



Universidad  
Tecnológica  
del Perú

Facultad de Ingeniería

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

## **SISTEMA DE INVENTARIO DE “TOP MODAS”**

Estudiantes:

Carbajal Añanca, Melany Daniela

Kim Negrillo, Lorena

Montesinos Poma, Britney Daphne

Rodriguez Martinez, Renzo Luciano

**Lima - Perú**

**2024**

## **METODOLOGIA**

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.

Comparación entre 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

(Esto hay que añadirlo al índice de tablas y la fuente es realización 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.

## MODELO (MELANY)

### Modelo Entidad - Relación

El modelo de entidad - relación (ER) fue introducido por Peter Chen en 1976 y actualmente sigue siendo muy utilizado, pues consiste en una técnica de modelado conceptual que generalmente se utiliza para representar de manera gráfica la estructura de una base de datos, su objetivo principal es describir los datos y las relaciones que se llevan a cabo entre ellos. Es una herramienta clave en la etapa de diseño, porque permite que los usuarios puedan observar la estructura entre los datos y cómo será su interacción. Por esta última razón, se le considera como un enfoque estándar para el diseño conceptual previo a la implementación en una base de datos relacional. Asimismo, el modelo de entidad – relación posee las siguientes componentes:

- **Entidad:** Representa un objeto con existencia independiente que se puede distinguir de otros, mediante el cual se desea almacenar información en la base de datos.
- **Entidad débil:** Se trata de una entidad que no puede ser identificada de forma única debido a sus propios atributos
- **Relación:** Consiste en la asociación entre dos o más entidades, describen cómo interactúan e indican las conexiones existen entre ellas. principalmente se clasifica en 3 tipos y son los siguientes:

**Uno a uno (1:1):** Una entidad de A está asociada con una y solo una entidad de B

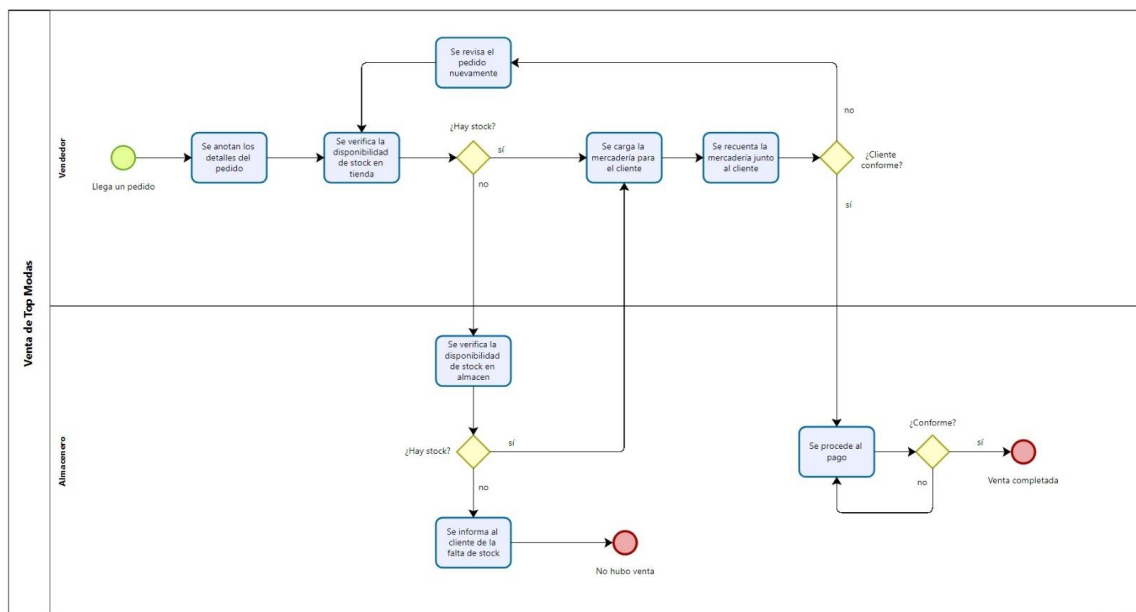
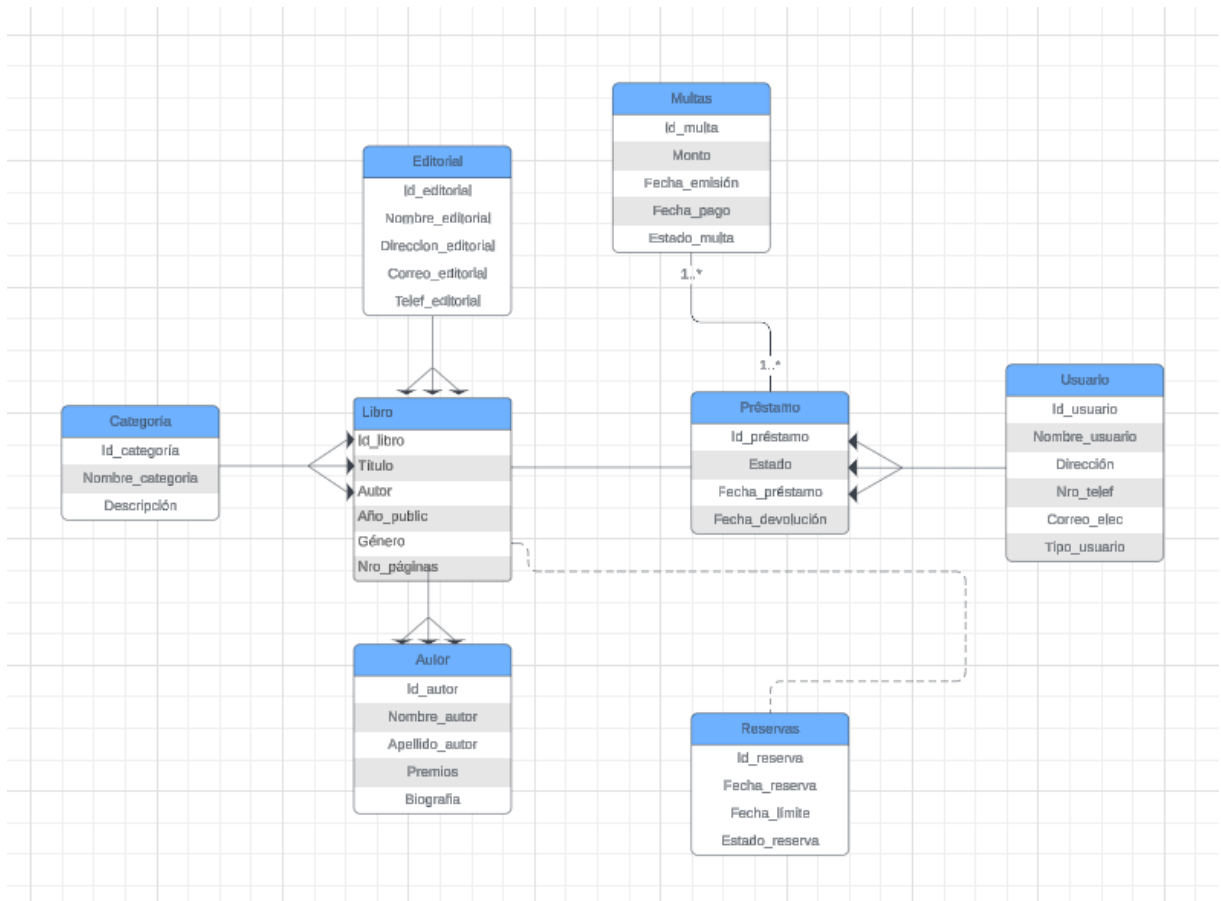
**Uno a muchos (1):** Una entidad de A está asociada con muchas entidades de B

