con un buen análisis de los indicadores, la empresa puede hacer ajustes estratégicos, ya sea en términos de personal o de optimización del sistema.

## **CAPÍTULO 3 - DESARROLLO DE LA SOLUCIÓN**

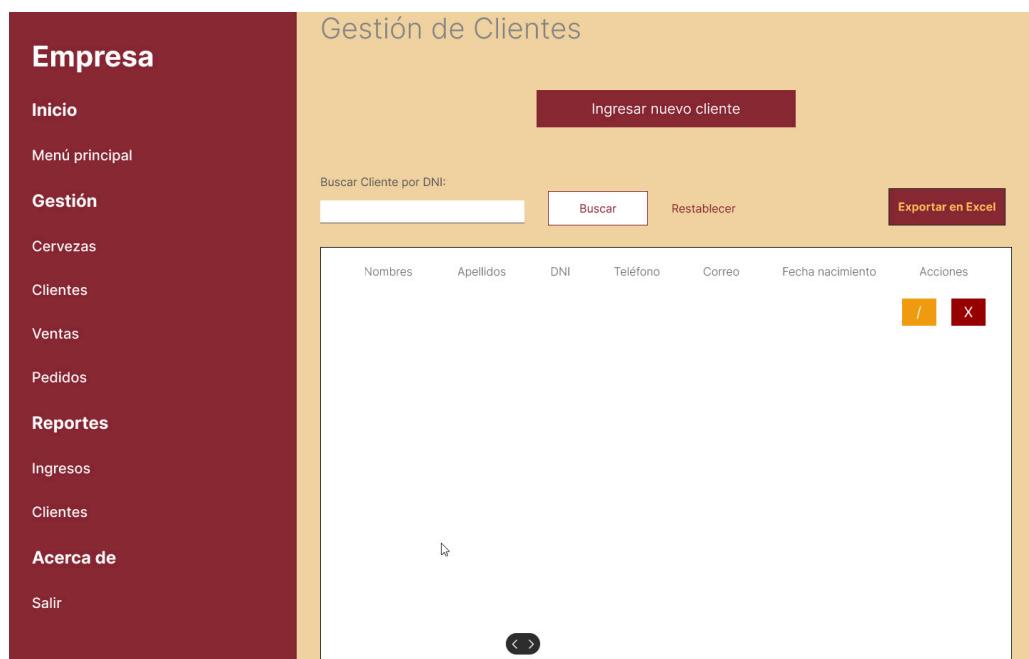
### **3.1. Prototipos**



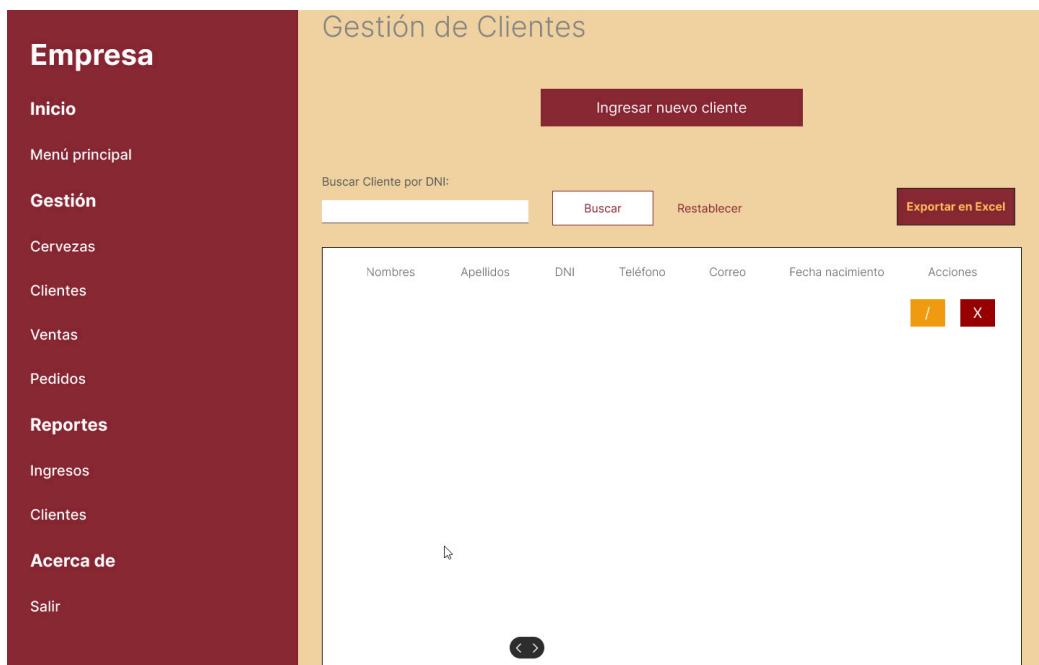
**Figura 5:** Prototipo de *Login*



**Figura 6:** Prototipo de *Inicio*



**Figura 7:** Prototipo de *Clientes*



**Figura 8:** Prototipo registro de producto

A wireframe prototype for entering a new beer. At the top left is a 'Volver' link. The main title is 'Ingresar nueva cerveza'. The form consists of several input fields arranged in pairs: 'Nombre' and 'Volumen', 'Marca' and 'Tipo', and 'Embase' and 'Precio Unitario'. Each pair has a horizontal input field. Below the fields is a dark red 'Guardar' button. At the bottom center is a small circular navigation icon with arrows.

**Figura 9:** Prototipo registro de nuevo producto

Volver

### Ingresar nuevo Cliente

**Información Personal**

Nombres	Apellidos
<input type="text"/>	<input type="text"/>
DNI	Fecha de nacimiento dd/mm/yyyy
