



Universidad  
Tecnológica  
del Perú

**Facultad de Ingeniería**

**Carrera Profesional de Ingeniería de  
Sistemas de Informática**

**Gestión de inventario para tienda mayorista  
de ropa juvenil "Top Modas"**

Estudiantes:

Carbajal Añanca, Melany Daniela U22222750

Kim Negrillo, Lorena U22229186

Montesinos Poma, Britney Daphne U21207912

Rodriguez Martinez, Renzo Luciano U22239115

**Lima - Perú**

**2024**

## INDICE

INTRODUCCIÓN.....	6
CAPÍTULO 1 - ASPECTOS GENERALES .....	7
1.1 Descripción del Problema .....	7
1.2 Definición de Objetivos.....	10
1.2.1 Objetivo General .....	10
1.2.2 Objetivos Específicos .....	10
1.3 Alcances y Limitaciones .....	10
1.3.1 Alcances .....	13
1.3.2 Limitaciones .....	14
1.4 Justificación .....	14
1.5 Estado del Arte .....	15
CAPITULO 02 – MARCO TEORICO.....	18
2.1 Marco teórico de programación orientada a objetos .....	18
2.2 Marco teórico de las formulas y/o el sistema .....	43
CAPITULO 03 – DESARROLLO DE LA SOLUCION .....	46
3.1 Prototipos.....	46
3.2 Modelo de procesos de negocios.....	58
3.3 Diagrama de clases .....	58
3.4 Diagrama de entidad relación.....	59
REFERENCIAS .....	62
ANEXOS.....	64

## INDICE DE FIGURAS

Figura 1 Diagrama de árbol de la tienda “Top Modas” .....	9
Figura 2 Módulo de administración de artículos del comercial maquinas Hidalgo ....	15
Figura 3 Interfaz de gráfica de facturación electrónica de Livecor .....	16
Figura 4 Interfaz de módulo de diseño de producción de confecciones y diseños Aguilera .....	17
Figura 5 Patrón Formulario de Autenticación “Top Modas” .....	23
Figura 6 Patrón Restablecimiento de Contraseña “Top Modas” .....	24
Figura 7 Patrón Notificaciones Modales “Top Modas” .....	24
Figura 8 Patrón Formulario de Contacto “Top Modas” .....	25
Figura 9 Patrón Validación de Entrada “Top Modas” .....	26
Figura 10 Patrón Vista Previa de Imagen “Top Modas” .....	26
Figura 11 Patrón Búsqueda y Filtro “Top Modas” .....	27
Figura 12 Patrón de Carrito de Compras “Top Modas” .....	27
Figura 13 Patrón de Selección de Archivos “Top Modas” .....	28
Figura 14 Patrón de Calendario .....	29
Figura 15 Modelo-Vista-Controlador (MVC).....	29
Figura 16 Patrón Singleton.....	30
Figura 17 Patrón Repository .....	31
Figura 18 Inicio de sesión cajero.....	46
Figura 19 Sección contactar administrador.....	47
Figura 20 Caja de verificación para la sección contactar administrador .....	47
Figura 21 Sección ¿Olvidaste tu contraseña? .....	48
Figura 22 Caja de verificación para restablecer contraseña.....	48
Figura 23 Cajero sección Dashboard.....	49
Figura 24 Cajero sección Caja .....	49
Figura 25 Boleta electrónica de la sección Caja .....	50
Figura 26 Cajero sección Inventario.....	51
Figura 27 Cajero sección Clientes .....	51
Figura 28 Cajero sección Reportes.....	52
Figura 29 Caja de verificación para cerrar sesión del cajero.....	52
Figura 30 Inicio sesión de Administrador .....	53
Figura 31 Administrador sección Dashboard.....	53
Figura 32 Administrador sección Ventas .....	54
Figura 33 Administrador sección Inventario.....	54

Figura 34 Administrador sección Clientes.....	55
Figura 35 Administrador sección Cajeros .....	55
Figura 36 Administrador sección Cierre de caja .....	56
Figura 37 Administrador sección Análisis .....	56
Figura 38 Administrador sección Configuración .....	57
Figura 39 Caja de verificación para cerrar sesión del Administrador .....	57
Figura 40 Modelo de negocios Bizagui .....	58
Figura 41 Diagrama de clase Top Modas .....	58
Figura 42 Diagrama de entidad relación de Top Modas.....	59

## INDICE DE TABLAS

TABLA 01: Requerimiento Funcional: Registro de Inventario Inicial .....	11
TABLA 02: Requerimientos Funcionales: Actualización de Inventario en Tiempo Real .....	11
TABLA 03: Requerimiento Funcional: Generar Reportes de Inventario .....	11
TABLA 04: Requerimiento Funcional: Gestión de Proveedores .....	12
TABLA 05: Requerimiento Funcional: Control de Acceso .....	12
TABLA 06: Requerimientos Funcionales: Gestión y Notificaciones de Alertas .....	12
TABLA 07: Requerimientos No Funcionales: Disponibilidad del Sistema .....	12
TABLA 08: Requerimientos No Funcionales: Usabilidad y Diseño del Sistema .....	13
TABLA 09: Requerimientos No Funcionales: Estructura Modular y Mantenibilidad .....	13
TABLA 10: Requerimientos No Funcionales: Plazos y Entregables .....	13
TABLA 11: Comparativa de metodologías .....	20
TABLA 12: Cliente .....	59
TABLA 13: Usuario .....	60
TABLA 14: Detalle_Pedido .....	60
TABLA 15: Pedido .....	60
TABLA 16: Categoría .....	60
TABLA 17: Prenda .....	61
TABLA 18: Pago .....	61

## INTRODUCCIÓN

En la era de la tecnología y la digitalización, las empresas enfrentan un imperativo creciente de adaptarse a nuevas herramientas y metodologías que les permitan mejorar su eficiencia operativa y competitividad. Sin embargo, en entornos altamente dinámicos como el emporio comercial de Gamarra, muchas tiendas aún no han dado el salto hacia la digitalización, enfrentando así desafíos significativos en su operación diaria. Este es el caso de Top Modas, una tienda mayorista de ropa para mujeres jóvenes que, a pesar de su éxito y alto volumen de transacciones, no cuenta con un sistema de gestión de inventario adecuado.

La idea de este proyecto surge al observar las dificultades que enfrenta Top Modas para gestionar su inventario de manera eficiente. La falta de un sistema de seguimiento adecuado ya sea digital o físico, ha llevado a la empresa a operar de manera menos eficiente, generando problemas como errores en el registro de stock y la incapacidad de responder ágilmente a las demandas del mercado. Estas deficiencias no solo afectan la operatividad diaria, sino que también comprometen la satisfacción del cliente y, en última instancia, la rentabilidad de la tienda.

El objetivo principal de este proyecto es desarrollar un sistema de gestión de inventario que permita a Top Modas modernizar su operación, optimizando el uso de recursos y mejorando la precisión en el manejo del stock. Este proyecto es crucial en el contexto actual, donde la digitalización es clave para mantener la competitividad en un mercado tan exigente como el de Gamarra.

El proyecto no solo responde a una necesidad urgente de la empresa, sino que también se inscribe en la tendencia global hacia la automatización y digitalización de procesos, un aspecto fundamental para la sostenibilidad y el crecimiento a largo plazo de cualquier negocio en la industria de la moda.

La importancia de este proyecto radica en su capacidad para transformar la gestión de inventarios de Top Modas, alineándola con las mejores prácticas de la industria y dotándola de herramientas que le permitan adaptarse rápidamente a los cambios del mercado, asegurando así su relevancia y competitividad a largo plazo.

## **CAPÍTULO 1 - ASPECTOS GENERALES**

### **1.1 Descripción del Problema**

En el contexto actual, donde la tecnología y la digitalización juegan un papel crucial en la eficiencia operativa de las empresas, muchas aún enfrentan desafíos significativos por no adaptarse a estos cambios. Este es el caso de Top Modas, una tienda de ropa juvenil para mujeres ubicada en el emporio comercial de Gamarra, conocido por su vasta oferta de productos textiles. Top Modas opera dos tipos de tiendas: una mayorista y otra minorista. El enfoque de este proyecto se centra en la tienda mayorista, que maneja un volumen considerable de transacciones debido a su diversa clientela.

Actualmente, Top Modas no cuenta con un sistema de gestión de inventario; la empresa no utiliza ningún software y tiene mal manejo de herramientas físicas (como registros en papel) para el seguimiento de su inventario. La razón principal de esta carencia es la percepción de que llevar un inventario manual consume demasiado tiempo, especialmente dado el alto volumen de clientes y transacciones. Además, la falta de herramientas digitales adecuadas ha impedido que la empresa implemente un sistema eficiente para esta tarea esencial.

Esta ausencia de un sistema estructurado no solo complica la gestión del stock, sino que también incrementa el riesgo de errores en el registro de inventario, lo que puede llevar a problemas como la sobreestimación o subestimación de existencias. Asimismo, sin un sistema adecuado, es difícil responder de manera ágil a las demandas del mercado, lo que podría afectar negativamente la satisfacción del cliente y, por ende, la rentabilidad de la tienda.

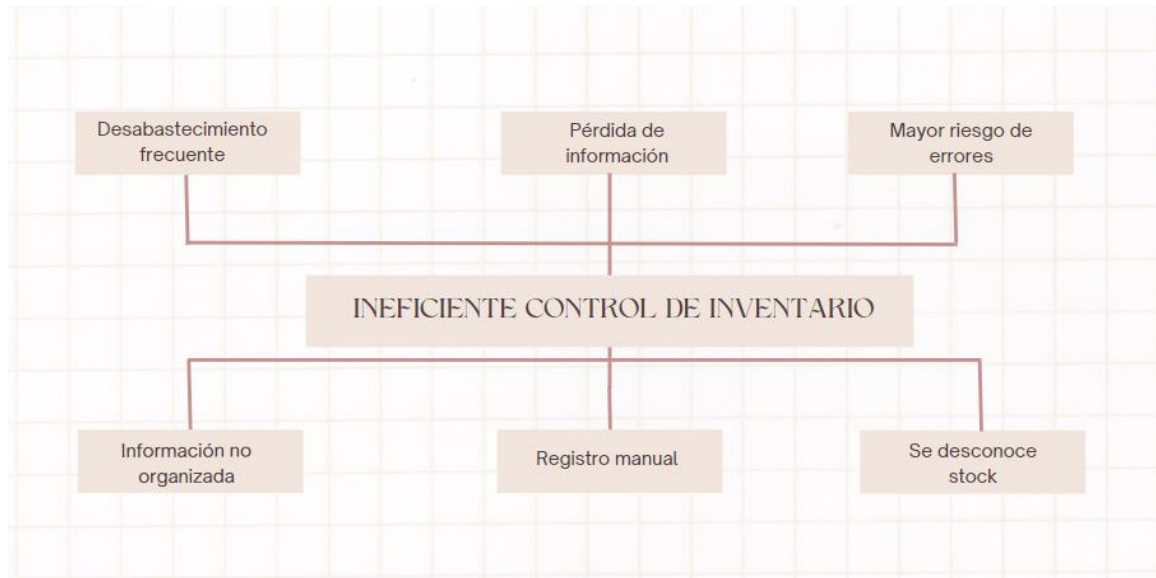
De acuerdo con el estudio destacado en *Innovaciones Clave en la Industria de la Moda Digital*, la digitalización de la gestión de inventarios mediante tecnologías avanzadas, como el Internet de las Cosas (IoT), es fundamental para mejorar la eficiencia operativa. Esta tecnología no solo permite una visibilidad en tiempo real del stock, sino que también optimiza la gestión de inventarios, mejora el uso de los recursos y potencia la capacidad de tomar decisiones basadas en datos precisos y actualizados, lo que es esencial para mantener la competitividad en la dinámica industria de la moda.

En el estado actual del negocio, los pedidos llegan al vendedor, el cual anota los detalles y es quien se encarga de manera manual de verificar la disponibilidad del stock en tienda. En caso no haya stock, el vendedor irá a avisarle al almacenero, quien realizará la verificación de stock en almacén, en caso aún así no haya stock, se le avisará al cliente de ello y la venta terminará como nula. En caso haya stock en la tienda, o en su defecto, no lo haya y solo se encuentren existencias en el almacén, el vendedor carga la mercadería al cliente, tras lo cual, se hará un recuento de la mercadería y verificará la conformidad del cliente con el pedido. En caso no haya conformidad, se revisará el pedido nuevamente, dando la posibilidad al cliente de hacer cambios en él, como añadir o quitar mercadería. Si esto ocurre, se volverá a verificar el stock manualmente hasta que esté listo. Cuando exista una conformidad del cliente con la mercadería solicitada, se procede al pago, se verifica la conformidad de éste, en caso no la haya, se vuelve a intentar el pago y cuando todo esté correcto, la venta se habrá concretado.

Por lo tanto, la implementación de un gestor de inventario digital en Top Modas es crucial para modernizar y mejorar la gestión del inventario en la tienda mayorista, asegurando así una operación más fluida y rentable que pueda adaptarse rápidamente a las dinámicas cambiantes del mercado de Gamarra.



## Diagrama de Árbol



**Figura 1 Diagrama de árbol de la tienda “Top Modas”**

*Fuente: Elaboración propia*

## Indicadores de Gestión

### 1. Nivel de Inventario Disponible

Se trata de la proporción del inventario que está disponible para la venta en cualquier momento.

Fórmula: Nivel de Inventario Disponible =  $(\text{Unidades disponibles} / \text{Unidades totales}) \times 100$

El objetivo es asegurar que siempre haya suficiente inventario disponible para satisfacer la demanda.

### 2. Tiempo de Reposición

Es el tiempo que tarda reponer un producto desde que se hace el pedido hasta que está disponible en stock.

Fórmulas:

Tiempo de Reposición Individual = Fecha de Disponibilidad - Fecha del Pedido

Tiempo de Reposición Promedio =  $\Sigma(\text{Tiempo de reposición individual})/\text{Número total de pedidos}$

Tiene como objetivo minimizar el tiempo de reposición para evitar la falta de stock.

### **3. Índice de Pérdida**

Consiste en medir las pérdidas de inventario debido a factores como robos, deterioro, errores, entre otros.

Fórmula: Índice de Pérdida de Inventario =  $(\text{Pérdida de inventario} / \text{Inventario total}) * 100$

Su objetivo es reducir el índice de pérdida de inventario a través de mejores controles y gestión.

#### **1.2 Definición de Objetivos**

##### **1.2.1 Objetivo General**

Mejorar la eficiencia operativa y la precisión en el manejo del stock de la tienda mayorista juvenil para mujeres Top Modas a través del desarrollo de un sistema de gestión de inventario.

##### **1.2.2 Objetivos Específicos**

- + Implementar una base de datos para el registro y seguimiento del inventario.
- + Desarrollar una interfaz intuitiva para el registro y actualización de inventario.
- + Implementar funcionalidades de búsqueda y filtrado avanzadas.

#### **1.3 Alcances y Limitaciones**

La manera en que se definen los alcances y limitaciones se dará mediante requerimientos funcionales y no funcionales.

Para los requerimientos funcionales y no funcionales a continuación, se definieron tres tipos de prioridades los cuales indican características específicas respecto a cuanto se necesita con cumplir con un requerimiento funcional.

- Alta: Se requiere con urgencia y su falta afecta de manera directa al proceso del negocio.
- Media: Su incumplimiento afectará al proceso del negocio luego de un tiempo.
- Baja: Su incumplimiento podría alentar levemente el proceso del negocio.

#### Requerimientos Funcionales:

**TABLA 01: Requerimiento Funcional: Registro de Inventario Inicial**

Número de requerimiento		RF01	
Nombre		Registrar inventario inicial	
Usuario	Administrador	Prioridad	Alta
El sistema debe permitir el registro inicial de todos los productos en inventario, incluyendo detalles como código del producto, descripción, cantidad disponible y ubicación en el almacén.			

*Fuente: Elaboración propia*

**TABLA 02: Requerimientos Funcionales: Actualización de Inventario en Tiempo Real**

Número de requerimiento		RF02	
Nombre		Actualizar Inventario	
Usuario	Administrador	Prioridad	Alta
El sistema debe permitir la actualización en tiempo real del inventario con cada venta, devolución o reposición de productos.			