**Muchos a muchos (M):** Muchas entidades de A pueden estar asociadas con muchas entidades de B

- **Atributo:** Característica o rasgo de un tipo de entidad que permite describirla
- **Clave Primaria:** Atributo o conjunto de atributos que identifica de manera única a cada entidad en una tabla
- **Clave Foránea:** Atributo en una tabla que hace referencia a la clave primaria de otra tabla, creando una relación entre ambas entidades
- **Diagrama entidad – relación:** Es una representación gráfica que muestra las entidades, sus atributos y las relaciones entre ellas dentro de un sistema

### **Ventajas del modelo ER**

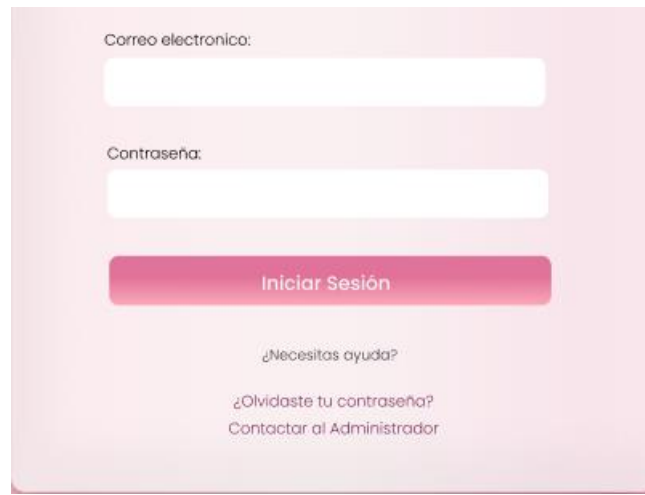
- Los diagramas de ER son fáciles de entender y al mismo tiempo proporcionan al usuario una visión clara de las estructuras de datos y sus relaciones
- El modelo ER puede ser convertido en un esquema de base de datos relacional, lo cual facilita su implementación
- El modelo ER es conceptual y no se encuentra ligado a ningún gestor de base de datos en específico, esto lo hace flexible para diferentes plataformas



## PATRONES DE LA GUI (BRITNEY)


### 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.

A login form with a light pink background. It features two input fields: the first is labeled 'Correo electrónico:' and the second is labeled 'Contraseña:'. Below the password field is a pink button with the text 'Iniciar Sesión'. At the bottom, there are two links: '¿Necesitas ayuda?' and '¿Olvidaste tu contraseña? Contactar al Administrador'.

### 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.

A password recovery form with a light pink background. It has a title 'RESTABLECER CONTRASEÑA' in bold. Below the title is a text prompt: 'Ingresa tu correo electrónico para recibir instrucciones sobre cómo restablecer tu contraseña.' There is an input field with the placeholder text 'Ingresa tu correo electrónico'. At the bottom is a pink button with the text 'Aceptar'.

