



**Universidad  
Tecnológica  
del Perú**

**Facultad de Ingeniería**

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

**“Control de Ventas para la Pastelería Vlady”**

**Estudiantes:**

Duran Lee, Daniel Alejandro U20210634

Ramirez Anton, Matias Fabrizio U22102894

Solidoro Cueto, Luis Guillermo U22212266

Ruiz Freyre, Lhara Rocio U22310877

Olivera Novoa, Pedro Joaquin U22222753

**Lima - Perú**

**2025**

# ÍNDICE

ÍNDICE .....	2
ÍNDICE DE TABLAS .....	3
ÍNDICE DE FIGURAS .....	3
INTRODUCCIÓN .....	4
<b>CAPITULO 1 – ASPECTOS GENERALES .....</b>	<b>5</b>
1.1 DESCRIPCIÓN DEL PROBLEMA .....	5
<b>1.2 DEFINICIÓN DE OBJETIVOS .....</b>	<b>9</b>
1.2.1 <i>Objetivo general</i> .....	9
1.2.2 <i>Objetivos específicos</i> .....	9
<b>1.3 ALCANCE Y LIMITACIONES .....</b>	<b>9</b>
1.3.1 <i>Alcances</i> .....	11
1.3.2 <i>Limitaciones</i> .....	12
<b>1.4 JUSTIFICACIÓN .....</b>	<b>12</b>
<b>1.5 ESTADO DEL ARTE .....</b>	<b>12</b>
<b>CAPITULO 02 – MARCO TEÓRICO .....</b>	<b>17</b>
2.1 MARCO TEÓRICO DE PROGRAMACIÓN ORIENTADA A OBJETOS .....	17
2.1.1 <i>Metodología</i> .....	17
2.1.2. <i>Modelo</i> .....	21
2.1.3 <i>Patrones de GUI</i> .....	26
2.1.4 <i>Patrones de desarrollo</i> .....	29
2.1.5 <i>Lenguajes de programación / framework</i> .....	33
2.1.6 <i>Gestor de Base de Datos</i> .....	38
2.1.7 <i>Herramientas adicionales:</i> .....	39
2.2 MARCO TEÓRICO DE LAS FÓRMULAS Y/O EL SISTEMA .....	48
2.2.1 <i>Fórmulas</i> .....	48
<b>CAPÍTULO 3 - DESARROLLO DE LA SOLUCIÓN .....</b>	<b>50</b>
3.1 MODELO DE PROCESO DE NEGOCIOS .....	50
3.2 PROTOTIPOS .....	51
3.3 DIAGRAMA DE CLASES: .....	53
3.4 MODELAMIENTO DE BASE DE DATOS: .....	54
3.4.1 <i>Diccionario de datos</i> .....	55
3.5 ENTORNO VISUAL DE LOS FORMULARIOS .....	57
<b>REFERENCIAS BIBLIOGRÁFICAS: .....</b>	<b>60</b>

## ÍNDICE DE TABLAS

<b>Tabla 01: Requerimiento funcional - Generación Ticket de Venta.....</b>	<b>8</b>
<b>Tabla 02: Requerimiento funcional - Consultamiento de Stock.....</b>	<b>8</b>
<b>Tabla 03: Requerimiento funcional - Registro de Ventas.....</b>	<b>9</b>
<b>Tabla 04: Requerimiento funcional - Reporte de Ventas General.....</b>	<b>9</b>
<b>Tabla 05: Requerimiento funcional - Reporte de Ventas de Cliente.....</b>	<b>9</b>
<b>Tabla 06: Requerimiento Funcional – Registro de Métodos de Pagos .....</b>	<b>9</b>
<b>Tabla 07: Requerimiento Funcional – Registro de Clientes .....</b>	<b>9</b>
<b>Tabla 08: Requerimiento Funcional – Login Seguro .....</b>	<b>10</b>
<b>Tabla 09: Requerimiento no funcional - Facilitación de Uso.....</b>	<b>10</b>
<b>Tabla 10: Requerimiento no funcional - Tiempo de Reacción.....</b>	<b>10</b>
<b>Tabla 11: Requerimiento no funcional – Seguridad.....</b>	<b>10</b>
<b>Tabla 12: Requerimiento no funcional – Disponibilidad.....</b>	<b>11</b>

## ÍNDICE DE FIGURAS

**NO SE ENCONTRARON ENTRADAS DE TABLA DE CONTENIDO.**

## Introducción

En los últimos años, la Pastelería Vlady ha experimentado un crecimiento notable, tanto en su volumen de ventas como en la variedad de productos que ofrece. Reconocida por su calidad artesanal y la personalización de sus postres, la pastelería ha logrado consolidar una clientela fiel, al mismo tiempo que atrae nuevos consumidores mediante recomendaciones y presencia en redes sociales. Este incremento en la demanda ha puesto en evidencia la necesidad de modernizar sus procesos comerciales para responder con mayor agilidad, orden y eficiencia.