*Fuente: Elaboración propia*

**TABLA 03: Requerimiento Funcional: Generar Reportes de Inventario**

Número de requerimiento		RF03	
Nombre		Generar reportes de inventario	
Usuario	Administrador	Prioridad	Alta
El sistema debe generar reportes periódicos de inventario, indicando las existencias actuales, productos más vendidos, productos con baja rotación, y alertas de productos próximos a agotarse.			

*Fuente: Elaboración propia*

**TABLA 04: Requerimiento Funcional: Gestión de Proveedores**

Número de requerimiento		RF04	
Nombre		Gestión de proveedores	
Usuario	Administrador	Prioridad	Alta
El sistema debe gestionar la información de los proveedores, permitiendo registrar, modificar y consultar datos como nombre, contacto, productos suministrados y condiciones de compra.			

*Fuente: Elaboración propia*

**TABLA 05: Requerimiento Funcional: Control de Acceso**

Número de requerimiento		RF05	
Nombre		Control de Acceso	
Usuario	Vendedor/Administrador	Prioridad	Alta
El sistema debe implementar control de acceso para asegurar que solo el personal autorizado pueda realizar modificaciones en el inventario y acceder a la información confidencial.			

*Fuente: Elaboración propia*

**TABLA 06: Requerimientos Funcionales: Gestión y Notificaciones de Alertas**

Número de requerimiento		RF06	
Nombre		Gestión de alertas	
Usuario	Vendedor	Prioridad	Alta
El sistema debe enviar alertas automáticas por correo electrónico o notificaciones en la aplicación cuando el stock de un producto alcance el nivel mínimo definido o cuando se detecten discrepancias en el inventario.			

*Fuente: Elaboración propia*

#### **Requerimientos no funcionales:**

**TABLA 07: Requerimientos No Funcionales: Disponibilidad del Sistema**

Número de Requerimiento	RNF01
Nombre	Disponibilidad del software
Prioridad	Alta
El software debe estar disponible y sin cierres inesperados durante las horas operativas de la tienda.	

*Fuente: Elaboración propia*

**TABLA 08: Requerimientos No Funcionales: Usabilidad y Diseño del Sistema**

Número de Requerimiento	RNF02
Nombre	Calidad de diseño
Prioridad	Baja
Se debe aplicar un diseño intuitivo y simple de comprender para la facilidad de los usuarios (vendedor, administrador)	

*Fuente: Elaboración propia*

**TABLA 09: Requerimientos No Funcionales: Estructura Modular y Mantenibilidad**

Número de Requerimiento	RNF03
Nombre	Modularidad
Prioridad	Baja
El código debe estar estructurado modularmente para facilitar su manenimiento y actualización	

*Fuente: Elaboración propia*

**TABLA 10: Requerimientos No Funcionales: Plazos y Entregables**

Número de Requerimiento	RNF04
Nombre	Tiempo de entrega
Prioridad	Media
Se requiere de entregar el sistema en un máximo plazo de entrega de 16 semanas	

*Fuente: Elaboración propia*

### 1.3.1 Alcances

- Se debe poder realizar un registro inicial de todos los productos en el inventario.
- El sistema se debe actualizar en tiempo real según las operaciones realizadas durante el trabajo.
- El sistema deberá poder realizar reportes de métricas de la tienda.
- El sistema será capaz de gestionar datos de los distintos proveedores de la tienda.
- Se deberá tener control de acceso en el sistema para garantizar su seguridad

- El sistema enviará alertas automáticas en caso el stock de los productos llegue esté cerca de agotarse o existan discrepancias de información.

### **1.3.2 Limitaciones**

- Únicamente se gestionarán los inventarios con este programa, otras áreas de la tienda como la de personal, tesorería o contabilidad, podrían requerir un software con diferentes capacidades.
- El software únicamente funcionará para las necesidades de esta tienda en específico, en caso se quiera usar en otra, serán necesarias adaptaciones o modificaciones para esta nueva tienda.
- El sistema se limitará a funcionar para el único local que tiene la tienda, en caso se tengan más locales, se tendrán que implementar cambios a la base de datos y al programa para que gestione por separado lo relacionado con distintos locales y la jerarquía de usuarios para procurar la seguridad de la información.
- Este proyecto tiene un límite de tiempo de desarrollo de 16 semanas.
- Se cuenta con un personal de 4 personas para el desarrollo de este proyecto.
- El presupuesto de este proyecto tendrá un límite determinado por la capacidad de inversión que provea la tienda que requiere del sistema.

### **1.4 Justificación**

El proyecto propuesto busca solucionar estas problemáticas mediante la digitalización del proceso de gestión de inventarios, permitiendo un control y una mayor agilidad en la toma de decisiones. Debido a que al no contar con un sistema los ha llevado a ineficiencias, errores en el control de stock y dificultades para adaptarse a las demandas del mercado, comprometiendo la satisfacción del cliente y la rentabilidad. La implementación de este sistema no solo podrá alinear a Top Modas con las tendencias de la industria, sino que también le proporcionará una ventaja competitiva esencial para mantenerse relevante en un mercado tan exigente como el de Gamarra. Al mejorar la eficiencia operativa, reducir el riesgo de errores y optimizar el uso de recursos, el sistema contribuirá directamente a la sostenibilidad y crecimiento a largo plazo de la empresa.

## 1.5 Estado del Arte

### Desarrollo e implementación de aplicación web para el control de inventario de local comercial máquinas Hidalgo

En este proyecto se utiliza **SCRUM** como metodología de desarrollo ágil donde se dividió el proyecto en sprints. Este proyecto fue desarrollado debido a la falta de un sistema de gestión adecuado, lo que generaba ineficiencias y errores en la gestión stock, el objetivo que se dio fue crear una aplicación web diseñada para mejorar el control de inventarios en las cuales se usa programas principales como: MySQL como gestor de base de datos, PHP para desarrollar la aplicación web junto a HTML y CSS para la interfaz de usuario. Nuestro proyecto se inspira en soluciones previamente desarrolladas, donde se desea usar el MySQL para la gestión de bases de datos, Consideramos que este programa es fundamental para garantizar una integración fluida y un manejo eficiente del sistema de inventario digital que proponemos implementar en Top Modas.

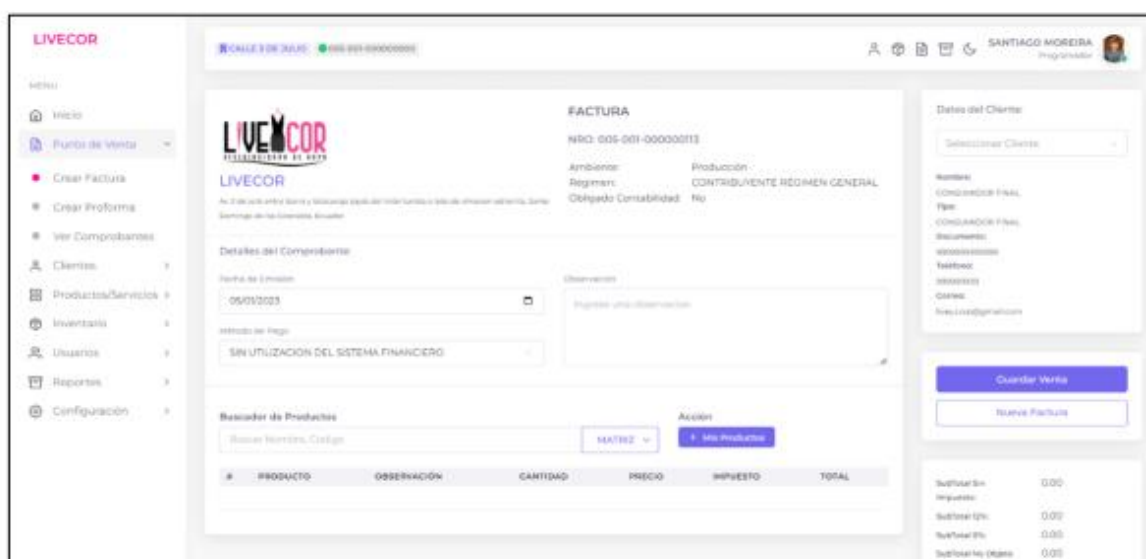
Nombre	Descripción	Precio Venta	Tipo	Características	Acciones
MÁQUINA COSEDORA INDUSTRIAL		\$10.00	Máquina industrial		
MÁQUINA COSEDORA PLANA		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		
Máquina cosedora		\$10.00	Máquina industrial		

**Figura 2 Módulo de administración de artículos del comercial maquinass Hidalgo**

(Lucas Vega,2017)

## Desarrollo de un sistema web de gestión de inventario y punto de venta para la distribuidora de ropa Livecor ubicada en la ciudad de Santo Domingo

El siguiente proyecto tuvo como objetivo el desarrollo de un sistema web de gestión de inventario y punto de venta con facturación electrónica. Esto debido a la carencia de un sistema que le permita gestionar la variedad de bienes, la falta de control de stock de los productos para su política de precios y el nivel de servicio al cliente de forma eficiente. Sus objetivos se alcanzaron utilizando la metodología **SCRUM**, que permitió la planificación y ejecución en sprints. Se hizo uso de los programas como MySQL como gestor de base de datos, JavaScript, PHP. De este proyecto se está haciendo consideración de los programas de HTML /CSS así como el modelo de la interfaz gráfica de facturación que lo adaptamos a nuestra creación de boleta para la creación de nuestro interfaz en Top Modas también para el desarrollo de la lógica estamos tomando en cuenta el uso de JavaScript.



**Figura 3 Interfaz de gráfica de facturación electrónica de Livecor**

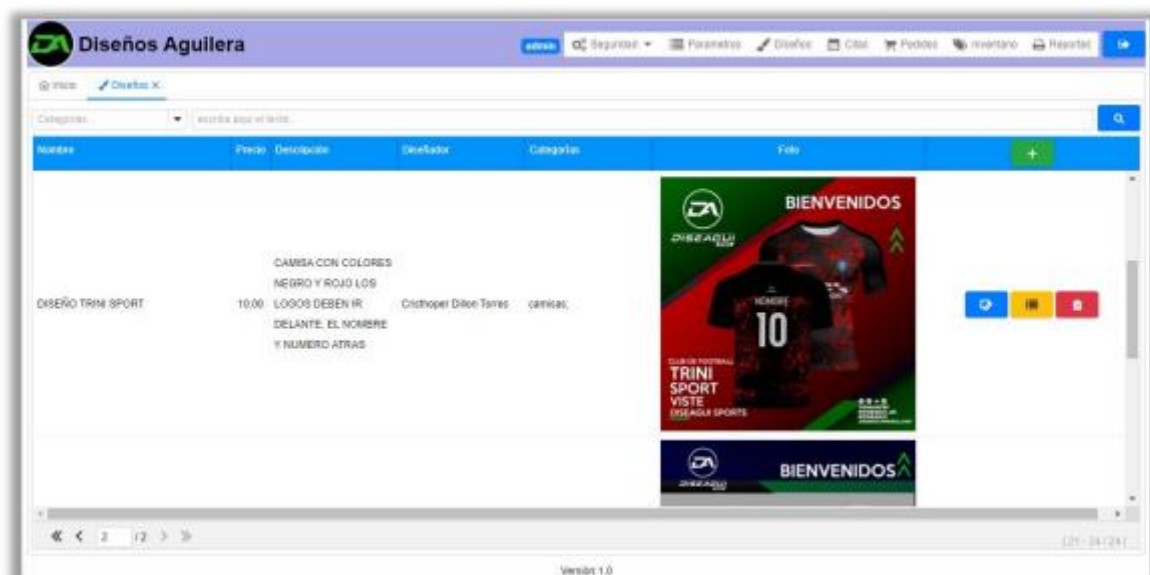
(Moreira Jiménez & Castillo Gutiérrez, 2023)

## Desarrollo de un sistema web para la gestión de producción de prendas de la empresa de confecciones y diseños Aguilera

En el proyecto "Diseños Aguilera", se utilizó la **metodología incremental** para el desarrollo del sistema web. El equipo logró sus objetivos mediante el desarrollo web



para la gestión de inventarios que se centra en el manejo eficiente de la materia prima necesaria para la confección de prendas. Utilizando MySQL como base de datos. El proyecto hace uso de HTML como parte de las tecnologías empleadas para desarrollar el sistema web. Se utiliza junto con el framework ZK para construir la interfaz de usuario y ZK genera automáticamente código JavaScript. Además, el sistema automatiza la actualización del inventario a medida que se generan y procesan los pedidos, asegurando un control preciso y en tiempo real de los materiales necesarios para la producción. De este proyecto quisiéramos implementar el uso del html así como el JavaScript que se utilizó para desarrollar la lógica del sistema.



**Figura 4 Interfaz de módulo de diseño de producción de confecciones y diseños Aguilera**

(Angie Madeline, 2022)

## **CAPITULO 02 – MARCO TEORICO**

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

#### **2.1.1 Metodología**

Para este proyecto usaremos la metodología SCRUM, pues la consideramos una alternativa muy eficaz, orientada a proyectos complejos y en evolución, permitiendo que este proyecto se escalara según las necesidades de Top Modas en el futuro.

#### **Product Owner (PO):**

Nuestro Product Owner en este caso es el representante de Top Modas. Ha sido este el encargado de brindarnos la información respecto a las necesidades y el funcionamiento de la tienda actual para definir los requisitos del sistema.

#### **Scrum Máster:**

El Scrum Máster se asegura de que el equipo siga las prácticas ágiles y gestiona soluciones para los obstáculos que interfieran en la productividad.

#### **Development Team:**

El equipo de desarrollo es el equipo de personas encargadas de diseñar y programar el sistema. En nuestro caso se estará trabajando en módulos separados definidos por nuestro requisitos funcionales y no funcionales, previamente ordenados para, de manera ordenada, sumar valor al producto final.

#### **Product Backlog:**

El Product Backlog la lista priorizada de todas las funcionalidades y tareas que el sistema debe considerar. El Product Owner se asegurará de priorizar estos elementos basándose en el valor que aportan al negocio. Para el sistema que haremos para Elite Auto Center, se tienen los siguientes.

- Registro del inventario inicial
- Actualización del inventario en tiempo real
- Generar reportes de inventario
- Gestión de proveedores

- Control de acceso
- Gestión y notificación de alertas

El enfoque principal de la metodología SCRUM es cómo gestiona el trabajo de manera ágil a través de eventos, artefactos y procesos que guían al equipo hacia sus metas. A continuación, se explica cómo se aplica SCRUM y sus fases o momentos clave:

### 1. **Sprint**

El Sprint es un ciclo de trabajo repetitivo, normalmente de 2 a 4 semanas, en el que el equipo desarrolla un incremento de producto potencialmente entregable. Todos los eventos de SCRUM ocurren dentro de este periodo. Cada Sprint tiene una meta clara y un conjunto de elementos a completar del Product Backlog.

### 2. **Sprint Planning (Planificación del Sprint)**

Antes de iniciar el Sprint, se realiza una reunión de planificación, donde el equipo define:

- Logro del Sprint.
- Cómo se va a lograr s
  - Se seleccionan los elementos del Product Backlog (historias de usuario, tareas) que serán trabajados en el Sprint.
  - Se desglosan los elementos seleccionados en tareas más pequeñas para gestionarlos de forma eficiente durante el Sprint.

### 3. **Daily Scrum (Reunión Diaria)**

Cada día durante el Sprint, el equipo se reúne por un máximo de 15 minutos para sincronizar el trabajo y ajustar el plan del día. Se abordan tres preguntas:

- ¿Qué hice ayer que ayudó al equipo a cumplir el Sprint?
- ¿Qué voy a hacer hoy para ayudar al equipo a cumplir el Sprint?
- ¿Hay algún impedimento que me esté bloqueando?

### 4. **Sprint Review (Revisión del Sprint)**

Al final de cada Sprint, el equipo presenta el incremento del producto desarrollado a los interesados. Aquí se evalúa lo que se ha completado comparado con el objetivo del Sprint. Es una oportunidad para obtener

retroalimentación valiosa por parte de los stakeholders y revisar si las metas del producto se están alineando con las expectativas.

## 5. **Sprint Retrospective (Retrospectiva del Sprint)**

Después de la revisión, el equipo celebra una retrospectiva para analizar el proceso y cómo ha sido el Sprint. Se evalúan aspectos positivos y áreas de mejora. El propósito es implementar pequeños ajustes que mejoren la eficiencia y efectividad del equipo en el próximo Sprint.