### 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".



### **Patrón Formulario de Contacto (Contact Form Pattern)**

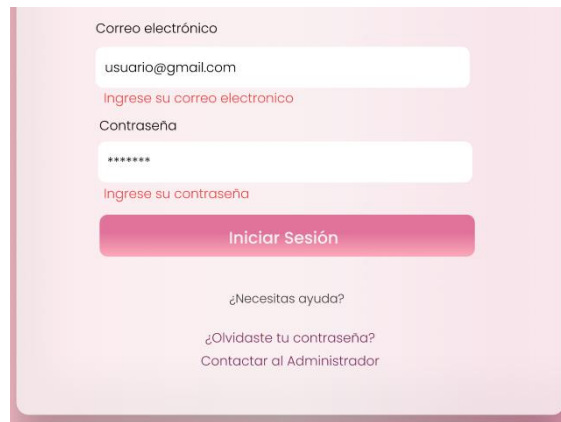
Este patrón se usa para proporcionar a los usuarios un medio de comunicación directo con el soporte o la empresa. Consiste en un conjunto de campos donde se solicita al usuario ingresar su nombre, correo electrónico y mensaje. El objetivo es facilitar una vía para que los usuarios hagan preguntas, den 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.

Una interfaz de usuario que muestra un formulario de contacto. El formulario es un recuadro rectangular con esquinas redondeadas, de color magenta claro, con un fondo más oscuro de color magenta. En la parte superior, el título "Contacto" aparece en un color magenta. Debajo del título, hay tres campos de entrada: un campo para "Nombre", un campo para "Correo electrónico" y un campo de texto grande para "Escriba su mensaje aquí...". En la parte inferior del formulario, hay un botón rectangular con el texto "Enviar email" en un color magenta.

### **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

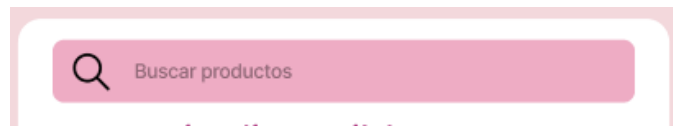
### Patrón Vista Previa de Imagen (Image Preview Pattern)

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.



### 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.



### 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.



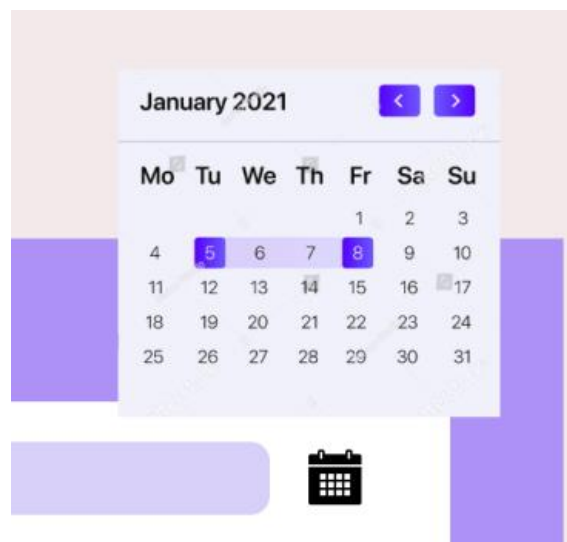
### Patrón de Selección de Archivos (File Picker Pattern)

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.



## Patrón de Calendario (Calendar Pattern)

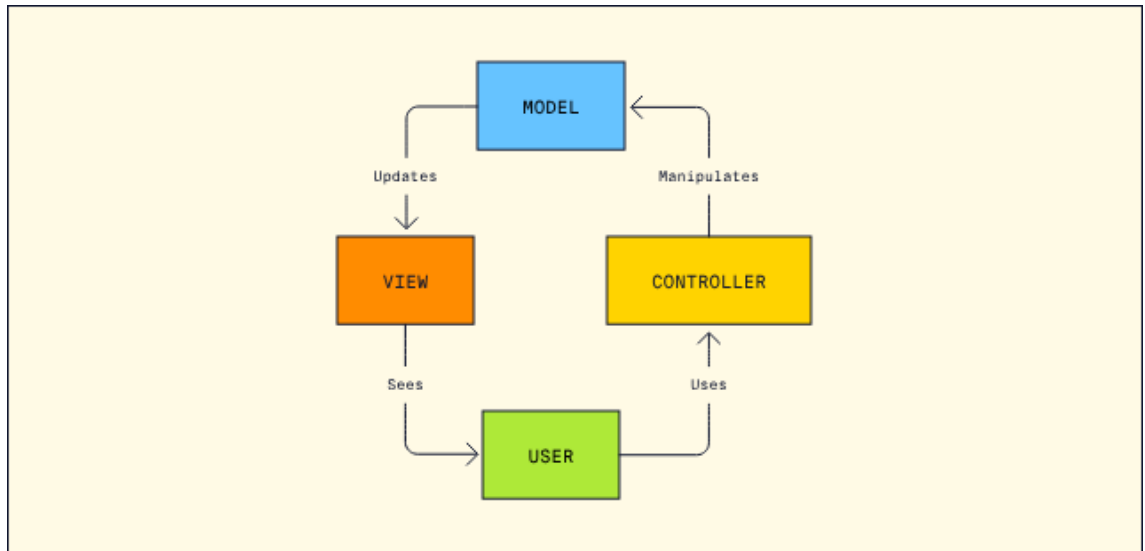
Es una interfaz de usuario diseñada para facilitar la selección de fechas en aplicaciones y sitios web. Este patrón presenta un diseño visual que muestra un mes completo, con días organizados en una cuadrícula. Los usuarios pueden interactuar fácilmente con el calendario al hacer clic en un día específico 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.



## PATRONES DE DESARROLLO (LORENA)

### 1. 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.



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.