Hasta ahora, la gestión de ventas en la pastelería se ha realizado de forma manual: los pedidos se registran en hojas de Excel o en libretas físicas, y la comunicación con los clientes se lleva a cabo por llamadas o mensajes dispersos. Si bien este método ha sido funcional en etapas tempranas del negocio, el ritmo actual de pedidos y la exigencia por entregas puntuales y personalizadas requieren un sistema más robusto. La falta de un control centralizado puede generar confusiones, retrasos, duplicaciones de pedidos o

incluso pérdidas de información, afectando directamente la experiencia del cliente y limitando el potencial de crecimiento del negocio.

## **Capítulo 1 – Aspectos Generales**

### **1.1 Descripción del Problema**

Hoy en día, la digitalización y el uso de sistemas automatizados han demostrado ser elementos importantes para la optimización de los procesos empresariales, mejorando la eficiencia y reduciendo errores operativos. Sin embargo, muchas pequeñas y medianas empresas todavía tienen desafíos significativos, ya que no cuentan con herramientas tecnológicas adecuadas para gestionar sus operaciones. Como es el caso de la Pastelería Vlady, un negocio dedicado a la venta de productos de panadería y repostería. Aún teniendo una clientela estable y ofrecer una variedad de productos, la empresa tiene dificultades para controlar sus ventas, lo que impacta en su eficiencia operativa y en la satisfacción de los clientes.

En la actualidad, la pastelería no cuenta con un sistema digital de control de ventas, obligando a los empleados a registrar las transacciones de manera manual en libretas o archivos de Excel. Este método presenta múltiples desventajas, como la posibilidad de errores en el registro de pedidos, dificultades en la gestión del inventario y una lenta actualización de las ventas, lo que puede llevar a errores como la venta de productos sin stock o el desconocimiento de las cantidades exactas disponibles. Además, la falta de un sistema automatizado impide que la empresa tenga un control eficiente sobre los ingresos y egresos diarios, dificultando la toma de decisiones basada en datos precisos. En consecuencia, el negocio enfrenta pérdidas económicas debido a errores humanos en los registros, demoras en la atención al cliente y falta de información confiable para planificar la producción.