## 6. **Product Backlog**

Es una lista priorizada de todo lo que se necesita para mejorar el producto. El Product Owner es responsable de mantener este backlog ordenado y actualizado, según las necesidades del negocio y el feedback de los stakeholders.

## 7. **Sprint Backlog**

Es un subconjunto del Product Backlog. Durante la planificación del Sprint, se seleccionan elementos para trabajarlos durante ese ciclo. El Sprint Backlog es la guía diaria del equipo durante el Sprint y puede ajustarse según el progreso.

**TABLA 11: Comparativa de metodologías**

Característica	SCRUM	RUP
Estabilidad	Innovación, productividad y competitividad	Centrado en la arquitectura
Flexibilidad	Herramientas y técnicas para el trabajo en equipo	Flexibilidad para adecuarse a contextos y necesidades de recursos
Rendimiento	Priorización sólida en las funciones de los equipos	Enfoque en procesos definidos
Diseño	Se orienta a cualquier situación o sistema de desarrollo para su escalabilidad	Sistemas orientados a objetos y escalabilidad
Implementación	Adecuado para proyectos de mediana y gran escala	Adecuado para proyectos de mediana y pequeña escala
Pruebas	Verificable y adaptable para la retrospectiva	Verificación de cada iteración por funciones

*Fuente: Elaboración propia*

Se seleccionó la metodología SCRUM por adecuarse de manera fluida a distintas escalas de proyectos, además de permitirnos revisar el progreso de manera iterada para perfeccionar los aspectos que componen todo el sistema de forma ordenada y

sumando valor constantemente, además de permitir la escalabilidad que se puede necesitar en el rubro de nuestro Product Owner, como lo es Top Modas con la venta de autos.

### 2.1.2 Modelo

#### Modelo entidad relación

Un Modelo Entidad-Relación (ER) es una representación gráfica que se utiliza para diseñar bases de datos y describir la estructura de los datos. Este modelo permite visualizar cómo interactúan las entidades (que representan conjuntos de datos o conceptos del mundo real) entre sí, a través de relaciones. Los elementos que conforman al Modelo ER son los siguientes

**Entidades:** Son los objetos o conceptos del mundo real que se desean modelar. Cada entidad puede tener

Tipos de Entidades:

- **Entidades Fuertes (Regulares):** Son aquellas que tienen existencia propia en el sistema y no dependen de otras entidades para ser identificadas. Tienen una clave primaria que las identifica de manera única.
- **Entidades Débiles:** Son aquellas que dependen de otra entidad (entidad fuerte) para su existencia y no tienen una clave primaria propia. Estas entidades se identifican mediante una clave parcial y se apoyan en la clave primaria de la entidad fuerte relacionada.

**Relaciones:** Representan las asociaciones entre dos o más entidades. Se dibujan como rombos y pueden tener atributos propios.

Tipos de Relaciones:

- **Uno a uno (1:1):** Cada instancia de una entidad se asocia con exactamente una instancia de la otra entidad.
- **Uno a muchos (1):**
- Una instancia de una entidad puede estar relacionada con muchas instancias de otra entidad, pero una instancia de la otra entidad solo se relaciona con una instancia de la primera.

- Muchos a muchos (N):
- Muchas instancias de una entidad pueden estar relacionadas con muchas instancias de otra entidad.

**Atributos:** Son las propiedades o características que describen a una entidad o relación. Se dibujan como óvalos que se conectan a la entidad o relación correspondiente.

Tipos de Atributos:

- Atributo Simple: No se puede subdividir. Ejemplo: Nombre.
- Atributo Compuesto: Puede subdividirse en otras partes. Ejemplo: Dirección (se puede subdividir en calle, número, ciudad).
- Atributo Multivalorado: Puede tener varios valores. Ejemplo: Teléfonos (un cliente puede tener más de un número de teléfono).
- Atributo Derivado: Se calcula a partir de otros atributos. Ejemplo: Edad puede derivarse de la fecha de nacimiento.

Tipos de Claves

- Clave Primaria (Primary Key): Es un atributo (o conjunto de atributos) que identifica de manera única a cada instancia de una entidad.
- Clave Foránea (Foreign Key): Es un atributo en una tabla que hace referencia a la clave primaria de otra tabla. Esto crea una relación entre las dos tablas y asegura la integridad referencial.
- Clave Candidata: Son aquellos atributos que podrían ser claves primarias, pero no se seleccionaron como tales.
- Clave Alternativa: Cuando una entidad tiene varias claves candidatas, la clave que no es seleccionada como primaria se llama clave alternativa.

### Notación Gráfica de Relaciones

- Línea simple: Indica que la participación de una entidad en una relación es opcional, es decir, una instancia de la entidad puede no estar relacionada con ninguna instancia de la otra entidad.
- Línea doble: Indica que la participación es obligatoria. Todas las instancias de esa entidad deben estar relacionadas con al menos una instancia de la otra entidad.

- Flecha con tres líneas: Indica una relación de tipo N. Las entidades no tienen restricciones de cardinalidad, es decir, varias instancias de una entidad pueden relacionarse con varias de la otra.

### Notación de Cardinalidades (1, 0...1, 0...N)

- 1: Indica que la relación es obligatoria y una instancia de la entidad debe estar relacionada con exactamente una instancia de la otra entidad.
- 0...1: Indica que la relación es opcional y una instancia de la entidad puede estar relacionada con una o ninguna instancia de la otra entidad.
- 0...N: Indica que la relación es opcional, pero puede haber muchas instancias relacionadas.
- 1...N: Indica que la relación es obligatoria y puede haber una o muchas instancias relacionadas.

### 2.1.3 Patrones de la GUI

#### Patrón Formulario de Autenticación (Login Form Pattern)

Este patrón es crucial para aplicaciones que requieren seguridad y control de acceso. Consiste en un formulario sencillo donde el usuario ingresa su nombre de usuario o correo electrónico junto con su contraseña. Normalmente, está acompañado de un botón para enviar los datos y, a menudo, de enlaces para restablecer la contraseña o crear una nueva cuenta. La simplicidad y claridad en el diseño son esenciales para que los usuarios no se confundan ni experimenten frustración durante el proceso de inicio de sesión.

El diagrama muestra un formulario de autenticación con un fondo rosa pálido. En la parte superior, el texto "Correo electrónico:" precede a un campo de entrada rectangular. Debajo, el texto "Contraseña:" precede a otro campo de entrada rectangular. Entre ambos campos hay un espacio. Debajo del campo de contraseña, hay un botón rectangular de color rosa más intenso con el texto "Iniciar Sesión" en blanco. En la parte inferior del formulario, hay tres líneas de texto: "¿Necesitas ayuda?", "¿Olvidaste tu contraseña?" y "Contactar al Administrador".

**Figura 5 Patrón Formulario de Autenticación “Top Modas”**

*Fuente: Elaboración propia*

### **Patrón Restablecimiento de Contraseña (Password Recovery Pattern)**

El patrón de restablecimiento de contraseña es utilizado para ayudar a los usuarios a recuperar el acceso a su cuenta en caso de que olviden su contraseña. Generalmente, pide que el usuario ingrese su correo electrónico registrado para enviar un enlace o un código temporal que permita la creación de una nueva contraseña. Este proceso debe ser simple y seguro, garantizando que el usuario pueda restablecer su contraseña sin complicaciones ni riesgos de seguridad.

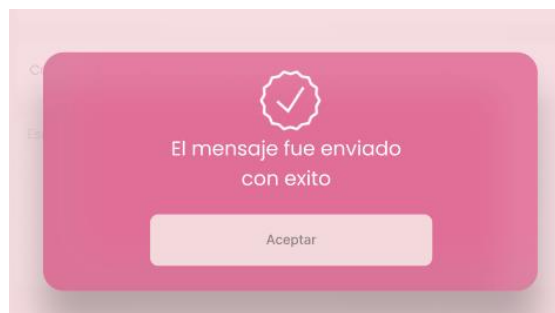
Un formulario de restablecimiento de contraseña con un fondo rosa. El título es "RESTABLECER CONTRASEÑA" en letras blancas. Debajo, un texto blanco indica: "Ingresa tu correo electrónico para recibir instrucciones sobre cómo restablecer tu contraseña." Hay un campo de entrada de texto blanco con el placeholder "Ingresa tu correo electrónico". En la parte inferior, hay un botón rectangular blanco con el texto "Aceptar".

**Figura 6 Patrón Restablecimiento de Contraseña “Top Modas”**

*Fuente: Elaboración propia*

### **Patrón Notificaciones Modales (Modal Notification Pattern)**

Las notificaciones modales son ventanas emergentes que aparecen sobre la interfaz principal para informar al usuario sobre el éxito o error de una acción. Se utilizan para destacar mensajes importantes que requieren la atención inmediata del usuario, como la confirmación de una acción completada o un error que necesita ser corregido. Los modales suelen bloquear la interacción con el resto de la interfaz hasta que el usuario tome una acción, como hacer clic en un botón de "Aceptar" o "Cerrar".

Un modal de notificación de éxito con un fondo rosa. En el centro, hay un icono de una marca de verificación dentro de un círculo. Debajo del icono, el texto blanco dice: "El mensaje fue enviado con éxito". En la parte inferior, hay un botón rectangular blanco con el texto "Aceptar".

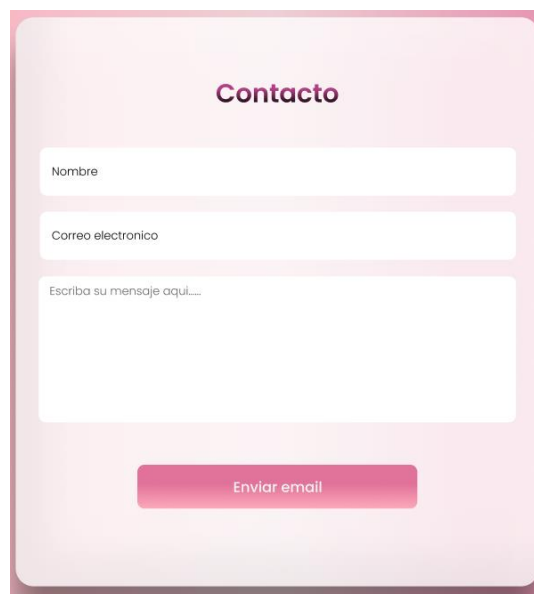
**Figura 7 Patrón Notificaciones Modales “Top Modas”**

*Fuente: Elaboración propia*



## Patrón Formulario de Contacto (Contact Form Pattern)

Este patrón se usa para proporcionar a los usuarios un medio de comunicación directo con el soporte o la empresa. Consiste en un conjunto de campos donde se solicita al usuario ingresar su nombre, correo electrónico y mensaje. El objetivo es facilitar una vía para que los usuarios hagan preguntas, den retroalimentación o soliciten asistencia. Es fundamental que este formulario sea fácil de entender y utilizar, asegurando que el usuario pueda enviar su consulta de manera rápida y efectiva.



El diagrama muestra un formulario de contacto con un fondo rosa pálido y una sombra. El título "Contacto" está en el centro superior. Debajo hay tres campos de entrada: "Nombre", "Correo electrónico" y un área de texto grande con el placeholder "Escriba su mensaje aquí...". En la parte inferior hay un botón rosa con el texto "Enviar email".

**Figura 8 Patrón Formulario de Contacto “Top Modas”**

*Fuente: Elaboración propia*

## Patrón Validación de Entrada (Input Validation Pattern)

El patrón de validación de entrada se enfoca en garantizar que los datos ingresados por el usuario cumplan con los requisitos establecidos, como formato, longitud y presencia de campos obligatorios. Cuando la entrada es incorrecta, el sistema muestra mensajes de error cercanos a los campos problemáticos, proporcionando una indicación clara de lo que necesita ser corregido. Este patrón es esencial para mejorar la precisión de los datos y evitar errores que podrían afectar el funcionamiento del sistema.

Correo electrónico

usuario@gmail.com

Ingrese su correo electronico

Contraseña

\*\*\*\*\*

Ingrese su contraseña

Iniciar Sesión

¿Necesitas ayuda?

¿Olvidaste tu contraseña?

Contactar al Administrador

**Figura 9 Patrón Validación de Entrada “Top Modas”**

*Fuente: Elaboración propia*

### **Patrón Vista Previa de Imagen (Image Preview Pattern)**

Este patrón se utiliza para visualizar imágenes de productos dentro de un modal. Este patrón es valioso en la gestión de inventario visual, ya que permite al usuario verificar que el producto añadido o editado, lo que es crucial en un sistema de ventas basado en productos visuales como ropa.

VER FOTO

Cambiar Foto

Eliminar Foto

Cancelar

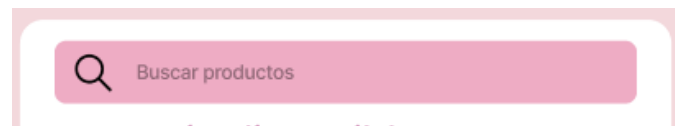
**Figura 10 Patrón Vista Previa de Imagen “Top Modas”**

*Fuente: Elaboración propia*

### **Patrón Búsqueda y Filtro (Search and Filter Pattern)**

Este patrón permite a los usuarios encontrar rápidamente elementos dentro de una gran colección de datos aplicando criterios de búsqueda y filtros específicos. Facilita la localización de información relevante sin necesidad de revisar manualmente largas listas o tablas. La Búsqueda generalmente implica una barra donde los usuarios

pueden introducir palabras clave para reducir los resultados a aquellos que coincidan con los términos ingresados. Los Filtros, por otro lado, permiten a los usuarios aplicar condiciones más específicas, como categorías, rangos de precios, fechas, o estados, afinando aún más la búsqueda. Este patrón es fundamental en interfaces con grandes volúmenes de datos, como catálogos de productos o sistemas de inventario, mejorando la eficiencia y precisión del proceso de búsqueda.

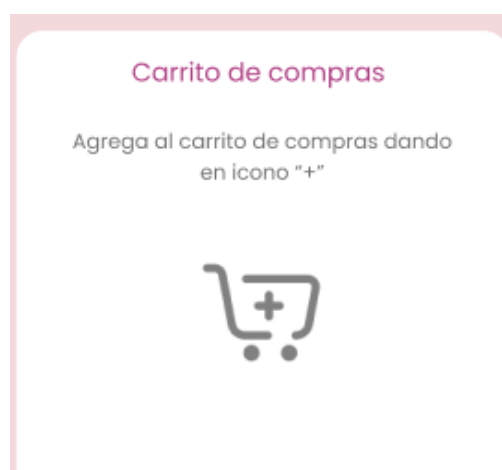


**Figura 11 Patrón Búsqueda y Filtro “Top Modas”**

*Fuente: Elaboración propia*

### **Patrón de Carrito de Compras (Shopping Cart Pattern)**

Es un diseño comúnmente utilizado en plataformas de comercio electrónico y sitios web en línea que permite a los usuarios reunir productos que desean comprar antes de completar la transacción. Este patrón imita el comportamiento de un carrito de compras físico, permitiendo a los usuarios navegar entre los productos, agregarlos a su carrito, modificar las cantidades, y revisar el costo total antes de proceder al pago. Las características clave de este patrón incluyen la visualización de detalles de los productos como precio y cantidad, una lista detallada de los artículos seleccionados, y opciones para eliminar o ajustar los productos.



**Figura 12 Patrón de Carrito de Compras “Top Modas”**

*Fuente: Elaboración propia*

## Patrón de Selección de Archivos (File Picker Pattern)

Es una solución de interfaz de usuario que permite a los usuarios cargar archivos desde su dispositivo local mediante un cuadro de diálogo emergente. Este cuadro de diálogo es invocado cuando el usuario hace clic en un botón o enlace de "Seleccionar archivo," lo que les permite navegar por los directorios de su dispositivo para elegir el archivo adecuado. Este patrón es común en aplicaciones web y móviles cuando se requiere la carga de archivos, como imágenes, documentos u otros tipos de medios. Es una forma eficiente y segura de interactuar con el sistema de archivos sin requerir interacción directa con el código o los directorios locales del usuario.