## 2. 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;
  }
}

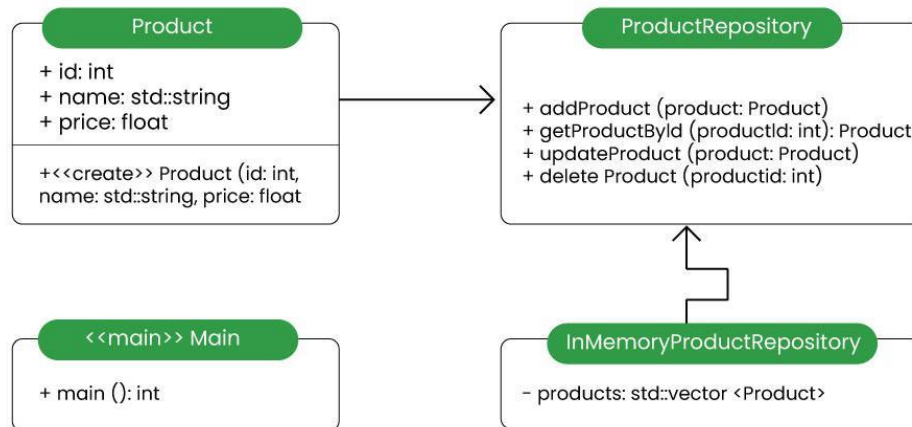
const instance1 = new Singleton();
const instance2 = new Singleton();
console.log(instance1 === instance2); // true
```

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.

### 3. 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.



Diagrammatic Representation of Repository Design Pattern



## 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();
```

#### 4. 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)

## LENGUAJES DE PROGRAMACION /FRAMEWORK (**LORENA**)

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.

### 1. JavaScript



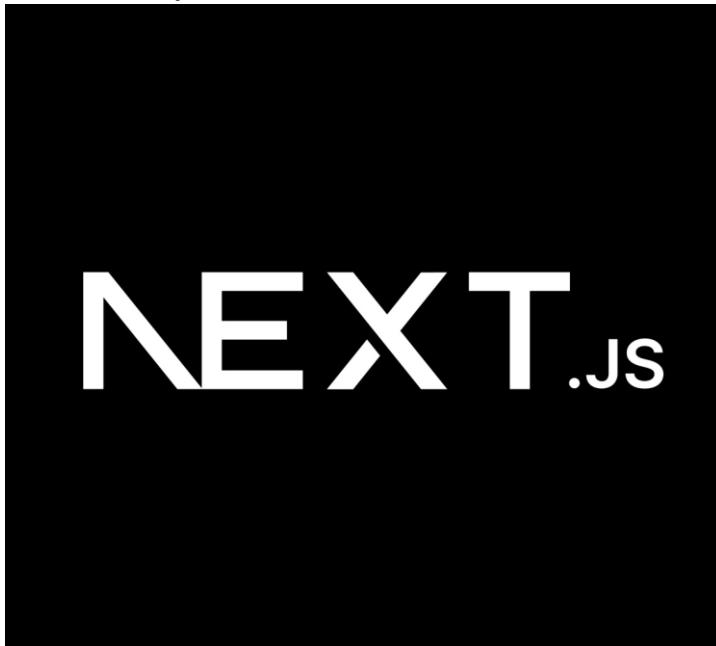
JavaScript es un lenguaje de programación que fue creado en 1995 por Brendan Eich mientras trabajaba en Netscape Communications Corporation. El objetivo inicial era desarrollar un lenguaje de scripting ligero y dinámico que pudiera ejecutarse en navegadores web para agregar interactividad a las páginas 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.

## 2. 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 tu 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 tu 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.

### 3. 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.

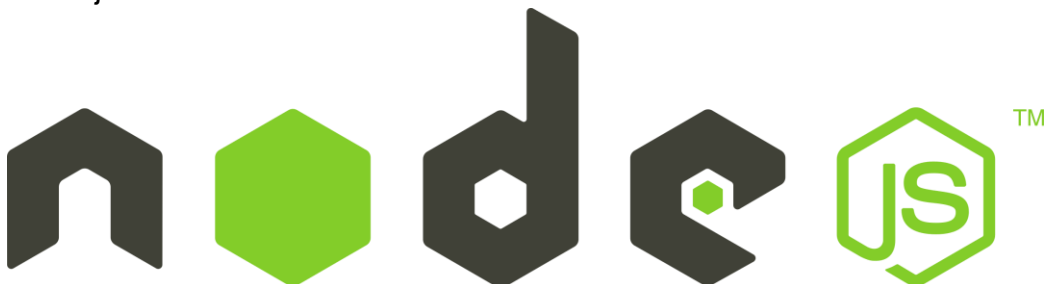
#### 4. MySQL

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.

#### 5. 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 a tu sistema de inventario, como autenticación de usuarios, gestión de bases de datos, manejo de solicitudes HTTP, y más.

## GESTOR DE BASE DE DATOS (RENZO)

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.

## HERRAMIENTAS ADICIONALES (BRITNEY)

### 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.

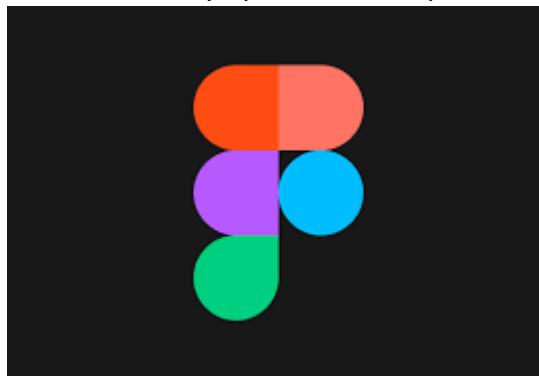


### FIGMA

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.