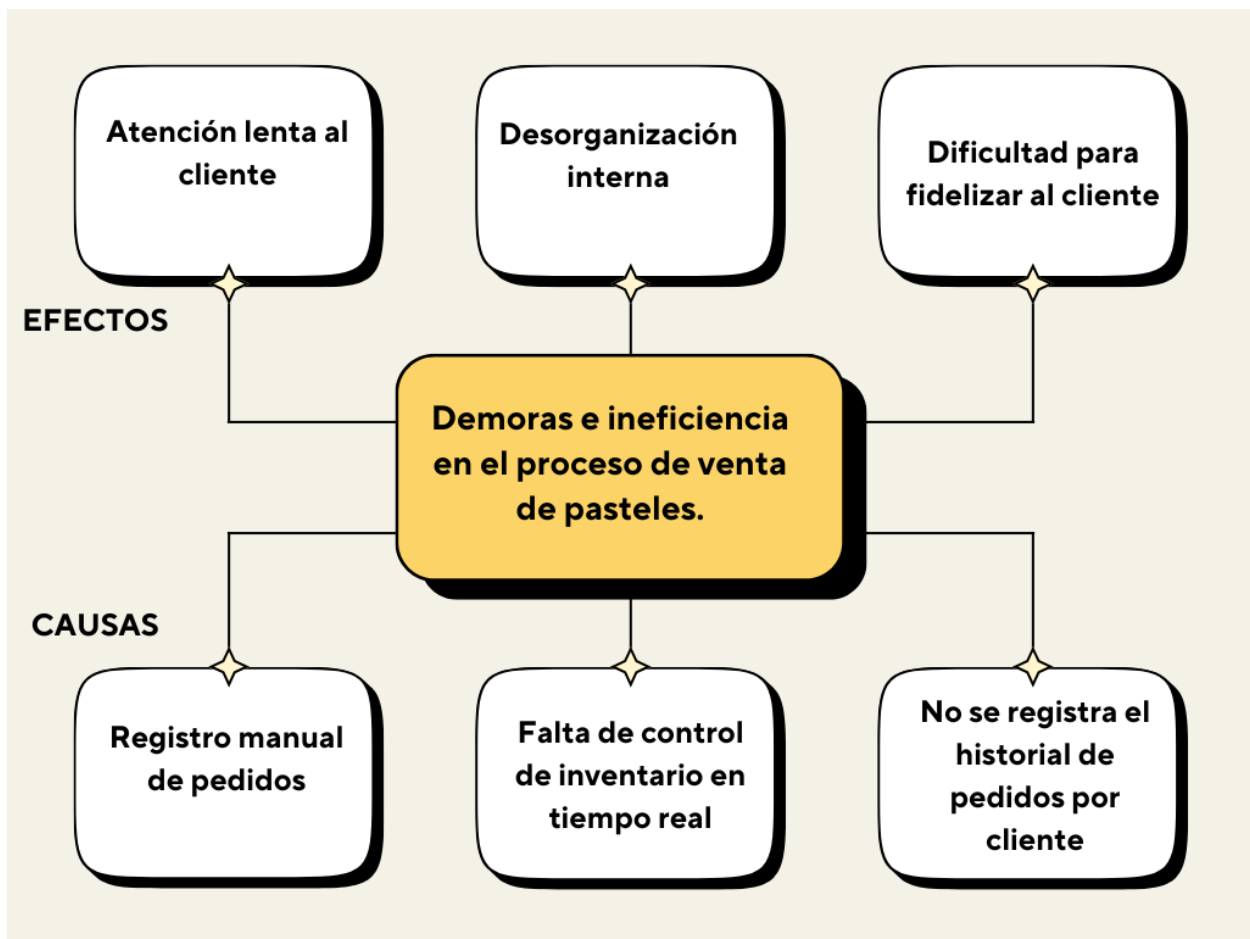
Actualmente, cuando un cliente desea comprar un pastel en la pastelería Vlady, el proceso de venta se inicia con la atención directa del vendedor, quien debe registrar el pedido de forma manual en una libreta o formato físico, anotando detalles como el tipo de pastel, tamaño, fecha de entrega y datos del cliente. Esta forma de registro es lenta, propensa a errores y poco segura, ya que puede extraviarse fácilmente. Luego, el vendedor debe verificar físicamente si el producto solicitado se encuentra disponible en stock, lo que implica dejar al cliente esperando mientras se revisa en el área de almacenamiento o exhibición. En caso de que el pastel no esté disponible, el vendedor debe comunicarse verbalmente o trasladarse hasta el área de producción para consultar la posibilidad de elaboración, el tiempo estimado y la carga de trabajo del personal. Este paso no solo demora la atención, sino que también interrumpe la operación en producción, generando un flujo desordenado e ineficiente.

Una vez obtenida la respuesta del área de producción, el vendedor regresa para informar al cliente si su pedido puede realizarse, lo cual no siempre es inmediato ni exacto debido a la falta de coordinación y visibilidad del estado del inventario. Si el cliente acepta las condiciones, procede a realizar el pago, que también se registra manualmente, dificultando el control financiero y la generación de comprobantes. Todo este proceso, desde el registro inicial hasta la confirmación y pago, depende de una gestión manual entre varias áreas que no están conectadas entre sí, lo que incrementa las demoras, reduce la eficiencia y puede afectar negativamente la experiencia del cliente.

En algunos casos, la falta de un control estructurado lleva a que ciertos productos se agoten sin previo aviso, afectando la experiencia del cliente y reduciendo las oportunidades de venta. Para abordar estos problemas, es fundamental la implementación de un Sistema de Control de Ventas que permita automatizar el registro de transacciones, optimizar la gestión del inventario y mejorar la precisión en la toma de decisiones. Un sistema digital contribuirá a una operación más eficiente, garantizando una atención más rápida y precisa para los clientes, al mismo tiempo que facilitará el crecimiento sostenible de la pastelería en un mercado cada vez más competitivo.

## **Diagrama de Árbol**

**Figura 1 Diagrama de árbol de la pastelería Vlady**



Fuente: Elaboración propia

### Indicadores de gestión:

**1. Venta promedio:** Es la cantidad de gasto promedio que realiza un cliente al comprar

Formulas:

$$Ticket\ Diario = \frac{Ingresos\ al\ dia}{Ventas\ diarias}$$

$$Ticket\ Semanal = \frac{Ingresos\ Semanales}{Ventas\ Semanales}$$

El objetivo es aumentar la cantidad promedio de ventas que realizan los clientes tanto de forma diaria como de forma semanal.