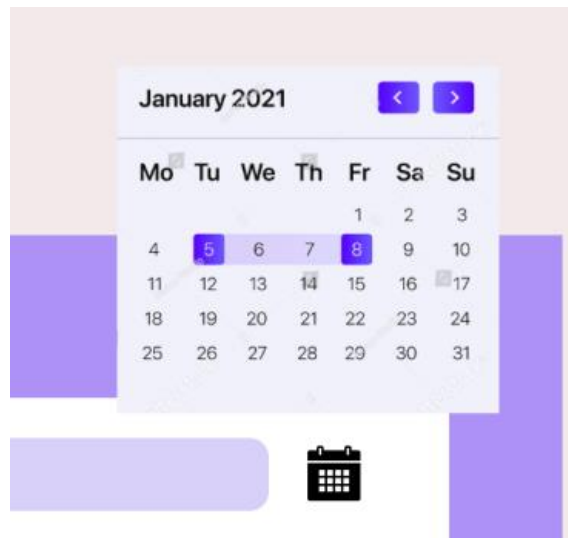


**Figura 13 Patrón de Selección de Archivos “Top Modas”**

*Fuente: Elaboración propia*

## Patrón de Calendario (Calendar Pattern)

Es una interfaz de usuario diseñada para facilitar la selección de fechas en aplicaciones y sitios web. Este patrón presenta un diseño visual que muestra un mes completo, con días organizados en una cuadrícula. Los usuarios pueden interactuar fácilmente con el calendario al hacer clic en un día específico para seleccionarlo. Este patrón no solo mejora la usabilidad al proporcionar un formato familiar y claro, sino que también permite incorporar características adicionales, como la navegación entre meses y la selección de rangos de fechas. Al implementar este patrón, las aplicaciones pueden ofrecer una experiencia más intuitiva y eficiente para la gestión de fechas y eventos.



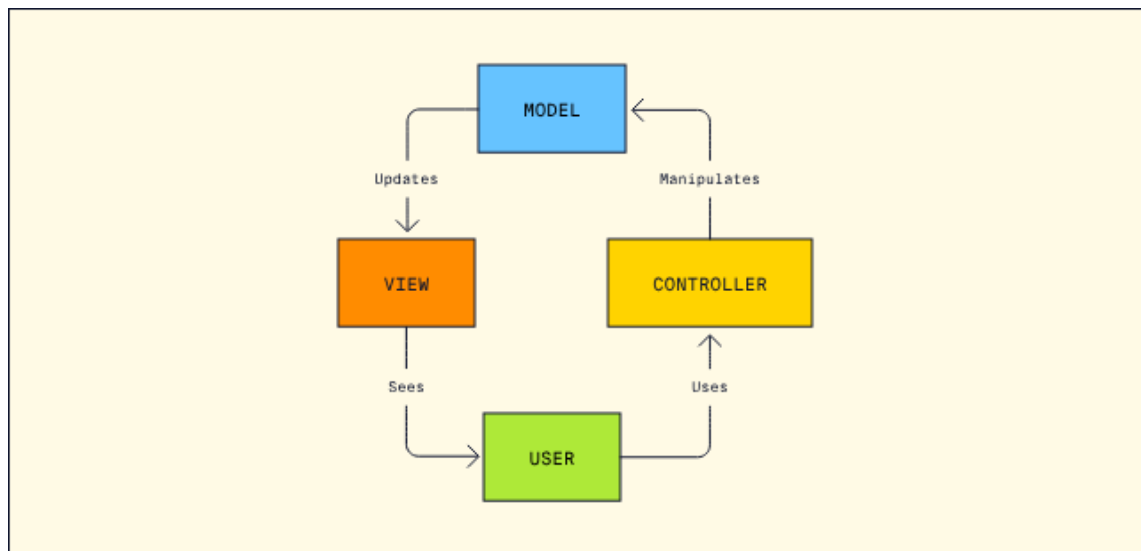
**Figura 14 Patrón de Calendario**

(Dreamstime. (n.d.))

#### 2.1.4 Patrones de desarrollo

##### **MVC (Model, view, controller)**

El Modelo-Vista-Controlador (MVC) es un patrón de diseño de software comúnmente utilizado para desarrollar interfaces de usuario, que divide la lógica del programa en tres elementos interconectados.



**Figura 15 Modelo-Vista-Controlador (MVC)**

(Adiga, S., 2022 )

Es una forma popular de organizar tu código. La gran idea detrás de MVC es que cada sección de tu código tiene un propósito, y esos propósitos son diferentes. Parte de tu código maneja los datos de tu aplicación, otra parte hace que tu aplicación se vea bien, y otra controla cómo funciona la aplicación.

MVC es una forma de organizar las funciones principales de tu código en cajas organizadas ordenadamente. Esto hace que sea mucho más fácil pensar en tu aplicación, revisarla y compartirla con otros de manera más clara y ordenada.

## Singleton

El patrón Singleton garantiza que solo se pueda crear una única instancia de una clase, sin importar cuántas veces se intente instanciar. Siempre se obtendrá la misma referencia a la instancia existente, lo que asegura un control centralizado de los recursos. Este patrón proporciona un punto de acceso global, lo que permite que la instancia única esté disponible a lo largo de toda la aplicación.

```
class Singleton {
  constructor() {
    if (!Singleton.instance) {
      Singleton.instance = this;
    }
    return Singleton.instance;
  }
}
const instance1 = new Singleton();
const instance2 = new Singleton();
console.log(instance1 === instance2); // true
```

**Figura 16 Patrón Singleton**

(Adiga, S., 2022 )

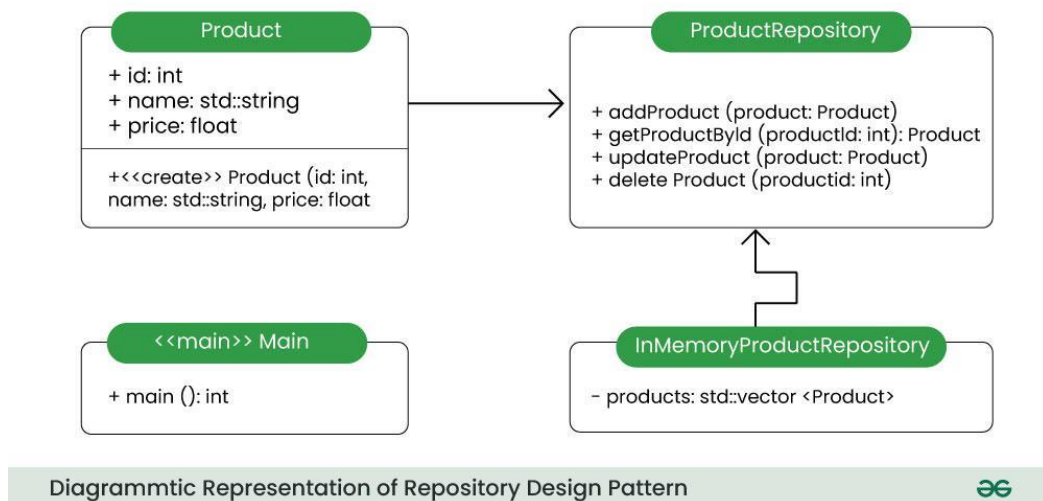
Una de las principales aplicaciones del patrón Singleton es la gestión de conexiones a bases de datos, como en el caso de MySQL en un backend. Implementar Singleton en la clase que maneja las conexiones asegura que

siempre se trabaje con una única conexión global, evitando la creación de múltiples conexiones innecesarias y optimizando los recursos del sistema. De esta forma, se reutiliza la misma conexión durante toda la ejecución de la aplicación, mejorando la eficiencia y garantizando un manejo centralizado de la base de datos.

## Repository

El patrón de repositorio es un valioso patrón de diseño que separa la lógica de acceso a datos del resto de la aplicación. Se usa para mantener una conexión flexible entre el cliente y los procedimientos de almacenamiento de datos en el servidor, ocultando toda la implementación compleja. Esto significa que el cliente no tendrá que preocuparse por cómo acceder a la base de datos, agregar o eliminar elementos de una colección, punteros, etc.

Con este patrón se puede mapear los atributos que vienen directamente de la base de datos al modelo del repositorio, lo que nos brinda flexibilidad. Si los atributos del elemento cambian, podemos modificarlos fácilmente en un único lugar donde se utilizan: el Repositorio. No es necesario ir a cada archivo donde se recupera el objeto, lo que respeta el principio SOLID.



**Figura 17 Patrón Repository**

(GeeksforGeeks. 2022)

### 1. Definiendo la interface repositorio

```
// IUserRepository.ts

export interface IUserRepository {
  findById(id: string): Promise<User | null>;
  findAll(): Promise<User[]>;
  create(user: User): Promise<User>;
  update(id: string, user: User): Promise<User | null>;
  delete(id: string): Promise<boolean>;
}
```

## 2. Implementando el repositorio

```
// UserRepository.ts

import { IUserRepository } from './IUserRepository';
import { User } from '../models/User';
import { Database } from '../services/Database'; // Assume a database service exists

export class UserRepository implements IUserRepository {
  private database: Database;

  constructor(database: Database) {
    this.database = database;
  }

  async findById(id: string): Promise<User | null> {
    // Implementation to fetch user by ID from the database
  }

  async findAll(): Promise<User[]> {
    // Implementation to fetch all users from the database
  }

  async create(user: User): Promise<User> {
    // Implementation to create a new user in the database
  }

  async update(id: string, user: User): Promise<User | null> {
    // Implementation to update a user in the database
  }
}
```



### 3. Usando el repositorio en nuestra aplicacion

```
// App.ts

import { UserRepository } from '../repositories/UserRepository';
import { User } from '../models/User';
import { Database } from '../services/Database';

const database = new Database(); // Initialize your database connection
const userRepository = new UserRepository(database);

async function exampleUsage() {
  const user = await userRepository.create({
    id: '1',
    name: 'John Doe',
    email: 'john@example.com',
  });

  console.log('Created user:', user);

  const foundUser = await userRepository.findById('1');
  console.log('Found user:', foundUser);

  const allUsers = await userRepository.findAll();
  console.log('All users:', allUsers);
}

exampleUsage();
```

### Object Relational Mapper (ORM)

Un ORM (Object Relational Mapper) es una herramienta que permite a los desarrolladores interactuar con una base de datos utilizando el paradigma de programación orientada a objetos, en lugar de escribir consultas SQL directamente. El ORM actúa como un puente entre el mundo del lenguaje de programación y las tablas de una base de datos relacional.

El ORM mapea las tablas de la base de datos a clases y las filas a instancias de esas clases, convirtiendo cada fila de la base de datos en un objeto. A su vez, los atributos de la clase corresponden a las columnas de la tabla, lo que facilita la manipulación de datos como si fueran objetos en el lenguaje de programación que estés usando, como Python, Java, PHP, etc. Al utilizar un ORM, los desarrolladores pueden interactuar con la base de datos sin tener que escribir consultas SQL complejas, lo que acelera el proceso de desarrollo.

#### ORMs populares:

- Hibernate (Java)
- Entity Framework (C#/.NET)
- Django ORM (Python)

- SQLAlchemy (Python)
- Doctrine (PHP)

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

Para el desarrollo del Sistema de Inventario de "Top Modas", se eligieron cuidadosamente los lenguajes de programación, frameworks y herramientas que mejor se adaptan a las necesidades del proyecto, considerando aspectos como rendimiento, facilidad de uso, mantenimiento y escalabilidad.

#### **JavaScript**



JavaScript es un lenguaje de programación que fue creado en 1995 por Brendan Eich mientras trabajaba en Netscape Communications Corporation. El objetivo inicial era desarrollar un lenguaje de scripting ligero y dinámico que pudiera ejecutarse en navegadores web para agregar interactividad a las páginas web. Originalmente, JavaScript fue desarrollado en tan solo 10 días y se llamaba Mocha, luego se renombró a LiveScript, y finalmente, fue bautizado como JavaScript para aprovechar la popularidad del lenguaje Java en ese momento, aunque no están directamente relacionados.

Desde entonces, JavaScript ha evolucionado significativamente y se ha convertido en un lenguaje de propósito general que se utiliza en el desarrollo web, aplicaciones móviles, desarrollo backend y más. Es uno de los pilares fundamentales del desarrollo web moderno, junto con HTML y CSS.

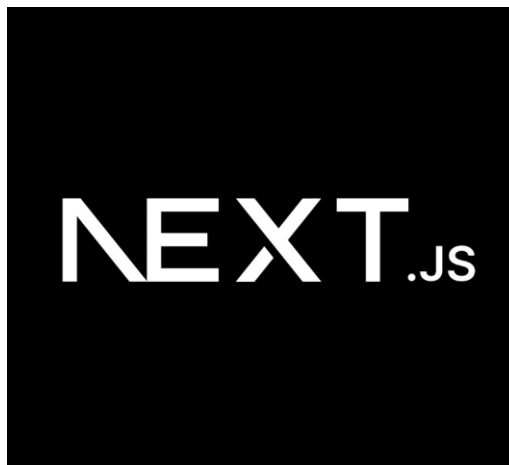
#### ***Ventajas de JavaScript:***

- JavaScript es uno de los pocos lenguajes que permite programar tanto en el frontend como en el backend gracias a Node.js. Esto facilita la

colaboración y reutilización de código entre distintas partes del proyecto, lo que resulta en un desarrollo más eficiente y menos propenso a errores.

- JavaScript cuenta con una comunidad de desarrolladores muy activa y un ecosistema de librerías y frameworks muy amplio (por ejemplo, React, Express para APIs, etc.). Esto significa que hay muchas herramientas, recursos y paquetes disponibles que facilitan y aceleran el desarrollo de aplicaciones web.
- Dado que JavaScript es nativo de los navegadores, permite que la aplicación sea accesible desde cualquier dispositivo con un navegador web, ya sea una computadora, tableta o smartphone, lo cual es ideal para un sistema de gestión de inventario accesible desde cualquier lugar.

## **Next.js**



Next.js es un framework de React de código abierto que fue creado por la empresa Vercel (anteriormente conocida como Zeit) y lanzado por primera vez en 2016. Next.js se creó para resolver algunas de las limitaciones de React, proporcionando un entorno más completo para desarrollar aplicaciones web. A diferencia de React, que se enfoca principalmente en la construcción de la interfaz de usuario, Next.js agrega funcionalidades de servidor y otras características que facilitan la construcción de aplicaciones web modernas y escalables.

### ***Ventajas:***

- **Renderizado Híbrido (SSR y SSG):** Next.js ofrece capacidades de renderizado del lado del servidor (Server-Side Rendering, SSR) y generación de páginas estáticas (Static Site Generation, SSG), lo que mejora el rendimiento y la velocidad de carga de la aplicación. Esto es especialmente útil en un sistema de gestión de inventario, ya que las páginas se pueden cargar más rápido y con datos más actualizados.
- **Full-Stack con API Routes:** Con Next.js, puedes desarrollar tanto el frontend como el backend en el mismo proyecto. Las API Routes de Next.js permiten definir endpoints directamente dentro de la aplicación, lo que facilita la construcción de un sistema de inventario completo, con funcionalidades como la gestión de productos, usuarios y órdenes sin necesidad de configurar un servidor adicional.
- **Mejor SEO y Rendimiento:** Gracias a su capacidad de renderizar páginas en el servidor, Next.js mejora la optimización para motores de búsqueda (SEO), ya que los buscadores pueden indexar el contenido generado de forma más efectiva. Esto es muy importante para aplicaciones web que necesitan ser visibles en los resultados de búsqueda.
- **Rutas Dinámicas y Sistema de Enrutamiento Integrado:** Next.js incluye un sistema de enrutamiento basado en archivos, lo que significa que no necesitas configurar manualmente las rutas de tu aplicación. Esto facilita la creación de páginas para gestionar diferentes productos o categorías en el sistema de inventario de forma rápida y organizada.
- **Recarga en Tiempo Real (Hot Reloading):** Durante el desarrollo, Next.js proporciona Hot Module Replacement (HMR), lo que significa que puedes ver los cambios en tiempo real sin necesidad de recargar toda la aplicación. Esto acelera el proceso de desarrollo y facilita la implementación de nuevas funcionalidades en el sistema de inventario.
- **Optimización de Imágenes y Recursos Estáticos:** Next.js incluye funcionalidades para optimizar imágenes y recursos estáticos, lo que mejora el rendimiento de la aplicación. En un sistema de gestión de inventario, esto garantiza que las imágenes de los productos se carguen de manera eficiente, proporcionando una experiencia de usuario más fluida.