## DESCRIBIR LAS FÓRMULAS (MELANY)

### 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

### 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

### 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

### **Indicadores de Recursos Humanos**

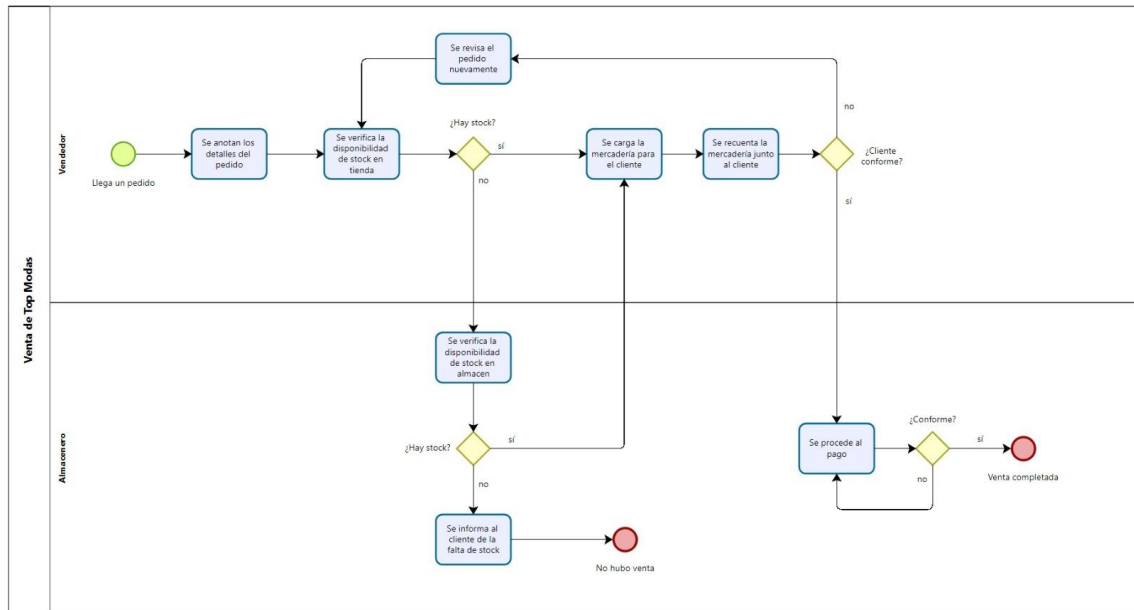
Ayudan a medir el desempeño y la satisfacción de los trabajadores, fundamental para que el equipo se mantenga productivo y pueda realizar adecuadamente sus labores.

- 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

### **Importancia de los indicadores**

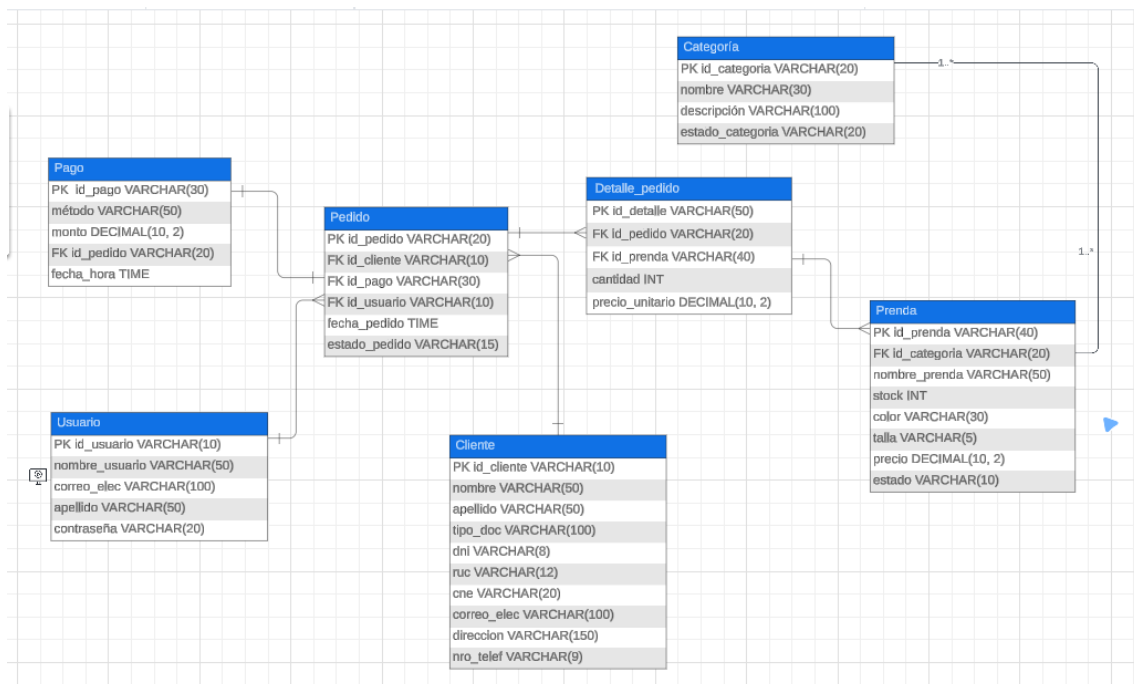
- Permiten realizar un monitoreo constante en distintas áreas de la empresa
- Brindan datos concretos para llevar a cabo una buena toma de decisiones
- Facilitan la detección de problemas en áreas que requieran algún tipo de mejora
- Ayudan a evaluar el desempeño del equipo de trabajo

## **MODELO DE PROCESOS DE NEGOCIOS (RENZO)**



## DIAGRAMA DE CLASES(LORENA)

## DIAGRAMA DE ENTIDAD RELACION (MELANY)



## DICCIONARIO DE DATOS (BRITNEY)

## **CAPITULO 04**

### **4.1 Resultados de la encuesta (PERSONA A)**

Proponer encuesta, google form, MS forms. Compañeros del curso deberan completar la encuesta

### **4.2 Presupuesto (RENZO)**

Cuanto invirtieron, y hacia adelante, cuanto se debería invertir.