<input type="text"/>	<input type="text"/> / <input type="text"/> / <input type="text"/>

**Información de Contacto**

Teléfono	Correo
<input type="text"/>	<input type="text"/>

**Guardar**

< >

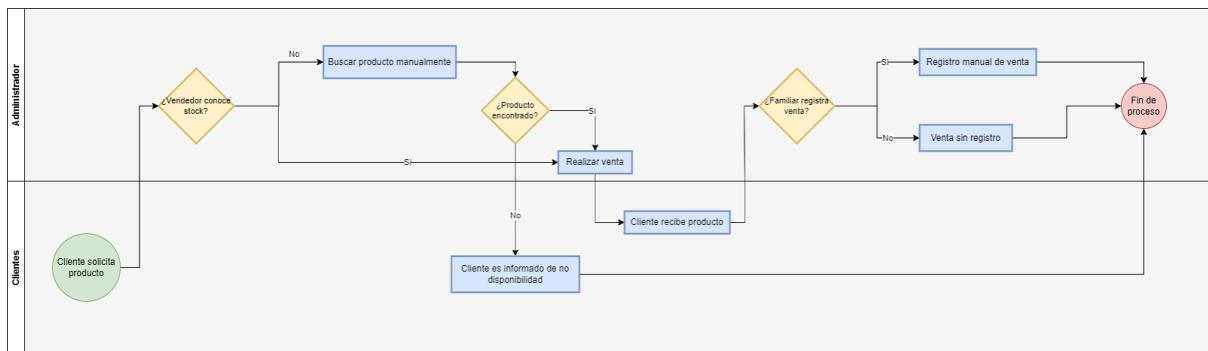
**Figura 10:** Prototipo de registro de nuevo cliente

### 3.2. Modelo de proceso de negocios

#### 3.2.1. Propuesta de valor

Nuestro sistema de ventas permitirá mejorar la gestión de la distribuidora de cerveza, logrando optimizar la operatividad de la empresa así como también aumentar y mejorar la percepción sobre la satisfacción de nuestros clientes. A través de una plataforma intuitiva y fácil de usar, podrás:

- **Gestionar el inventario eficientemente:** lo que nos permitirá realizar un seguimiento de los productos, evitando quedar sin stock y asimismo minimizando pérdidas.
- **Ofrecer un mejor servicio a los clientes:** lo que nos permitirá garantizar la disponibilidad de los productos, entregas a tiempo y una atención personalizada.
- **Aumentar la rentabilidad:** Reduce costos operativos, mejora la eficiencia y aumenta tus ventas.