## HTML



HTML, por sus siglas HyperText Markup Language, es el lenguaje de marcado estándar utilizado para crear y estructurar el contenido de las páginas web. Fue desarrollado por Tim Berners-Lee en 1991 mientras trabajaba en el CERN (Organización Europea para la Investigación Nuclear) y es uno de los componentes fundamentales de la World Wide Web. HTML ha evolucionado a lo largo del tiempo, y su versión actual es HTML5, que fue lanzada en 2014 y ha ampliado las capacidades del lenguaje para crear páginas web más interactivas y modernas.

HTML es el lenguaje encargado de definir la estructura y organización del contenido de una página web. En el sistema de gestión de inventario, HTML permite organizar la información de los productos, categorías, y otros elementos de manera clara y coherente, lo que facilita su presentación al usuario. Además, es fundamental para la optimización en motores de búsqueda (SEO), ya que permite utilizar etiquetas semánticas que ayudan a los motores de búsqueda a entender mejor el contenido de la página. Esto mejora la visibilidad del sistema de inventario en los resultados de búsqueda, lo que es útil si se desea atraer a más usuarios o clientes potenciales.

## MySQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que fue creado por MySQL AB, una empresa sueca fundada por Michael Widenius, David Axmark, y Allan Larsson en 1995. En 2008, Sun Microsystems adquirió MySQL, y luego en 2010, Oracle Corporation compró Sun Microsystems, convirtiéndose en el propietario actual de MySQL. Desde su creación, MySQL ha evolucionado y se ha convertido en una de las bases

de datos más populares y ampliamente utilizadas en el mundo, gracias a su rendimiento, flexibilidad y facilidad de uso. Es una de las bases de datos más fiables y estables del mercado, utilizada por grandes compañías como Uber, Pinterest, Netflix o Amazon.

***Ventajas:***

- Es conocido por ser altamente eficiente y capaz de manejar grandes volúmenes de datos de manera rápida y confiable. Esto es fundamental para un sistema de gestión de inventario que necesita almacenar y acceder a información de productos, proveedores, y transacciones de forma eficiente.
- Permite realizar las operaciones básicas de bases de datos conocidas como CRUD (Create, Read, Update, Delete), que son esenciales para gestionar los datos de inventario. Estas operaciones permiten agregar, recuperar, actualizar y eliminar información de productos, categorías, usuarios, y otras entidades del sistema de inventario.

**NODE.JS**



Node.js es un entorno de ejecución de JavaScript de código abierto y multiplataforma que fue creado por Ryan Dahl en 2009. La motivación detrás de Node.js era permitir que los desarrolladores ejecutaran JavaScript fuera del navegador, lo que abrió la posibilidad de utilizarlo para desarrollo del lado del servidor. Node.js se basa en el motor V8 de Google Chrome, que es un motor de JavaScript de alto rendimiento. Desde su lanzamiento, Node.js ha ganado popularidad y se ha convertido en una de las tecnologías más utilizadas para el desarrollo backend de aplicaciones web y de servidor.

Una de las mayores ventajas de Node.js es que permite utilizar JavaScript en el backend, lo que significa que puedes desarrollar tanto el frontend como el

backend utilizando el mismo lenguaje. Esto facilita la colaboración entre los desarrolladores y la reutilización del código, ya que no es necesario aprender o usar un lenguaje adicional para el servidor. Asimismo, node posee NPM (Node Package Manager) el cual es el administrador de paquetes de Node.js y ofrece acceso a un vasto repositorio de más de un millón de paquetes y librerías que facilitan y aceleran el desarrollo. Esto significa que puedes aprovechar paquetes preexistentes para agregar funcionalidades como autenticación de usuarios, gestión de bases de datos, manejo de solicitudes HTTP, y más.

### 2.1.6 Gestor de base de datos

MySQL es un motor de bases de datos relacional de código abierto y gratuito, muy popular en el mercado de desarrollo de aplicaciones web.



MySQL utiliza el lenguaje SQL (Structured Query Language) para gestionar y manipular bases de datos. Es conocido por su velocidad, confiabilidad y facilidad de uso, lo que lo convierte en una opción común en proyectos que van desde sitios web pequeños hasta grandes aplicaciones web empresariales. Algunas de sus características más resaltables son:

- **Multiplataforma:**  
Funciona en diferentes sistemas operativos como lo pueden ser Windows, Linux y macOS.
- **Alto rendimiento:**  
Puede manejar grandes volúmenes de datos y usuarios simultáneos de manera eficiente.
- **Escalabilidad:**

Soporta desde pequeñas aplicaciones hasta grandes plataformas con millones de registros.

- **Seguridad:**

Proporciona control de acceso robusto, con capacidades de autenticación y cifrado.

- **Soporte de la Comunidad (nombre de la comunidad):**

Como es de código abierto, tiene una gran comunidad de desarrolladores, además de versiones empresariales con soporte.

- **Accesible mediante interfaces**

MySQL se puede acceder mediante interfaces gráficas como se puede con PHPMyAdmin para su fácil uso y administración.

### 2.1.7 Herramientas adicionales

#### **BIZAGI**

Es una plataforma de automatización de procesos de negocio que permite a las organizaciones diseñar, ejecutar y optimizar flujos de trabajo de manera eficiente y efectiva. Su enfoque visual y centrado en el usuario facilita la creación de modelos de procesos a través de una interfaz intuitiva, lo que permite que tanto los expertos como los no expertos en tecnología puedan participar en la documentación y mejora de los procesos empresariales. Esto no solo mejora la colaboración entre equipos, sino que también ayuda a estandarizar las prácticas dentro de la organización.

Algunas de sus características más destacadas son:

- **Modelado Visual de Procesos:** Bizagi permite a los usuarios crear diagramas de procesos utilizando una interfaz gráfica simple. Esto facilita la visualización de flujos de trabajo complejos y ayuda a los equipos a identificar áreas de mejora y optimización de manera más clara y directa.
- **Integración con Sistemas Existentes:** La plataforma ofrece robustas capacidades de integración, lo que permite a las organizaciones conectar Bizagi con aplicaciones y bases de datos ya en uso. Esto asegura que los datos fluyan de manera coherente entre diferentes sistemas, mejorando la precisión y eficiencia de las operaciones.



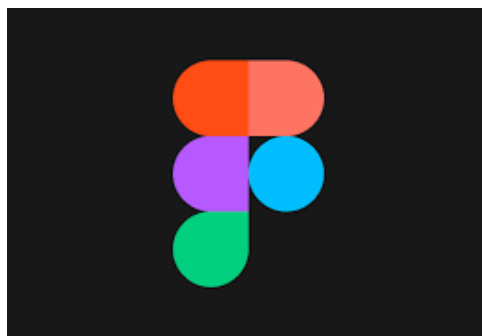


## FIGMA

Es una plataforma de diseño colaborativo basada en la nube que permite a equipos de diseño, desarrollo y otras partes interesadas trabajar juntos en tiempo real en proyectos de interfaz de usuario (UI) y experiencia de usuario (UX). Su enfoque centrado en la colaboración facilita la creación de prototipos interactivos y el diseño de sistemas de diseño de manera eficiente, lo que fomenta la comunicación efectiva entre los miembros del equipo. Además, al ser una herramienta basada en la web, elimina la necesidad de instalaciones complicadas y permite el acceso desde cualquier dispositivo con conexión a internet.

Algunas de sus características más destacadas son:

- **Colaboración en Tiempo Real:** Figma permite que varios usuarios trabajen simultáneamente en un mismo proyecto, viendo los cambios al instante. Esta funcionalidad es fundamental para facilitar la comunicación entre diseñadores y desarrolladores, reduciendo el tiempo de revisión y las iteraciones necesarias.
- **Prototipado Interactivo:** Con Figma, los diseñadores pueden crear prototipos interactivos sin necesidad de herramientas adicionales. Esto permite a los equipos presentar y probar sus diseños de manera efectiva, facilitando la obtención de feedback inmediato y optimizando el proceso de diseño.



## CANVAS

Es una plataforma de diseño gráfico en línea que permite a los usuarios crear una amplia variedad de materiales visuales, como presentaciones, infografías, carteles y publicaciones en redes sociales, de manera sencilla y accesible. Su interfaz intuitiva y basada en arrastrar y soltar permite que tanto diseñadores profesionales como principiantes puedan producir contenido visual atractivo sin necesidad de experiencia técnica previa. Además, Canva cuenta con una extensa biblioteca de plantillas, imágenes, iconos y elementos gráficos que facilitan el proceso de diseño y garantizan resultados de alta calidad.

Algunas de sus características más destacadas son:

- **Biblioteca de Plantillas y Recursos:** Canva ofrece miles de plantillas predefinidas para diferentes tipos de diseño, lo que ahorra tiempo y esfuerzo a los usuarios. También proporciona acceso a una amplia gama de imágenes de stock, ilustraciones y elementos gráficos que se pueden utilizar para personalizar sus proyectos.
- **Colaboración en Equipo:** La plataforma permite a los usuarios trabajar en proyectos de manera colaborativa, lo que facilita la revisión y el feedback en tiempo real. Esta funcionalidad es ideal para equipos de marketing, educación y otras industrias que requieren una colaboración fluida entre diferentes miembros del equipo.



## **LUCIDICHART**

Lucidchart es una plataforma en línea para la creación de diagramas y visualización de procesos que permite a los usuarios mapear ideas, flujos de trabajo y sistemas complejos de manera intuitiva y colaborativa. Su enfoque centrado en el diseño visual facilita la creación de diagramas de flujo, organigramas, mapas mentales, y otros tipos

de diagramas, lo que mejora la comunicación y la comprensión entre los equipos. Lucidchart es utilizado ampliamente en diversos campos, desde la ingeniería de software hasta la gestión de proyectos, por su capacidad de simplificar visualmente conceptos complejos y de integrarse con otras herramientas de productividad.

Algunas de sus características más destacadas son:

- **Colaboración en Tiempo Real:** Lucidchart permite que varios usuarios trabajen simultáneamente en un mismo documento, lo que facilita la colaboración entre equipos dispersos geográficamente y acelera el proceso de diseño y revisión.
- **Integración con Otras Herramientas:** La plataforma se integra fácilmente con herramientas populares como Google Drive, Microsoft Office, y Slack, permitiendo a los usuarios incorporar diagramas directamente en sus flujos de trabajo sin interrupciones.



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

### **2.2.1 Fórmulas**

#### **Indicadores Financieros**

Su función es medir el desempeño económico, son de gran importancia porque ayudan a evaluar la salud financiera de la empresa y a tomar decisiones relacionadas a inversiones y gastos. Además, usar estos indicadores de manera eficiente puede brindar ventajas competitivas, entre las cuales destacan las siguientes:

- Evaluar la optimización de recursos para identificar procesos innecesarios en distintas áreas de la empresa
- Sugerir soluciones frente a problemas relacionados con una mala gestión
- Detectar oportunidades de mejora para optimizar el rendimiento de los procesos que son llevados a cabo en todas las áreas que conforman la empresa

Ejemplo:

- Margen de utilidad bruta: Este indicador mide qué porcentaje de los ingresos totales corresponde a la ganancia bruta.

$$\text{Margen de Utilidad Bruta} = \left( \frac{\text{Ingresos Totales} - \text{Costo de Ventas}}{\text{Ingresos Totales}} \right) \times 100$$

### **Indicadores de Ventas**

Ayudan a medir las ventas realizadas según cantidad y valor, además, permiten identificar nuevas tendencias de acuerdo con el comportamiento del usuario. Estos indicadores nos ayudan a identificar tácticas para llevarnos a un mejor resultado, presentan las siguientes características:

- Permiten tomar decisiones estratégicas basadas en datos, estas pueden ser sobre qué productos promover o si se deben hacer cambios en los precios
- Permiten evaluar el impacto de campañas o lanzamientos para identificar nuevas estrategias de marketing
- Facilitan la adaptación del negocio a las tendencias del mercado

Ejemplo:

- Tasa de crecimiento de ventas: Mide el incremento o decremento de las ventas en un periodo comparado con otro.

$$\text{Tasa de Crecimiento de Ventas} = \left( \frac{\text{Ventas del Periodo Actual} - \text{Ventas del Periodo Anterior}}{\text{Ventas del Periodo Anterior}} \right) \times 100$$

### **Indicadores de Satisfacción del Cliente**

Permiten medir tanto la satisfacción como la lealtad del cliente con la empresa, puesto que es fundamental para su desarrollo y crecimiento, también presentan las siguientes características:

- Impulsan la recomendación, debido a que los clientes satisfechos son más propensos a recomendar la marca a otros usuarios
- Fomentan la innovación, puesto que la empresa se puede inspirar en los clientes para el desarrollo de nuevos productos, lo cual la mantiene competitiva en el mercado
- Mejoran la experiencia del cliente, en vista de que estos indicadores se centran en conocer a los clientes y superar sus expectativas

Ejemplo:

- Net Promoter Score (NPS): Mide la lealtad del cliente y su disposición para recomendar la empresa a otros.

$$NPS = \%Promotores - \%Detractores$$

### **Indicadores de Recursos Humanos**

Ayudan a medir el desempeño y la satisfacción de los trabajadores, fundamental para que el equipo se mantenga productivo y pueda realizar adecuadamente sus labores.

- Permiten evaluar el clima laboral, lo cual es importante para conocer la satisfacción de los empleados y tener la certeza de que se encuentran en un ambiente de trabajo positivo
- Fomentan un incremento en la productividad, ya que se tendrá información sobre el desempeño y la satisfacción laboral, ambos puntos contribuirán a mejorar la productividad dentro de la empresa
- Ayudan a identificar necesidades de capacitación, porque al llevar a cabo un monitoreo acerca del desempeño, se les podrá brindar formación a los empleados para potenciar sus habilidades según los objetivos de la empresa

Ejemplo:

- Rotación de personal: Mide el porcentaje de empleados que dejan la empresa en un periodo específico.

$$\text{Rotación de Personal} = \left( \frac{\text{Número de Empleados que Dejaron la Empresa}}{\text{Promedio de Empleados Totales}} \right) \times 100$$

### **Importancia de los indicadores**

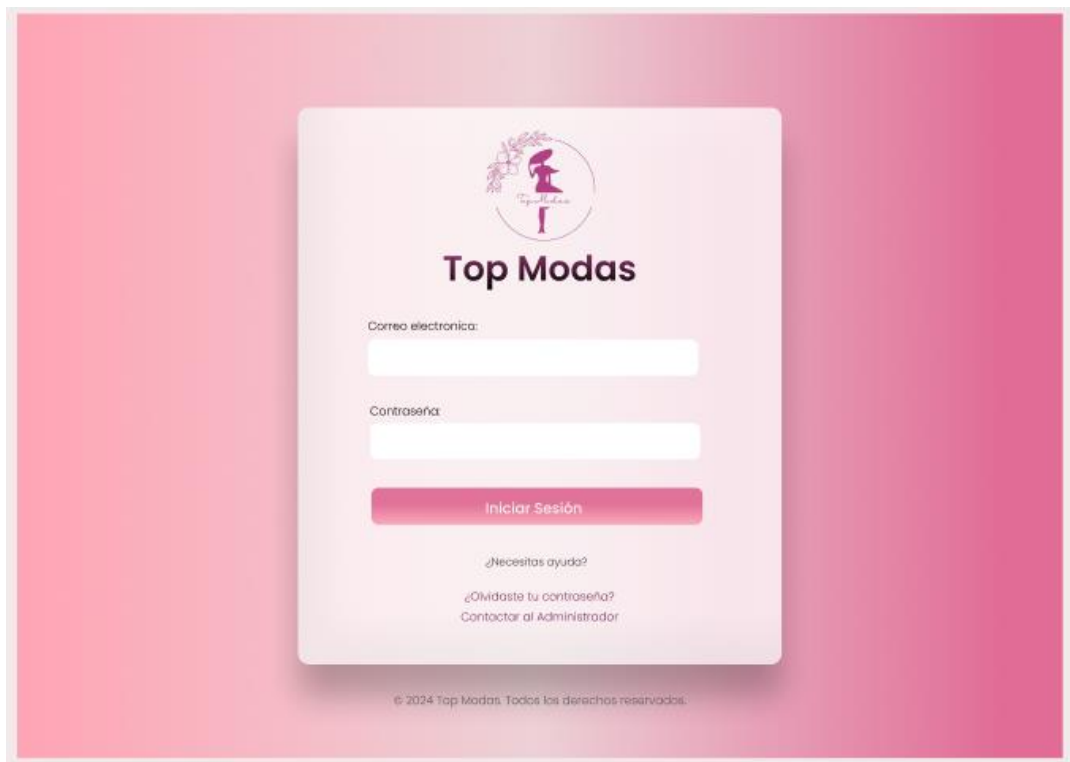
- Permiten realizar un monitoreo constante en distintas áreas de la empresa
- Brindan datos concretos para llevar a cabo una buena toma de decisiones
- Facilitan la detección de problemas en áreas que requieran algún tipo de mejora
- Ayudan a evaluar el desempeño del equipo de trabajo