### **Recomendaciones (BRITNEY)**

- Se podría desarrollar un sistema de predicción de demanda utilizando inteligencia artificial. Este sistema analizaría el comportamiento de ventas y las tendencias del mercado para anticipar la demanda de productos, optimizando la reposición de inventarios y evitando tanto el exceso de stock como la falta de productos clave.
- Incorporar un sistema de fidelización de clientes vinculado al inventario permitiría recompensar a los clientes frecuentes con descuentos o promociones automáticas basadas en el stock disponible. Esto no solo incentivaría la lealtad del cliente, sino que también contribuiría a una mayor venta de productos, especialmente aquellos que no tienen alta demanda.
- Implementar un sistema de auditoría automatizada de inventario, que realice revisiones periódicas y genere informes sobre discrepancias entre el stock. Esto permitiría detectar problemas rápidamente, como robos o errores en el registro, ayudando a mantener la precisión y la integridad del inventario.

- La implementación de un sistema de recomendaciones de productos basado en historial de ventas permitiría a Top Modas sugerir productos complementarios o adicionales a los clientes. Esto no solo aumentaría el ticket promedio de compra, sino que también ayudaría a maximizar la exposición de todo el catálogo de productos.
- Se podría implementar un sistema de alertas de productos de alta demanda que notifique al personal cuando ciertos productos están siendo vendidos rápidamente. Esto permitiría planificar de forma más efectiva la reposición de inventario, evitando quiebres de stock en artículos populares y maximizando las oportunidades de venta.
- Se puede desarrollar una plataforma para la gestión de ventas por temporada, donde se anticipen las demandas de productos en base a estaciones, eventos especiales o tendencias de moda. Esto ayudaría a preparar el inventario adecuadamente para las épocas de alta demanda, reduciendo el riesgo de agotamiento y maximizando las ventas en momentos clave.

## **Conclusiones (BRITNEY)**

En conclusión, se ha logrado implementar una base de datos eficiente para el registro y seguimiento del inventario en Top Modas. Esta base de datos permite a la empresa mantener la información de forma más precisa y detallada, facilitando un control preciso y en tiempo real de los productos disponibles. Gracias a esta solución, se reduciría los errores que se producían con los registros manuales y se ha mejorado la visibilidad de los movimientos de inventario. La empresa puede optimizar la planificación de sus reposiciones, lo que ha resultado en una gestión más eficiente de los recursos y una respuesta más fluida a la demanda del mercado.

El desarrollo de una interfaz de usuario intuitiva ha sido un componente fundamental en la mejora de los procesos internos de Top Modas. Esta interfaz permite a los empleados ya administrador registrar y actualizar el inventario de manera rápida y sencilla, minimizando la posibilidad de cometer errores durante el ingreso de datos. Además, el uso de patrones de GUI como el patrón Formulario de Autenticación, Notificaciones Modales, Validación de Entrada, Vista Previa de Imagen, etc permiten que la interfaz sea agradable en el uso de esta, la facilidad de uso reduciría considerablemente el tiempo de capacitación necesario para el personal, logrando que se adapten rápidamente al nuevo sistema.

Con la implementación de funcionalidades avanzadas de búsqueda y filtrado, Top Modas ha conseguido optimizar aún más su capacidad de gestión de inventario. Estas herramientas permiten a los empleados localizar productos de forma precisa, utilizando filtros por categorías, tallas, colores, y otros atributos. Si un empleado necesita encontrar rápidamente todos los vestidos de la tienda que son de la talla M, color azul, y que están disponibles el filtrado aceleraría el proceso de búsqueda, así como de atención al cliente. Esta mejora no solo impactaría positivamente en la experiencia del cliente, sino que también ha permitido a la empresa analizar mejor el comportamiento de su inventario y tomar decisiones más estratégicas en cuanto a las compras y distribución de productos.

Por último, el proyecto de implementación del sistema de gestión de inventario para Top Modas representa un avance significativo en la modernización de sus operaciones de la tienda. A través del desarrollo e integración de soluciones tecnológicas, la empresa lograría optimizar sus procesos, mejorar el control y facilitando la toma de decisiones estratégicas. La implementación de un sistema integral de inventario ha permitido una mayor precisión en el manejo de productos, minimizando errores y mejorando la eficiencia operativa. Esto no solo ha facilitaría la atención al cliente, sino que también contribuiría a un mejor análisis del comportamiento de ventas y reposición de stock. El proyecto proporciona a Top Modas una plataforma tecnológica muy buena que la posiciona de manera competitiva en el mercado y le permite adaptarse con mayor flexibilidad a las demandas comerciales actuales, lo que asegura un crecimiento sostenido y más eficiente en el futuro.