**Figura 11:** Prototipo de *registro de nuevo cliente*

### 3.2.2. Diagrama de clases

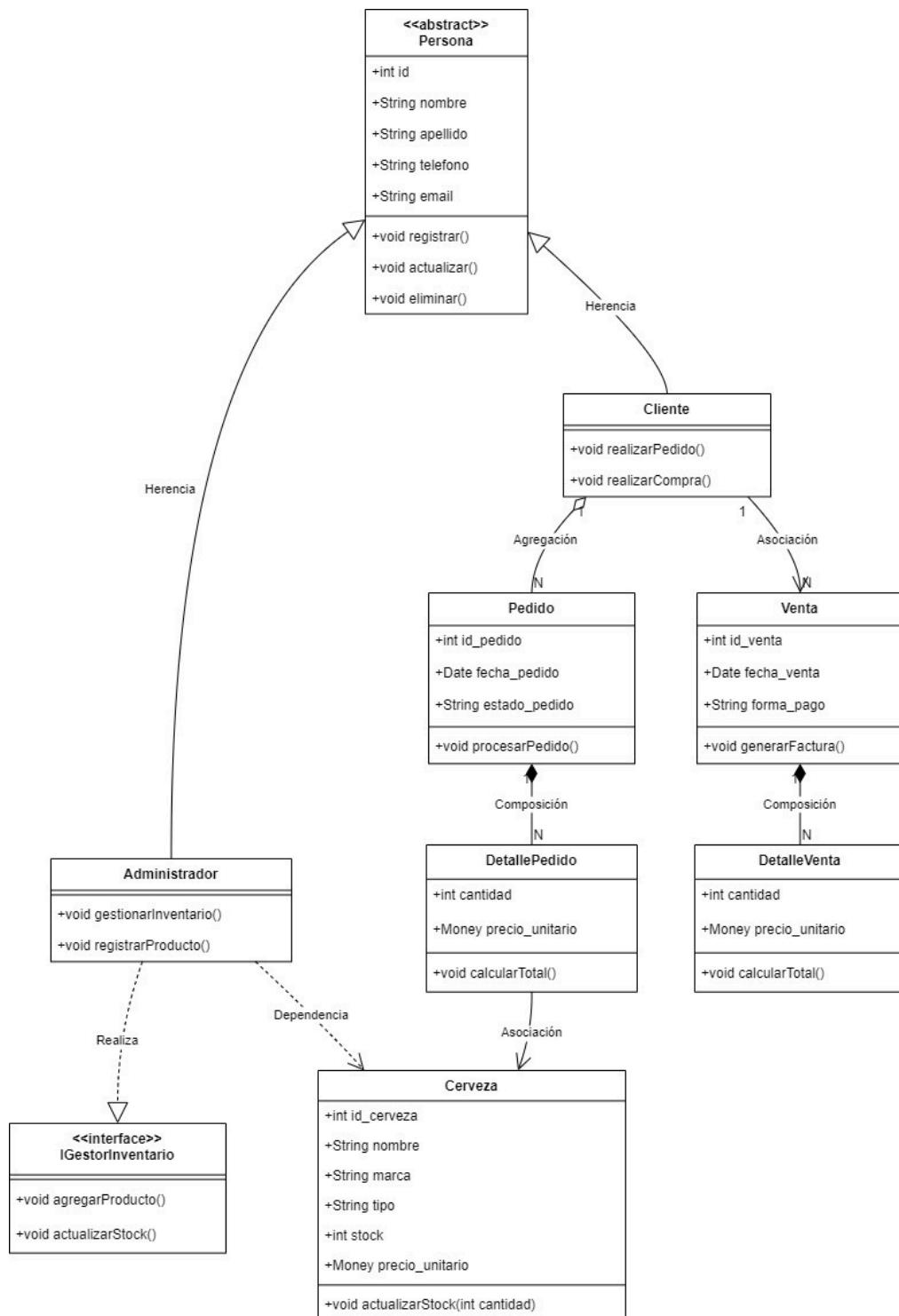


Figura 12: Diagrama de clases

### 3.2.3. Diagrama entidad - relación

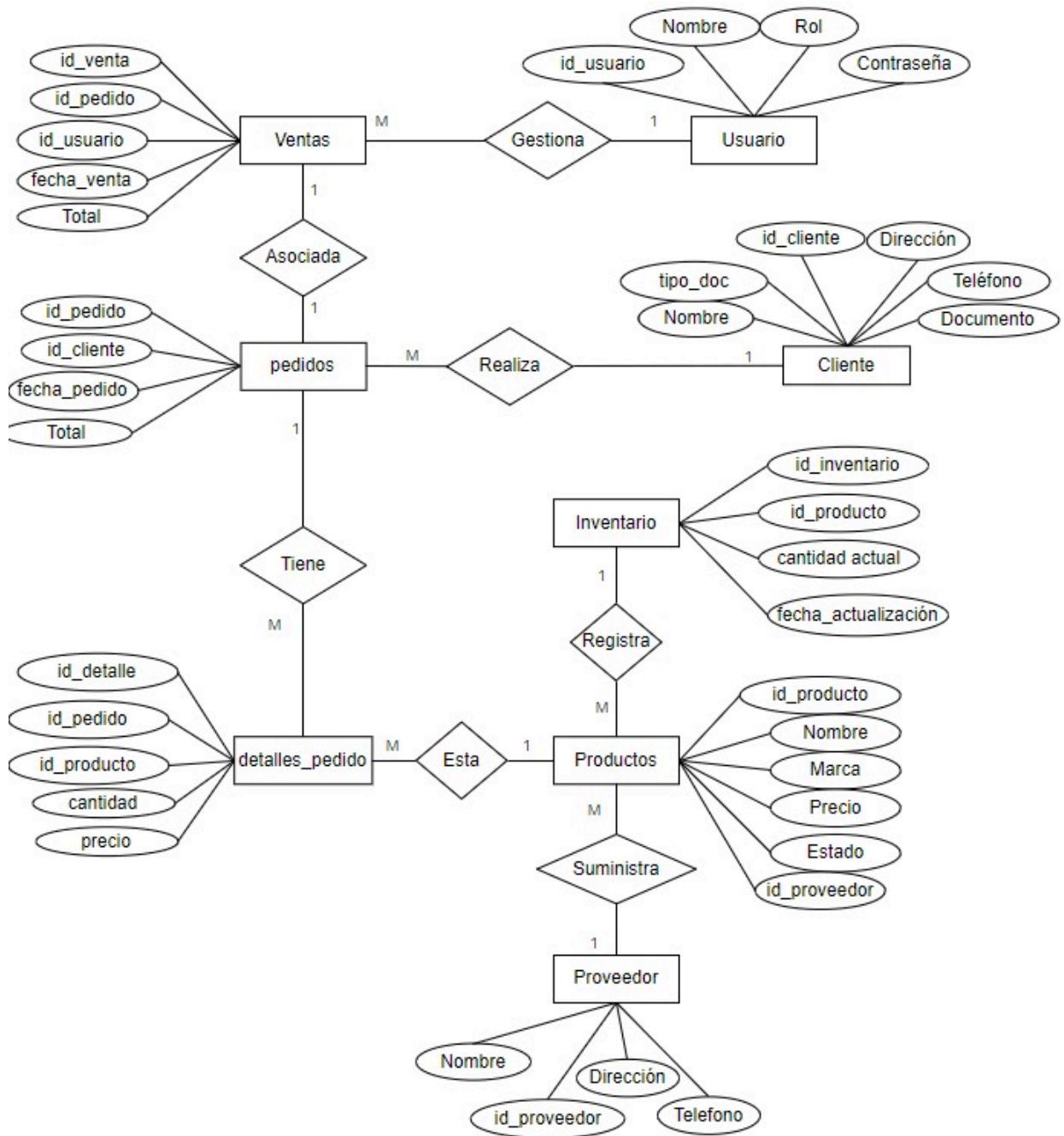
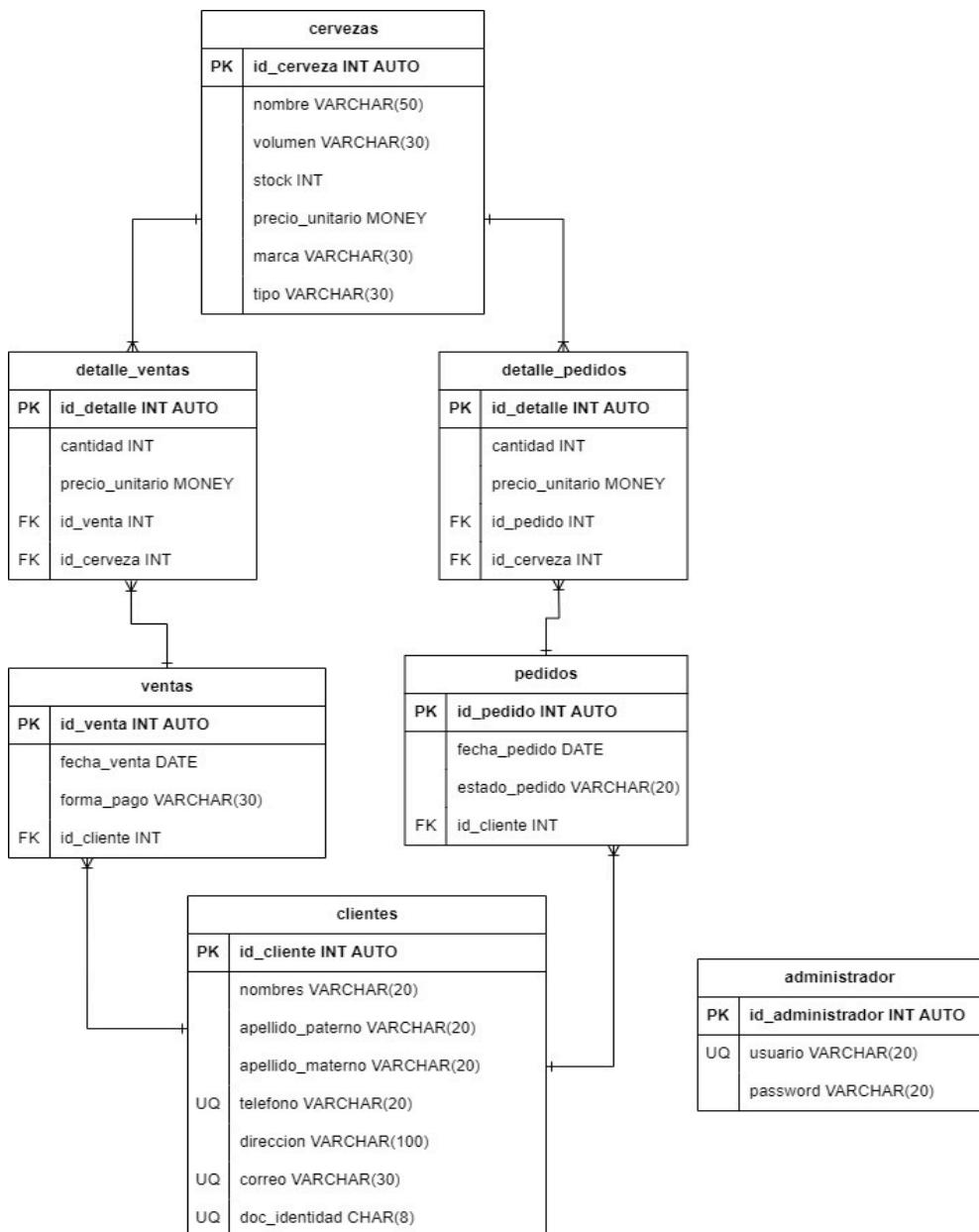