**2. Stock de productos:** Se encarga de medir la cantidad de productos disponibles.

Formula:

$$Stock\ Disponible = Stock\ inicial - ventas$$

El objetivo es asegurar que la pastelería tenga un mínimo aceptable de unidades de cada producto.

**3. Tiempo de atención:** Se encarga de medir cuanto tiempo se tarda en completar una venta cuando un cliente hace su pedido.

Formula:

$$Promedio\ diario = \frac{\sum(Tiempo\ de\ venta\ diaria)}{Ventas\ diarias}$$

Su objetivo es reducir el tiempo de los clientes cuando realizan su pedido

**4. Ingresos de ventas:** Se encarga de medir cuánto dinero a ganado la pastelería.

Formulas:

$$Ingresos\ diarios = \sum (Precio\ de\ cada\ producto\ vendido\ al\ dia)$$

$$Ingresos\ semanales = \sum (Precio\ de\ cada\ producto\ vendido\ a\ la\ semana)$$

Su objetivo es generar una cantidad de dinero significativa para que la pastelería siga siendo un negocio rentable.



## 1.2 Definición de objetivos

### 1.2.1 Objetivo general

Agilizar la atención al cliente y centralizar la información de venta en la pastelería Vlady mediante el desarrollo e implementación de un software de gestión.

### 1.2.2 Objetivos específicos

- Digitalizar el registro de pedidos.
- Implementar control de inventario en tiempo real.
- Registrar historial de pedidos por cliente.

## 1.3 Alcance y limitaciones

### Requerimientos funcionales:

Tabla 01: Requerimiento funcional - Generación Ticket de Venta

Nro. De Requerimiento		RF01	
Nombre		Ticket de Venta	
Usuario	Vendedor	Prioridad	Media
El sistema tiene que generar de forma automática un ticket de venta cuando termina la transacción, el ticket se podrá imprimir o guardarse de forma automática.			

Fuente: Elaboración propia

Tabla 02: Requerimiento funcional - Consultamiento de Stock

Nro. De Requerimiento		RF02	
Nombre		Stock de Productos	
Usuario	Vendedor/Admin	Prioridad	Alta
El sistema permitirá la visualización del stock de los productos en tiempo real, usando que se tengo sobre su cantidad y descripción.			

Fuente: Elaboración propia

Tabla 03: Requerimiento funcional - Registro de Ventas

Nro. De Requerimiento		RF03	
Nombre		Registro de ventas	
Usuario	Vendedor	Prioridad	Alta
El sistema va a permitir que se registren ventas al ingresar los datos del cliente, el producto(s), el precio y el medio de pago			

Fuente: Elaboración propia

Tabla 04: Requerimiento funcional - Reporte de Ventas General

Nro. De Requerimiento		RF04	
Nombre		Reporte de ventas General	
Usuario	Administrador	Prioridad	Media
El sistema va a generar reportes de las ventas al día, semana y mes. Estos reportes van a incluir la cantidad de transacciones realizada, y el total de ingresos.			

Fuente: Elaboración propia

Tabla 05: Requerimiento funcional - Reporte de Ventas de Cliente

Nro. De Requerimiento		RF04	
Nombre		Reporte de ventas de cliente	
Usuario	Administrador	Prioridad	Media
El sistema va a generar reportes sobre las transacciones y pedidos de los clientes a lo largo de un periodo de tiempo.			

Fuente: Elaboración propia

Tabla 06: Requerimiento Funcional – Registro de Métodos de Pagos

Nro. De Requerimiento		RF05	
Nombre		Registro de Métodos de Pagos	
Usuario	Administrador	Prioridad	Alta
Permite el acceso rápido a el registro de pagos en efectivo, tarjeta u otros medios, para llevar un mejor orden con los métodos de pago.			

Fuente: Elaboración propia

Tabla 07: Requerimiento Funcional – Registro de Clientes

Nro. De Requerimiento		RF06	
Nombre		Registro de Clientes	
Usuario	Vendedor	Prioridad	Baja
Permitir registrar clientes frecuentes para un seguimiento y promociones.			

Fuente: Elaboración Propia

Tabla 08: Requerimiento Funcional – Login Seguro

Nro. De Requerimiento		RF06	
Nombre		Login Seguro	
Usuario	Vendedor	Prioridad	Baja
Permitir ingresar al sistema únicamente mediante el ingreso de un usuario y contraseña previamente creada.			