### **Bibliografías (PERSONA D)**

**Codecademy. (s.f.). MVC: Model, View, Controller. <https://www.codecademy.com/article/mvc>**

**Calibraint. (s.f.). Singleton Design Pattern in JavaScript. <https://www.calibraint.com/blog/singleton-design-pattern-in-javascript>**

**Balatsoukas, T. (2021, 11 de febrero). Repository pattern with javascript. DEV Community. <https://dev.to/thanasimpalatsoukas/repository-pattern-with-javascript-4nl>**

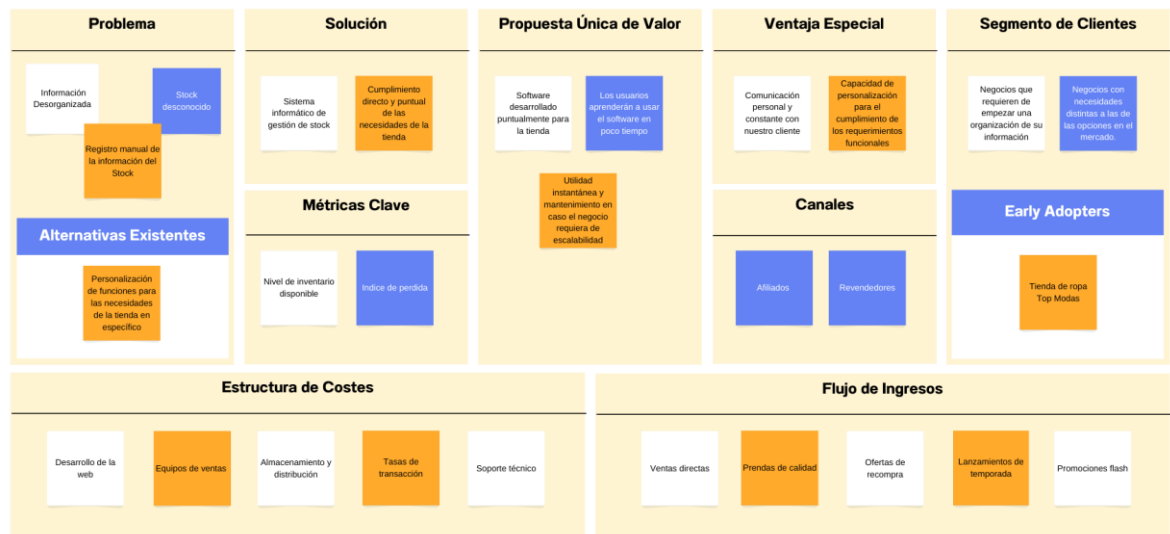
**Scheid. (2022, 15 de marzo). Using Clean Architecture and the Unit of Work pattern on a NodeJS application. DEV Community. <https://dev.to/scheid/using-clean-architecture-and-the-unit-of-work-pattern-on-a-nodejs-application-3pc9>**

**Bits, N. (2023, 24 de abril). JavaScript Factory Design Pattern: A Comprehensive Guide. Medium. <https://medium.com/@nile.bits/javascript-factory-design-pattern-a-comprehensive-guide-9266b726ee5c>**

### **Anexos (PERSONA D)**

- Modelo CANVAS

# Lean Canvas



## - Modelo de encuesta

### 1. Facilidad de uso

- ¿Cómo calificarías la facilidad de uso del sitio web en una escala del 1 al 5?
- ¿Consideras que la estructura del sitio web es fácil de navegar?
- ¿Te resultó sencillo moverte entre diferentes secciones del sitio?

### 2. Navegación

- ¿Encontraste fácilmente la información o los productos que estabas buscando?
- ¿Las categorías y subcategorías están bien organizadas?
- ¿Hay botones para regresar a secciones anteriores, como "Volver" o "Inicio"?

### 3. Diseño y apariencia

- ¿Te parece atractivo el diseño del sitio web?
- ¿Crees que los colores y tipografía visual del sitio facilitan la lectura y navegación?
- ¿Las secciones del sitio están bien organizadas y son fáciles de identificar?

### 4. Funcionalidad

- ¿El sitio web cumplió con tus expectativas en cuanto a funciones básicas como búsquedas o el proceso de compra?

- ¿Tuviste algún problema técnico al usar el sitio web?
- ¿El carrito actualiza correctamente los precios y cantidades?

## 5. Proceso de registro y compra

- ¿Cómo calificarías la experiencia de registrarte o iniciar sesión en el sitio?
- ¿Fue fácil agregar productos al carrito de compra y completar el pago?
- ¿Te pareció sencillo el proceso de revisión de la compra y confirmación de los detalles?

## 6. Accesibilidad

- ¿Los botones y enlaces están lo suficientemente espaciados para evitar clics accidentales?
- ¿El motor de búsqueda del sitio ofrece sugerencias accesibles y relevantes mientras escribes?
- ¿Los botones tienen suficiente tamaño y espacio para ser seleccionados fácilmente, incluso en dispositivos táctiles?

## 7. Satisfacción general

- ¿Cómo describirías tu nivel de satisfacción general con la experiencia en el sitio web?
- ¿Qué aspectos del sitio web consideras que podrían mejorar?
- ¿Recomendarías este sitio web a otras personas?