Figura 13: Diagrama entidad relación lógico



**Figura 14: Diagrama entidad relación físico**

### 3.2.4. Diccionario de datos

Para poder elaborar el diccionario de datos se debe definir los puntos referentes al nombre, tipo de dato, longitud, formato y descripción de cada atributo como la presentación del producto relacionado a los mililitros de las bebida que se encuentra en el punto de metodología.

Nombre de la tabla		Descripción		
Producto		Almacena el nombre de la cerveza		
Tabla de Producto				
	Nombre de la columna	Descripción	Tipo de dato	NULL
PK	id_cerveza	Identificador único de la marca del producto	int	no
	nombre	Nombre del producto	varchar(50)	no
	volumen	Volumen de la cerveza	varchar(30)	no
	stock	Cantidad disponible en inventario	int	no
	precio_unitario	Tipo de presentación de envasado de la cerveza	money	no
	marca	Marca de la cerveza	varchar(30)	no
	tipo	Tipo de cerveza	varchar(30)	no

**Tabla 19: Productos**

Nombre de la tabla		Descripción		
detalle_ventas		Detalles de la compra del producto		
Tabla de Distribución				
	Nombre de la columna	Descripción	Tipo de dato	NULL
PK	id_detalle	Identificador único de la distribución	int	no
	cantidad	Cantidad de cervezas vendidas	int	no
	precio_unitario	Precio unitario al momento de la venta	money	no
FK	id_venta	Identificador de la venta	int	
FK	id_cerveza	Identificador de la cerveza vendida	int	

**Tabla 20: Detalle de venta**

Nombre de la tabla		Descripción		
ventas		Almacena información sobre las ventas de cervezas en distintos distritos		
<b>Tabla de Ventas</b>				
	Nombre de la columna	Descripción	Tipo de dato	NULL
PK	id_venta	Identificador único de la venta	int	no
	fecha_venta	Fecha en la que se realiza la venta del producto	date	no
	forma_pago	La modalidad de pago con la que se va a pagar el producto	varchar(30)	no
FK	id_cliente	Identificador del cliente	int	

**Tabla 21: Venta**

Nombre de la tabla		Descripción		
detalles_pedidos		Detalles de los pedidos hechos por los clientes		
<b>Tabla Detalle Pedidos</b>				
	Nombre de la columna	Descripción	Tipo de dato	NULL
PK	id_detalle	Identificador único del detalle del pedido	int	no
	cantidad	Cantidad de cervezas pedidas	int	no
	precio_unitario	Precio unitario al momento del pedido	money	no
FK	id_pedido	Identificador del pedido	int	
FK	id_cervezas	Identificador de la cerveza en el pedido	int	

**Tabla 22: Detalle de pedido**

Nombre de la tabla		Descripción		
pedidos		Pedidos hechos por los clientes		
<b>Tabla de Pedidos</b>				
	Nombre de la columna	Descripción	Tipo de dato	NULL
PK	id_pedido	Identificador único del detalle del pedido	int	no
	fecha_pedido	Fecha en la que se realizó el pedido	date	no
FK	estado_pedido	Estado del pedido (ej. 'pendiente', 'completado')	varchar(20)	no
FK	id_cliente	Identificador del cliente	int	

**Tabla 23: Pedidos**

Nombre de la tabla		Descripción		
cliente		Almacena el nombre del cliente		
<b>Tabla de Cliente</b>				
	Nombre de la columna	Descripción	Tipo de dato	NULL
PK	id_cliente	Identificador único del cliente	int	no
	nombres	Nombres del cliente	varchar(20)	no
	apellido_paterno	Apellido paterno del cliente	varchar(20)	no
	apellido_materno	Apellido materno del cliente	varchar(20)	no
UQ	telefono	Número telefónico del cliente	varchar(20)	si
	direccion	Dirección donde reside el cliente	varchar(100)	si
UQ	correo	Correo electrónico del cliente	varchar(30)	si
UQ	doc_identidad	Documento de identidad del cliente	varchar(8)	no