## CAPITULO 03 – DESARROLLO DE LA SOLUCION

### 3.1 Prototipos

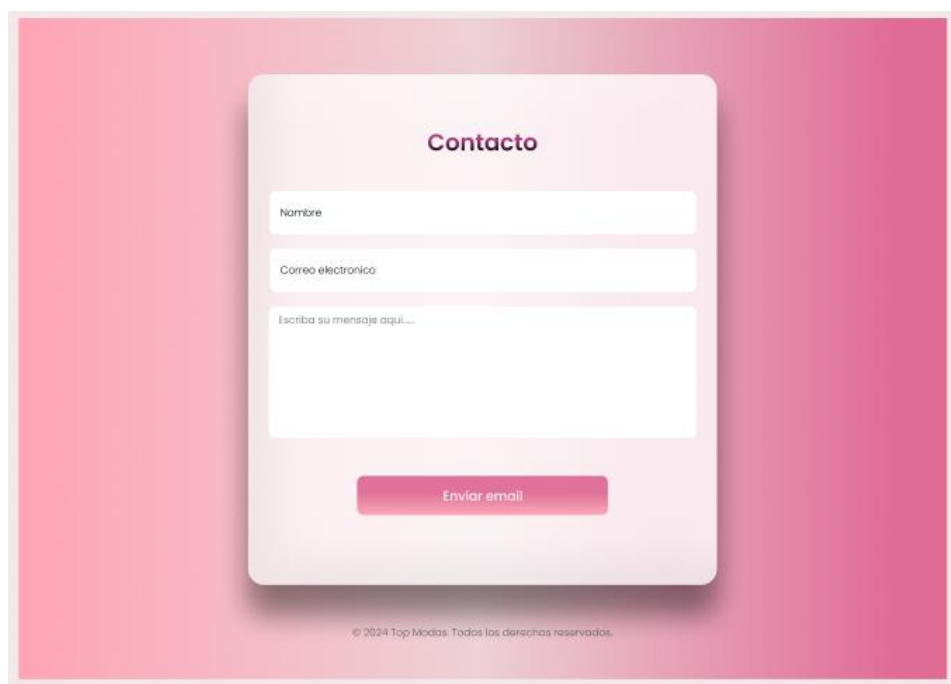
En esta sección se mostrarán los prototipos del sistema de gestión de inventario de la tienda "Top Modas", donde se podrá observar el diseño y la estructura de las diferentes interfaces. Las pantallas se dividen en 4 partes incluyen: inicio de sesión cajero, Cajero, inicio de sesión administrador y administrador. Estos prototipos ofrecen una visión detallada del flujo de navegación, así como del diseño visual del sistema, incorporando tablas, formularios y elementos gráficos que facilitarán la gestión de productos en la tienda. Además, se muestra la estructura de las imágenes de los productos y su correspondiente formato, asegurando que el sistema sea fácil de usar y visualmente coherente para los usuarios.

**Figura 18 Inicio de sesión cajero**



*Fuente: Elaboración propia*

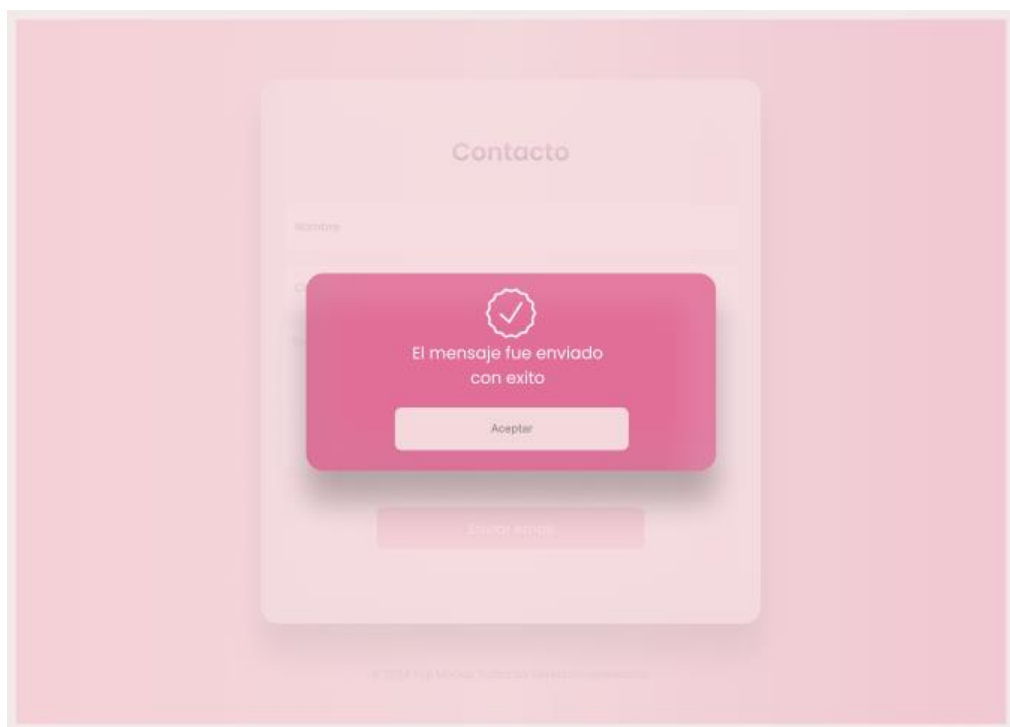
**Figura 19 Sección contactar administrador**



Formulario de contacto con el título "Contacto". Incluye tres campos de entrada: "Nombre", "Correo electrónico" y un área de texto para el mensaje con el placeholder "Escriba su mensaje aquí...". Debajo de los campos hay un botón "Enviar email". En la parte inferior del formulario, se muestra el copyright "© 2024 Top Modas. Todos los derechos reservados."

*Fuente: Elaboración propia*

**Figura 20 Caja de verificación para la sección contactar administrador**



Caja de verificación de éxito que indica que el mensaje fue enviado con éxito. El mensaje dice "El mensaje fue enviado con éxito" y hay un botón "Aceptar". En la parte inferior del formulario, se muestra el copyright "© 2024 Top Modas. Todos los derechos reservados."

*Fuente: Elaboración propia*

**Figura 21 Sección ¿Olvidaste tu contraseña?**

**RESTABLECER CONTRASEÑA**

Ingresa tu correo electrónico para recibir instrucciones sobre cómo restablecer tu contraseña.

Ingresa tu correo electrónico

✓ Se envía con este al correo electrónico escrito

Aceptar

[¿Olvidaste tu contraseña?](#)

[¿Olvidaste tu contraseña?](#)

[Contactar al administrador](#)

© 2023 Tuga Medios. Todos los derechos reservados

*Fuente: Elaboración propia*

**Figura 22 Caja de verificación para restablecer contraseña**

✓

Mensaje enviado. Revisa tu correo electrónico

Aceptar

[¿Olvidaste tu contraseña?](#)

[¿Olvidaste tu contraseña?](#)

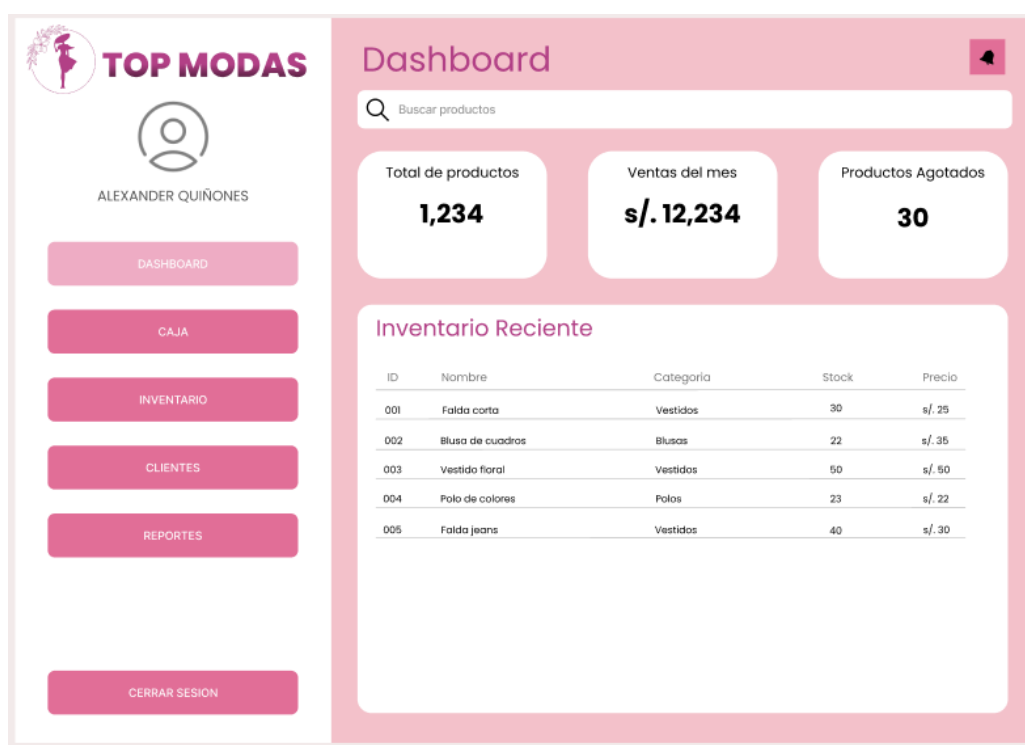
[Contactar al administrador](#)

© 2023 Tuga Medios. Todos los derechos reservados

*Fuente: Elaboración propia*

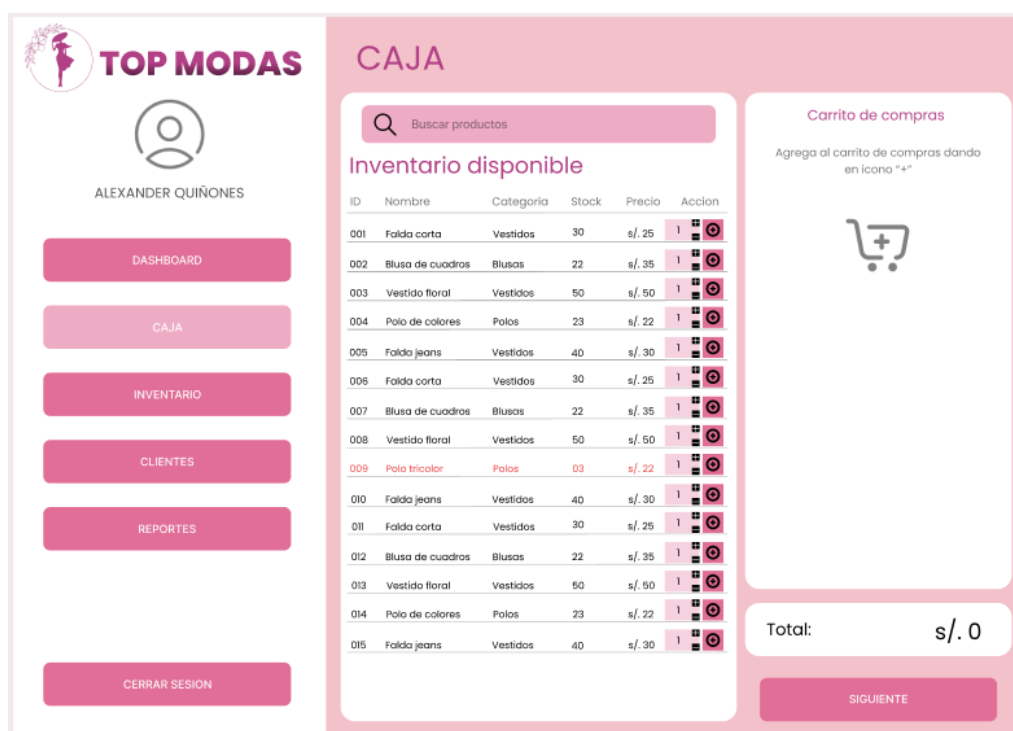


**Figura 23 Cajero sección Dashboard**



*Fuente: Elaboración propia*

**Figura 24 Cajero sección Caja**



*Fuente: Elaboración propia*

Figura 25 Boleta electrónica de la sección Caja


**TOP MODAS**  
R.U.C.: 20608430301  
CENTRAL: Av. Defensores Del Morro Nro.  
1277 Lima - Lima - Chorrillos  
E37 - BREÑA - LIMA - 1139  
Av.Venezuela N°651

**BOLETA DE VENTA ELECTRONICA**  
FECHA DE EMISIÓN: 20/09/2024 11:48:06  
CORRELATIVO: 0000456469  
CAJA/TURNO: 14  
TIPO DE MONEDA: PEN  
CAJERO : Alexander Quiñonez  
DNI Cliente: 75862146  
Cliente: Claudia Ruiz Montes

Descripción:	Cantidad x precio unitario	TOTAL
Falda corta	3 X s/. 25	s/. 75
IMPORTE TOTAL		s/. 75
Importe a pagar		s/. 75

Total: s/. 75

Guarda tu voucher,  
Es el sustento para validar tu compra.



Representación impresa de la boleta de venta electrónica  
Consulta documento en [www.TopModas.com.pe](http://www.TopModas.com.pe)  
Autorizado mediante resolución N° 0/SUNAT

Cambios y devoluciones:  
Salvo lo dispuesto por los art. 97, 98  
y demás de Ley 29571, aplicará lo dispuesto en nuestra política de  
satisfacción garantizada que podrás revisar en: Tienda Top Modas

Fuente: Elaboración propia

Figura 26 Cajero sección Inventario

**TOP MODAS**

ALEXANDER QUIÑONES

DASHBOARD

CAJA

**INVENTARIO**

CLIENTES

REPORTES

CERRAR SESION

### INVENTARIO

Consulta de inventario

STOCK ▼

Por Categoría ▼

Verifique la disponibilidad de productos

Q Buscar productos

ID	Nombre	Talla	Categoria	Stock	Precio
001	Falda corta		Vestidos	30	s/. 25
002	Blusa de cuadros	S, M, L	Blusas	22	s/. 35
003	Vestido floral	S, M, L	Vestidos	50	s/. 50
004	Polo de colores	S, M, L	Polos	23	s/. 22
005	Falda jeans	S, M, L	Vestidos	40	s/. 30
006	Falda corta	S, M, L	Vestidos	30	s/. 25
007	Blusa de cuadros	S, M, L	Blusas	22	s/. 35
008	Vestido floral	S, M, L	Vestidos	50	s/. 50
009	Polo tricolor	S, M, L	Polos	Producto por agotarse! 03	s/. 22
010	Falda jeans	S, M, L	Vestidos	40	s/. 30
011	Falda corta	S, M, L	Vestidos	30	s/. 25
012	Blusa de cuadros	S, M, L	Blusas	22	s/. 35
013	Vestido floral	S, M, L	Vestidos	ABOTADO 00	s/. 50
014	Polo de colores	S, M, L	Polos	23	s/. 22
015	Falda jeans	S, M, L	Vestidos	40	s/. 30

Fuente: Elaboración propia

Figura 27 Cajero sección Clientes

**TOP MODAS**

ALEXANDER QUIÑONES

DASHBOARD

CAJA

INVENTARIO

**CLIENTES**

REPORTES

CERRAR SESION

### CLIENTES

Registrar Nuevo Cliente

Buscar Clientes

Q Buscar cliente...