- Project Charter

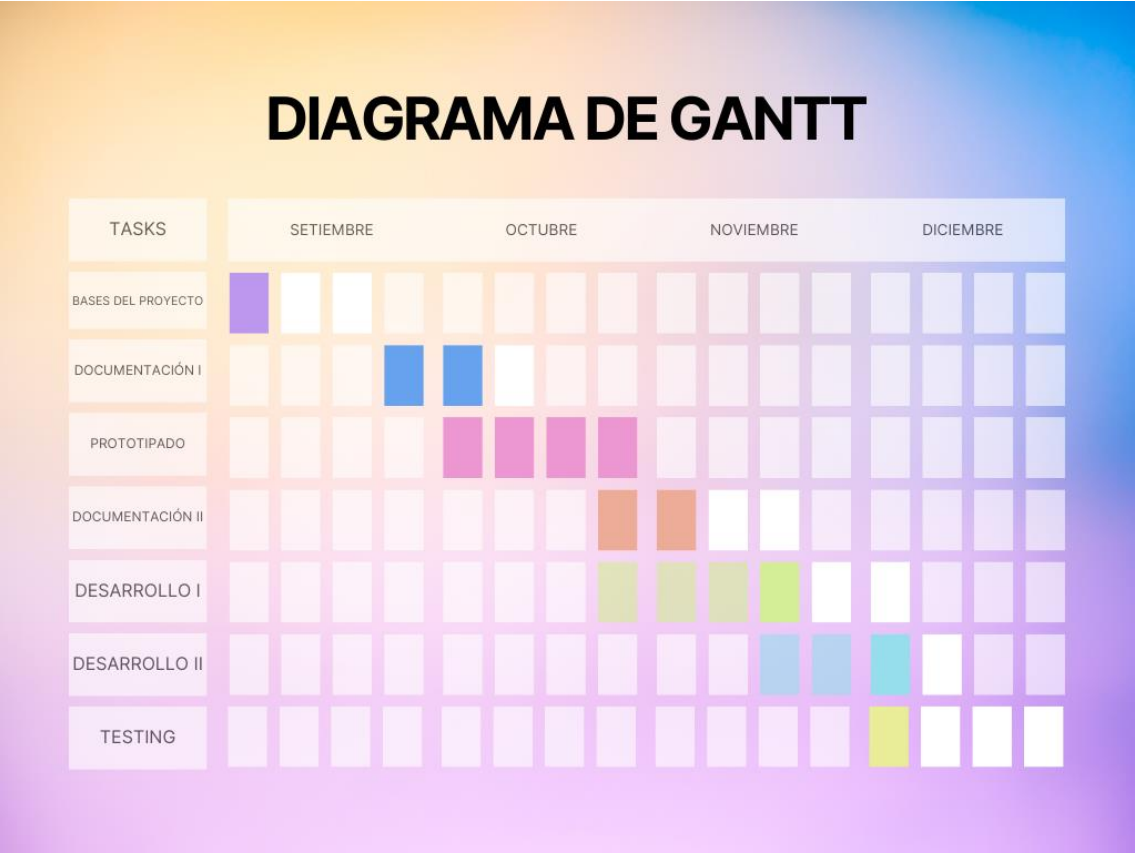
PROYECT CHARTER			
TITULO DE PROYECTO	Sistema de inventario "Top Modas"		
Fecha de inicio	06/09/2024	Gerente de proyecto	Lorena Kim Negrillo
Fecha de finalización	06/12/2024	Patrocinador del proyecto	Top Modas
Descripción y necesidad del proyecto			
El sistema de inventario para la tienda mayorista de ropa juvenil para mujeres Top Modas ayudará en mejorar la eficiencia operativa y la precisión en el manejo del stock a través del desarrollo de un sistema de gestión de inventario.			



Alcance del proyecto		Riesgos y problemas	
<div>1. Registro inicial del inventario con detalles como código de producto, descripción, cantidad disponible y ubicación.</div> <div>2. Actualización automática del inventario en tiempo real con cada venta, devolución o reposición.</div> <div>3. Generación de reportes periódicos que incluirán productos más vendidos, productos de baja rotación y alertas de productos con bajo stock.</div> <div>4. Implementación de un sistema de control de acceso para garantizar la seguridad y confidencialidad de la información, permitiendo que solo personal autorizado pueda realizar modificaciones.</div> <div>5. Alertas automáticas por correo o notificaciones en la aplicación cuando el stock alcance niveles mínimos o se detecten discrepancias en el inventario.</div>		<div>1. Problemas de adaptación del personal a la nueva herramienta.</div> <div>2. Fallos en las herramientas o software.</div> <div>3. Cambio en los requisitos del cliente.</div>	
Entregables		Supuestos	
<div>1. Sistema de gestión de inventario funcional.</div> <div>2. Sistema de alertas automáticas.</div> <div>3. Generación de reportes.</div> <div>4. Sistema de control de acceso.</div> <div>5. Documentación del proyecto.</div>		<div>1. Todos los requisitos planteados se cumplirán con el sistema.</div> <div>2. Todos los trabajadores se capacitarán y trabajarán con el sistema.</div>	
Presupuesto			
75.500 soles peruanos			
Hitos			
hito	Fecha prevista de finalización		Fecha actual
documentación de proyecto	01/11/24		01/11/24
Prototipo del proyecto	04/10/24		01/10/24
1ª fase de desarrollo	02/11/24		--
2° fase de desarrollo	01/12/24		--

Testeo del programa		06/12/24		--	
Miembros del equipo			Aprobación		
Desarrollador de software	Lorena Kim Negrillo	Patrocinador		Flor de María Mendez Rivera	
Analista de negocios Tester	Melany Daniela Carbajal Añaca	CEO		Flor de María Mendez Rivera	
Diseñador de UX Diseñador de UI	Britney Daphne Montesinos Poma				
Diseñador de base de datos	Renzo Luciano Rodriguez Martinez				

- Diagrama Gantt



**-pedazos de codigo**