**Tabla 24: Cliente**

Nombre de la tabla		Descripción		
usuario		Contiene la información de los administradores del sistema, con su nombre de usuario y contraseña para gestionar el acceso y los permisos administrativos.		
<b>Tabla de Usuario</b>				
	Nombre de la columna	Descripción	Tipo de dato	NULL
PK	id_administrador	Identificador único del administrador	int	no
UQ	usuario	Nombre del usuario	varchar(20)	no
	password	Contraseña del usuario que permite acceso al sistema	varchar(20)	no

**Tabla 25: Usuario**

## BIBLIOGRAFÍA

1. Auromecar. (2020, julio 28). *Sistema de Ventas para Licorería*.  
<https://www.youtube.com/watch?v=C1Pwoc0Dmu4>
2. Bautista, L., Blas, A., & Hidalgo, I. (2023). *Sistema de punto de venta y control de inventario de la bodega J'ABDIEL en la provincia de Jauja*.  
[https://repositorio.continental.edu.pe/bitstream/20.500.12394/13306/1/IV\\_FIN\\_103\\_TE\\_Bautista\\_Blas\\_Hidalgo\\_2023.pdf](https://repositorio.continental.edu.pe/bitstream/20.500.12394/13306/1/IV_FIN_103_TE_Bautista_Blas_Hidalgo_2023.pdf)
3. Correa, J. (2023). *Desarrollo de sistema de gestión de venta de licores para la licorería la nenita : Backend*.  
<http://bibdigital.epn.edu.ec/handle/15000/24971>
4. Creately. (n.d.). *Class Diagram for Point of Sale System (POS)*. Retrieved 2024, from  
<https://creately.com/diagram/example/iliw20n31/class-diagram-for-point-of-sale-system-pos>
5. GuayaHack. (2023, agosto 21). *¿Cuál es la diferencia entre un Diagrama de Clases y un Diagrama de Entidad Relación?*  
<https://guayahack.co/community/member/davidadarme/reto-diferencias-diagrama-clases-entidad-relacion/>
6. IBM. (2021, marzo 8). *Modelos lógicos de datos*. InfoSphere Data Architect 9.1.2.  
<https://www.ibm.com/docs/es/ida/9.1.2?topic=modeling-logical-data-models>
7. IBM. (2021, marzo 16). *Modelos físicos de datos*. Rational Application Developer for WebSphere Software 9.6.1.  
<https://www.ibm.com/docs/es/radfw/9.6.1?topic=modeling-physical-data-models>

8. Lucidchart. (n.d.). *¿Qué es un diagrama entidad-relación?* Lucidchart. Retrieved 2024, from <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>
9. Peña, L. (2021). *Sistema informático para venta de licores*. Tecnológico de Antioquia, Institución Universitaria.
- <https://dspace.tdea.edu.co/handle/tdea/2598>
10. Quispe, T., & Calderón, E. (2020). *Implementación de un sistema de control interno en la gestión de inventario en la empresa constructora Carmencita E.I.R.L.* 2019.
- <https://repositorio.cientifica.edu.pe/bitstream/handle/20.500.12805/1419/TB-Quispe%20T-Calder%c3%b3n%20E-Ext.pdf?sequence=1&isAllowed=y>
11. Sánchez, H., Romero, C., & Mejía, K. (2018). *Manual de términos en investigación científica, tecnológica y humanística*.
- <https://www.urp.edu.pe/pdf/id/13350/n/libro-manual-de-terminos-eninvestigacion.pdf>
12. Stack Overflow. (n.d.). *Entity Relationship Diagram Sales*. Retrieved 2024, from <https://stackoverflow.com/questions/21022216/entity-relationship-diagram-sales>