**Datos**

DNI:

Nombre:

Apellido:

Correo Electronico:

Nota:   
 (opcional )

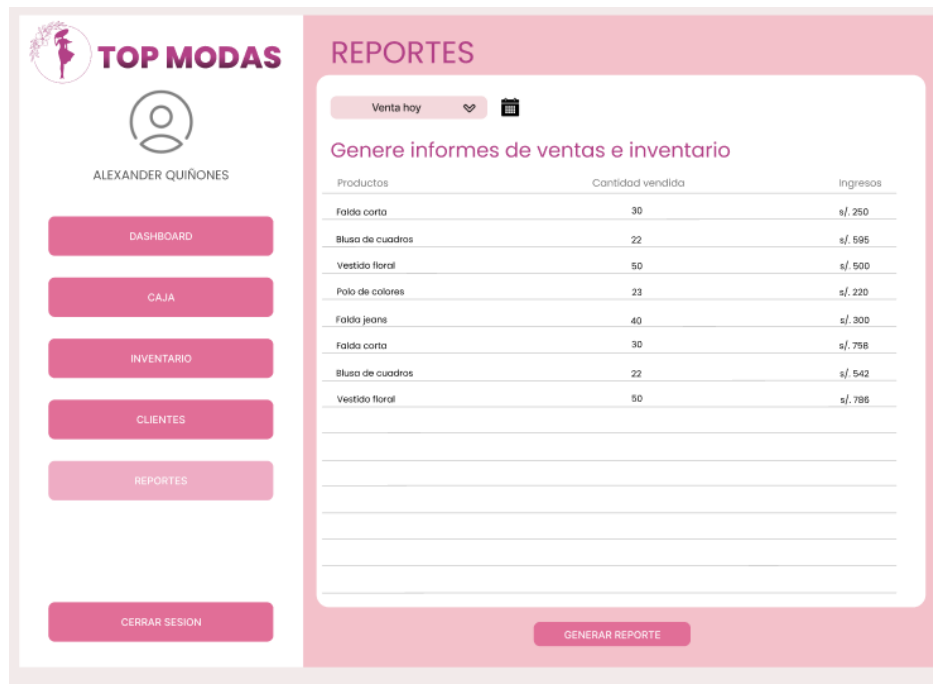
**GUARDAR CLIENTE**

**LIMPIAR CAMPO**

Nombre	Apellido	Correo Electronico	Accion
Rosa Maria	Ruiz Quiroz	Rosa_ruiz@gmail.com	<input checked="" type="checkbox"/> <input type="checkbox"/>
Luis Antonio	Anampa Quispe	Luis_quispe@gmail.com	<input checked="" type="checkbox"/> <input type="checkbox"/>
Diego Marco	Salas Quiñones	Diego_salas@gmail.com	<input checked="" type="checkbox"/> <input type="checkbox"/>

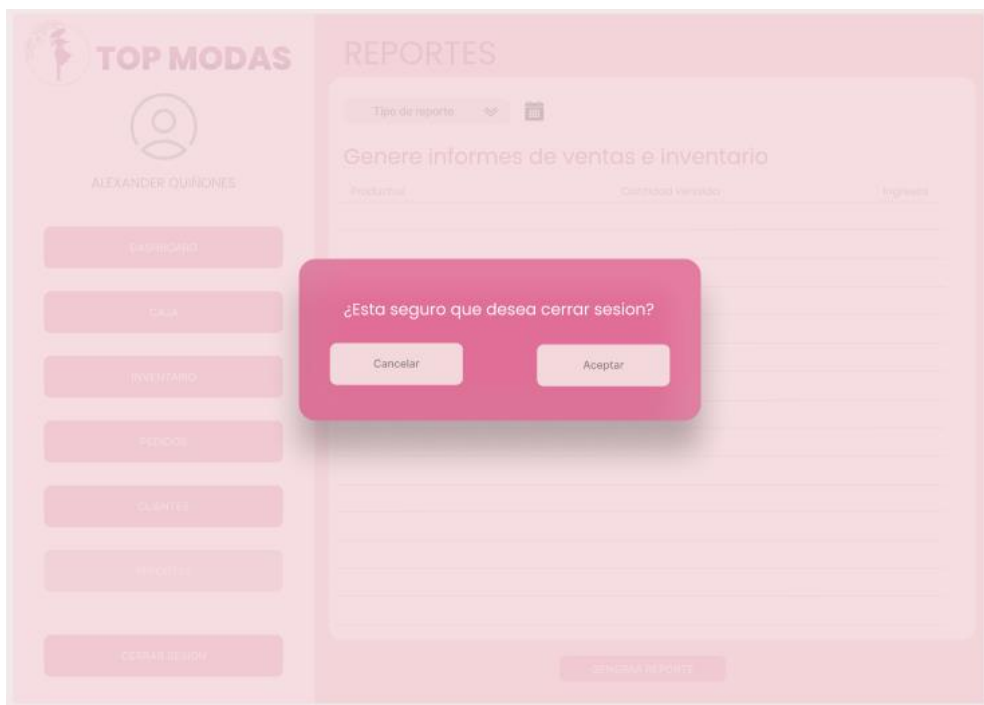
Fuente: Elaboración propia

**Figura 28 Cajero sección Reportes**



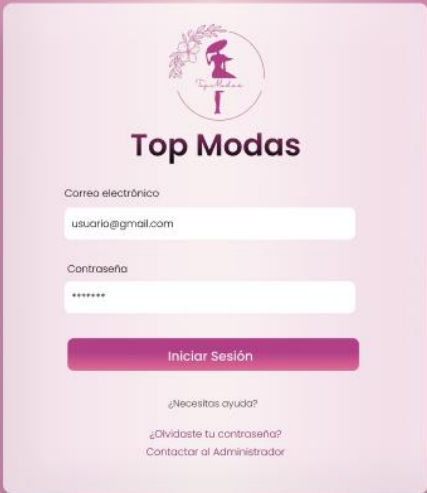
*Fuente: Elaboración propia*

**Figura 29 Caja de verificación para cerrar sesión del cajero**



*Fuente: Elaboración propia*

Figura 30 Inicio sesión de Administrador



The login form for the Top Modas administrator is centered on a purple gradient background. It features the Top Modas logo at the top, followed by input fields for 'Correo electrónico' (containing 'usuario@gmail.com') and 'Contraseña' (masked with asterisks). A purple 'Iniciar Sesión' button is positioned below the password field. Links for '¿Necesitas ayuda?', '¿Olvidaste tu contraseña?', and 'Contactor al Administrador' are located at the bottom of the form area.

Top Modas

Correo electrónico  
usuario@gmail.com

Contraseña  
\*\*\*\*\*

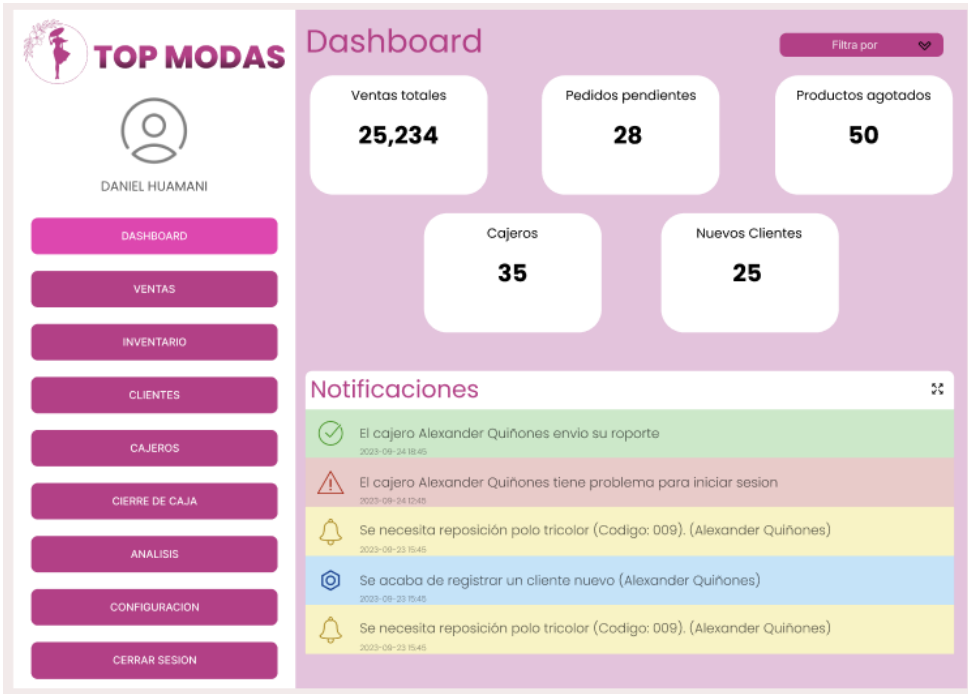
Iniciar Sesión

¿Necesitas ayuda?  
¿Olvidaste tu contraseña?  
Contactor al Administrador

© 2024 Top Modas. Todos los derechos reservados.

Fuente: Elaboración propia

Figura 31 Administrador sección Dashboard



The administrator dashboard for Top Modas features a left sidebar with navigation links and a main content area with summary cards and notifications. The sidebar includes the Top Modas logo, the user's name 'DANIEL HUAMANI', and buttons for Dashboard, Ventas, Inventario, Clientes, Cajeros, Cierre de Caja, Analisis, Configuración, and Cerrar Sesión. The main area displays five summary cards: Ventas totales (25,234), Pedidos pendientes (28), Productos agotados (50), Cajeros (35), and Nuevos Clientes (25). Below these is a 'Notificaciones' section with five items, including reports and inventory alerts.

TOP MODAS

DANIEL HUAMANI

DASHBOARD

VENTAS

INVENTARIO

CLIENTES

CAJEROS

CIERRE DE CAJA

ANALISIS

CONFIGURACION

CERRAR SESION

**Dashboard**

Filtrar por

Ventas totales  
**25,234**

Pedidos pendientes  
**28**

Productos agotados  
**50**

Cajeros  
**35**

Nuevos Clientes  
**25**

**Notificaciones**

- ✓ El cajero Alexander Quiñones envió su reporte  
2023-09-24 18:45
- ⚠ El cajero Alexander Quiñones tiene problema para iniciar sesion  
2023-09-24 12:45
- 🔔 Se necesita reposición polo tricolor (Codigo: 009). (Alexander Quiñones)  
2023-09-23 15:45
- 🔔 Se acaba de registrar un cliente nuevo (Alexander Quiñones)  
2023-09-23 15:45
- 🔔 Se necesita reposición polo tricolor (Codigo: 009). (Alexander Quiñones)  
2023-09-23 15:45

Fuente: Elaboración propia

Figura 32 Administrador sección Ventas

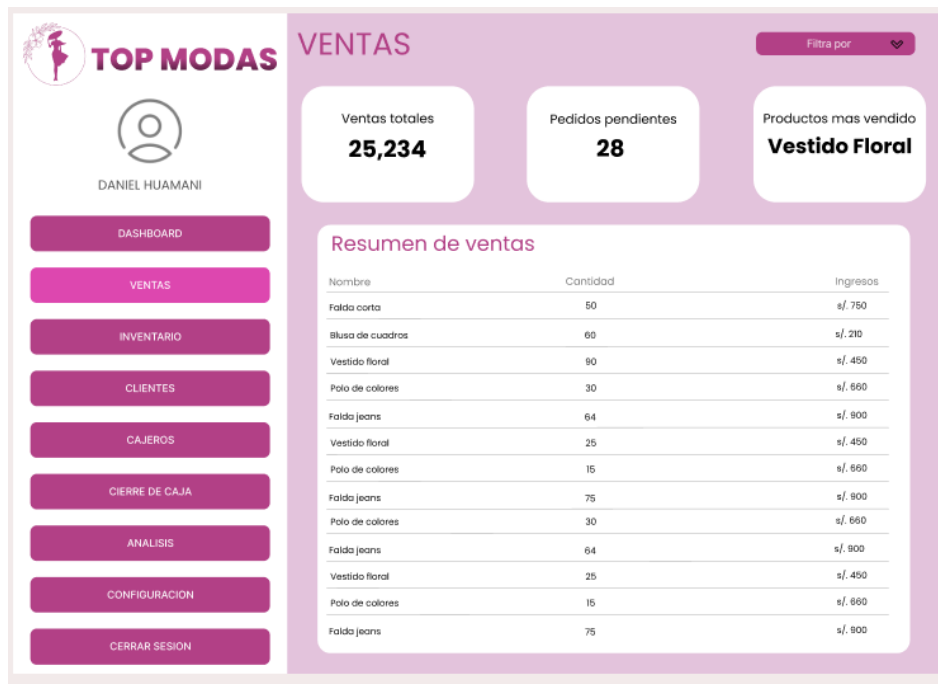


Figura 33 Administrador sección Inventario



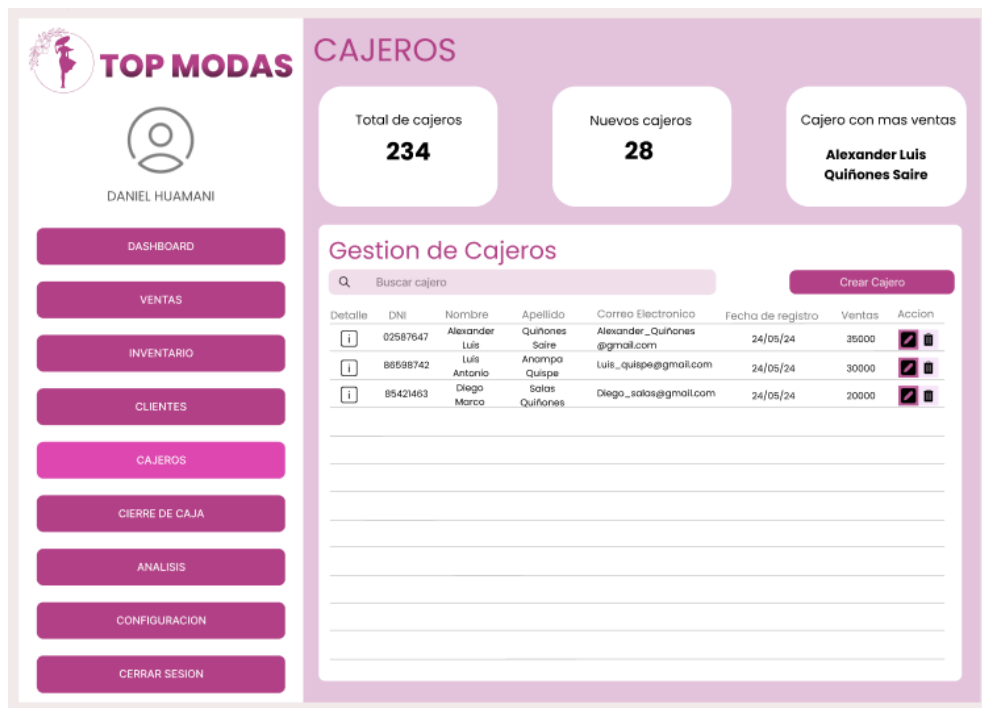
Fuente: Elaboración propia

**Figura 34 Administrador sección Clientes**



*Fuente: Elaboración propia*

**Figura 35 Administrador sección Cajeros**



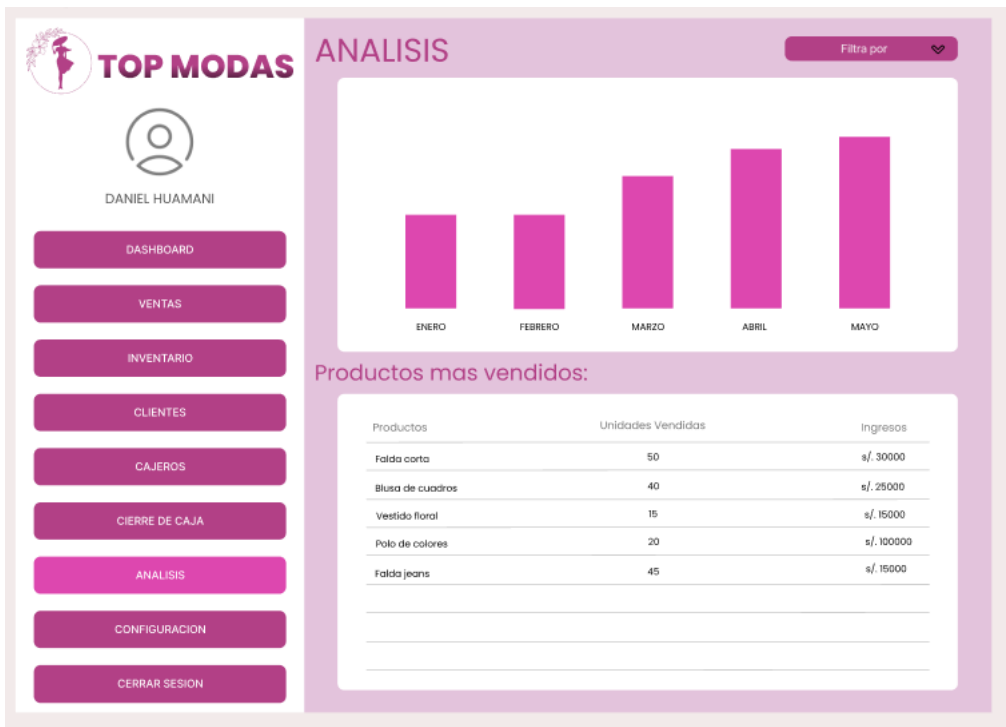
*Fuente: Elaboración propia*

Figura 36 Administrador sección Cierre de caja



Fuente: Elaboración propia

Figura 37 Administrador sección Análisis



Fuente: Elaboración propia



**Figura 38 Administrador sección Configuración**

**TOP MODAS**

DANIEL HUAMANI

DASHBOARD

VENTAS

INVENTARIO

CLIENTES

CAJEROS

CIERRA DE CAJA

ANALISIS

**CONFIGURACION**

CERRAR SESION

### CONFIGURACION

#### Cambio de contraseña

Inserta el código de seguridad de la tienda

Inserta antigua contraseña

Inserta la nueva contraseña:

Correo Electronico

**GUARDAR CAMBIOS**

CANCELAR

*Fuente: Elaboración propia*

**Figura 39 Caja de verificación para cerrar sesión del Administrador**

**TOP MODAS**

DANIEL HUAMANI

DASHBOARD

VENTAS

INVENTARIO

CLIENTES

CAJEROS

CIERRA DE CAJA

ANALISIS

CONFIGURACION

CERRAR SESION

### CONFIGURACION

#### Cambio de contraseña

Inserta el código de seguridad de la tienda

Inserta antigua contraseña

Inserta la nueva contraseña:

Correo Electronico

**GUARDAR CAMBIOS**

CANCELAR

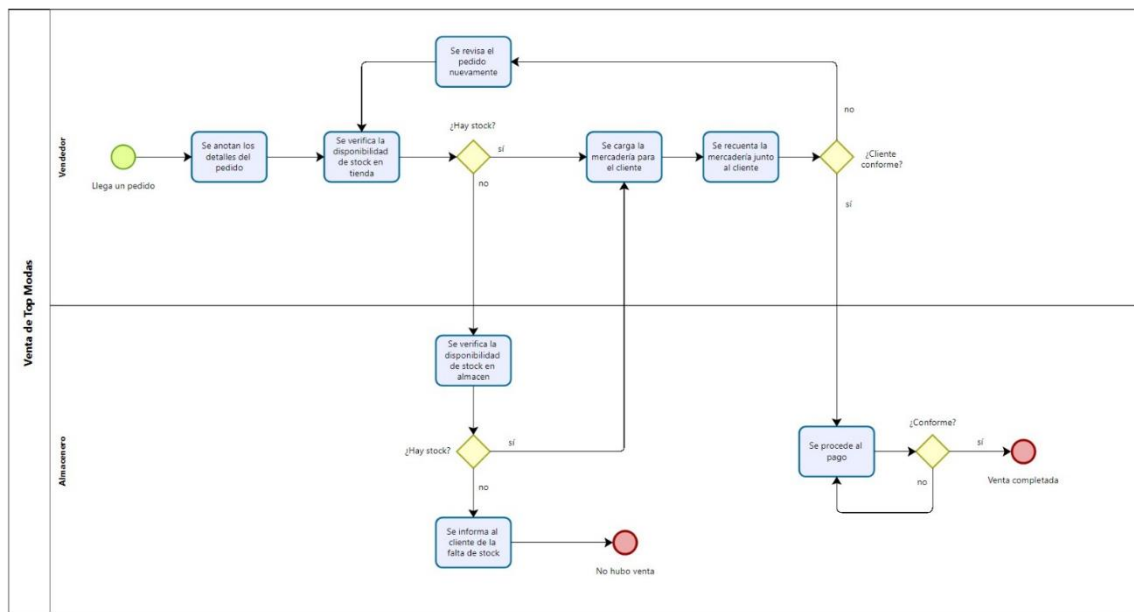
**¿Estas seguro que deseas cerrar sesion ?**

Cancelar Aceptar

*Fuente: Elaboración propia*

## 3.2 Modelo de procesos de negocios

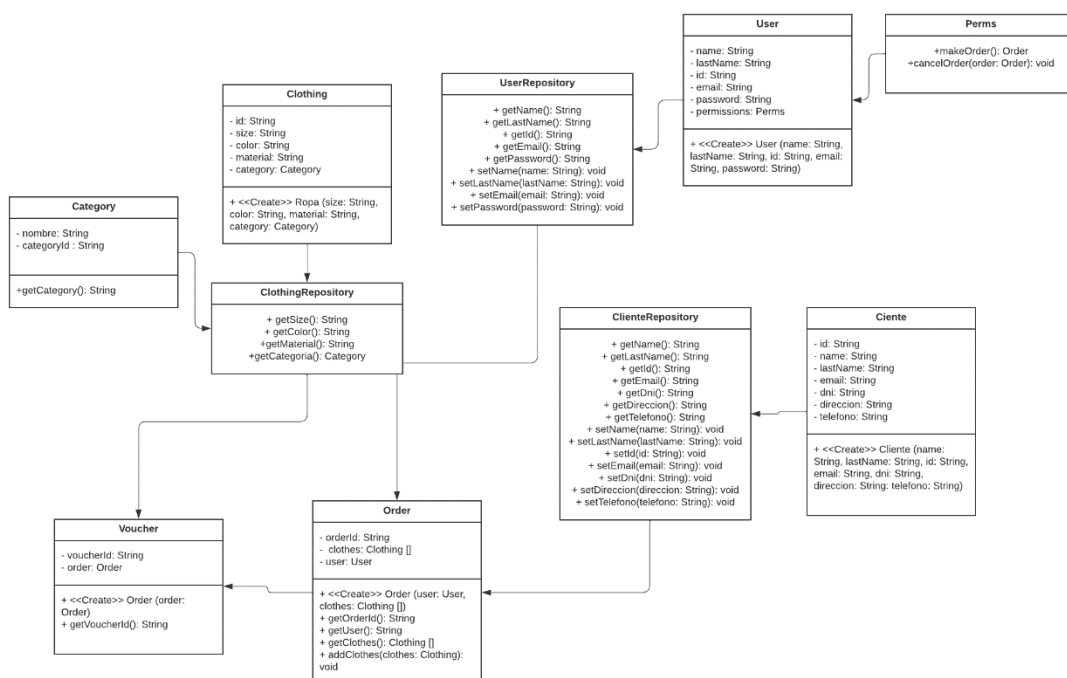
Figura 40 Modelo de negocios Bizagui



Fuente: Elaboración propia

## 3.3 Diagrama de clases

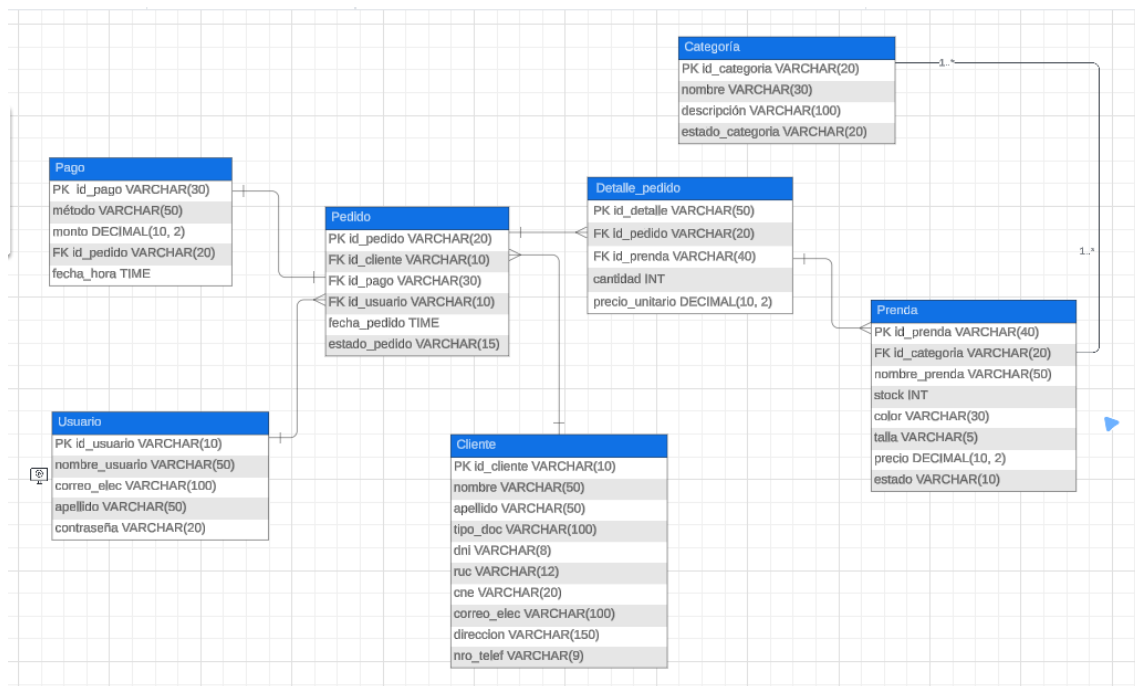
Figura 41 Diagrama de clase Top Modas



Fuente: Elaboración propia

### 3.4 Diagrama de entidad relación

Figura 42 Diagrama de entidad relación de Top Modas



Fuente: Elaboración propia

#### 3.4.1 Diccionario de datos

TABLA 12: Cliente

	Columna	Tipo de Dato	Descripción	Null
PK	id_cliente	VARCHAR (10)	Número de identificación del cliente	
	nombre	VARCHAR (50)	Primer nombre del cliente	
	apellido	VARCHAR (50)	Primer apellido del cliente	
	tipo_doc	VARCHAR (100)	Tipo de documento del cliente	
	dni	VARCHAR (8)	Documento nacional de identidad del cliente	
	ruc	VARCHAR (12)	Registro único de contribuyentes	
	cne	VARCHAR (20)	Carnet de extranjería	
	correo_elec	VARCHAR (100)	Correo electrónico	
	direccion	VARCHAR (150)	Dirección del cliente	
	nro_telef	VARCHAR (9)	Número de teléfono del cliente	

Fuente: Elaboración propia

**TABLA 13: Usuario**

	Columna	Tipo de Dato	Descripción	Null
PK	id_usuario	VARCHAR (10)	Numero de identificación del usuario	
	nombre_usuario	VARCHAR (50)	Nombre completo de usuario	
	correo_elec	VARCHAR (100)	Correo electrónico de usuario	
	apellido	VARCHAR (50)	Apellido completo del usuario	
	contraseña	VARCHAR (20)	Contraseña de usuario	

*Fuente: Elaboración propia*

**TABLA 14: Detalle\_Pedido**

	Columna	Tipo de Dato	Descripción	Null
PK	id_detalle	VARCHAR (100)	Numero de identificador de detalle de pedido	
FK	id_pedido	VARCHAR (50)	Numero de identificador de pedido	
FK	id_prenda	VARCHAR (20)	Numero de identificador de prenda	
	cantidad	INT	Cantidad de pedidos	
	precio_unitario	DECIMAL (10,2)	Precio unitario del producto	

*Fuente: Elaboración propia*

**TABLA 15: Pedido**

	Columna	Tipo de Dato	Descripción	Null
PK	id_pedido	VARCHAR (20)	Numero de identificador de pedido	
FK	id_cliente	VARCHAR (10)	Numero de identificador de cliente	
FK	id_pago	VARCHAR (30)	Numero de identificador de pago	
FK	id_usuario	VARCHAR (10)	Numero de identificador de usuario	
	fecha_pedido	TIME	Fecha de pedido	
	estado_pedido	VARCHAR (15)	Estado de pedido	

*Fuente: Elaboración propia*

**TABLA 16: Categoría**

	Columna	Tipo de Dato	Descripción	Null
PK	id_categoria	VARCHAR (20)	Numero de identificador de categoría	
	nombre	VARCHAR (30)	Nombre de la categoría	
	descripcion	VARCHAR (100)	Descripción del producto	
	estado_categoria	VARCHAR (20)	Tipo de categoría	

*Fuente: Elaboración propia*

**TABLA 17: Prenda**

	Columna	Tipo de Dato	Descripción	Null
PK	id_prenda	VARCHAR (40)	Numero de identificador de la prenda	
FK	id_categoria	VARCHAR (20)	Numero de identificador de la categoría	
	nombre_prenda	VARCHAR (50)	Nombre de la prenda	
	stock	INT	Stock de la prenda	
	color	VARCHAR (30)	Color de la prenda	
	talla	VARCHAR (5)	Talla de la prenda	
	precio	DECIMAL (10,2)	Precio de la prenda	
	estado	VARCHAR (10)	Estado de la prenda	

*Fuente: Elaboración propia*

**TABLA 18: Pago**

	Columna	Tipo de Dato	Descripción	Null
PK	id_pago	VARCHAR (30)	Numero de identificador de pago	
	metodo	VARCHAR (50)	Método de pago	
	monto	DECIMAL(10,2)	Monto de pago	
FK	id_pedido	VARCHAR (20)	Numero de identificador de pedido	
	fecha_hora	TIME	Fecha y hora del pago	

*Fuente: Elaboración propia*

## REFERENCIAS

Adiga, S. (2022, octubre 2). Using the Model-View-Controller (MVC) pattern in Flutter. Medium. <https://santhosh-adiga-u.medium.com/using-the-model-view-controller-mvc-pattern-in-flutter-fdc7dbeafe27>

Bits, N. (2023, febrero 2). *JavaScript factory design pattern: A comprehensive guide*. Medium. <https://medium.com/@nile.bits/javascript-factory-design-pattern-a-comprehensive-guide-9266b726ee5c>

Calibrant. (2022, junio 10). Singleton design pattern in JavaScript. Calibrant. <https://www.calibrant.com/blog/singleton-design-pattern-in-javascript>

Castillo Panchana, A. M. (2022). Desarrollo de un sistema web para la gestión de producción de prendas de la empresa de confecciones Diseños Aguilera (Tesis de Ingeniería en Tecnologías de la Información). Universidad Estatal Península de Santa Elena. <https://ejemplo.com/upse-tti-2022-0025>

Codecademy. (n.d.). *MVC: Model-View-Controller*. Codecademy. <https://www.codecademy.com/article/mvc>

Dreamstime. (n.d.). Concepto de calendario para aplicaciones móviles y web con selectores degradado rosa: UI template neumorphism design [Ilustración]. Dreamstime. <https://es.dreamstime.com/concepto-de-calendario-para-aplicaciones-m%C3%B3viles-y-web-con-selectores-degradado-rosa-ui-template-neumorphism-design-ilustraci%C3%B3n-image210901920>

GeeksforGeeks. (2022, agosto 3). Repository design pattern. GeeksforGeeks. <https://www.geeksforgeeks.org/repository-design-pattern/>

Hidalgo, M., & Torres, G. (2023). Normativa de espesores para componentes estructurales (T-ESPESD-003297). Instituto de Ingeniería Civil. <https://ejemplo.com/t-espesd-003297>

Lucas Vega, K. B. (2017). Desarrollo e implementación de aplicación web para el control de inventario del local comercial Máquinas Hidalgo [Proyecto técnico, Universidad Politécnica Salesiana]. Repositorio Institucional de la Universidad

Politécnica Salesiana. <https://dspace.ups.edu.ec/bitstream/123456789/15097/1/UPS-GT002054.pdf>

Mpalatsoukas, T. (2023, abril 16). *Repository pattern with JavaScript*. DEV. <https://dev.to/thanasismpalatsoukas/repository-pattern-with-javascript-4nl>

Scheid. (2022, diciembre 12). *Using clean architecture and the unit of work pattern on a Node.js application*. DEV. <https://dev.to/scheid/using-clean-architecture-and-the-unit-of-work-pattern-on-a-nodejs-application-3pc9>

## ANEXOS

# Lean Canvas