Fuente: Elaboración Propia

### Requerimientos no Funcionales:

Tabla 09: Requerimiento no funcional - Facilitación de Uso

Nro. De Requerimiento		RNF01	
Nombre		Interfaz fácil de usar	
Usuario	Todos	Prioridad	Alta
La interfaz del sistema tiene que ser fácil de usar y debe ser intuitiva, para que no sea complicada para los usuarios.			

Fuente: Elaboración propia

Tabla 10: Requerimiento no funcional - Tiempo de Reacción

Nro. De Requerimiento		RNF02	
Nombre		Tiempo de reacción	
Usuario	Vendedor	Prioridad	Alta
La respuesta a las acciones del usuario tiene que ser de unos pocos segundos, para evitar demoras en las acciones del sistema.			

Fuente: Elaboración propia

Tabla 11: Requerimiento no funcional – Seguridad

Nro. De Requerimiento		RNF03	
Nombre		Seguridad	
Usuario	Administrador	Prioridad	Alta
Información y acceso al sistema restringido a personas autorizadas.			

Fuente: Elaboración propia

Tabla 12: Requerimiento no funcional – Disponibilidad

Nro. De Requerimiento		RNF04	
Nombre		Disponibilidad	
Usuario	Admin/Vendedor	Prioridad	Media
El sistema debe estar disponible el 99% del tiempo durante horarios laborales, para así poder tener más eficacia al momento de realizar las tareas correspondientes con el sistema.			

Fuentes: Elaboración Propia

#### 1.3.1 Alcances

- Desarrollo de un sistema de ventas para registrar pedidos de forma rápida y eficiente.
- Implementación de un módulo de control de inventario en tiempo real.
- Registro y gestión del historial de pedidos por cliente.

- Generación de reportes de ventas diarios, semanales y mensuales.
- Control de stock automático al realizar una venta.
- Interfaz amigable y de fácil uso para el personal.
- Acceso restringido según niveles de usuario (por ejemplo, ventas vs. administrador).
- Posibilidad de consultar productos disponibles y su stock actualizado.
- Sistema diseñado para operar en local sin necesidad de conexión constante a internet.
- Base de datos local que permita respaldo y recuperación de información.

### 1.3.2 Limitaciones

- No se incluirá un sistema de facturación electrónica validado por SUNAT.
- El sistema no contará con una plataforma web o aplicación móvil en esta versión inicial.
- No se integrará con medios de pago electrónicos (POS, Yape, Plin, etc.).
- La carga inicial de datos (productos, stock, clientes) deberá hacerse mediante digitación en el sistema.
- La interfaz será desarrollada en una sola estación de trabajo (no multiusuario en red).
- El mantenimiento del sistema dependerá del usuario final o de soporte externo, ya que no se incluye un módulo de autodiagnóstico o actualización automática.

## 1.4 Justificación

La necesidad de digitalizar los procesos de ventas e inventario en la Pastelería Vlady surge ante las constantes dificultades en la gestión manual de pedidos y control de stock. Actualmente, estas tareas se realizan de forma escrita o mediante registros simples, lo que ocasiona errores humanos, pérdida de información, demoras en la atención al cliente y una visión limitada sobre las existencias reales del negocio.

Con la implementación del sistema propuesto, se espera reducir el tiempo de atención, evitar pérdidas por errores de stock, mejorar la fidelización del cliente y ofrecer una experiencia de compra más rápida y ordenada. Además, permitirá analizar el comportamiento de compra de los clientes para ofrecerles productos personalizados o promociones basadas en su historial. En resumen, este sistema no solo optimiza los procesos internos, sino que fortalece la relación con el cliente y contribuye al crecimiento sostenible del negocio.

## 1.5 Estado del Arte

En el contexto peruano, muchos pequeños y medianos negocios, especialmente en el rubro gastronómico como panaderías y pastelerías, aún operan con procesos manuales o semi-manuales en la gestión de ventas e inventarios. Esta situación genera múltiples inconvenientes: pérdidas por productos vencidos o mal registrados, errores en los pedidos, duplicación de tareas y escasa trazabilidad de las operaciones. Todo esto repercute negativamente en la eficiencia del negocio y en la experiencia del cliente.

Si bien existen soluciones comerciales de software como POS (Point of Sale) nacionales e internacionales —por ejemplo, sistemas como Openpay, VendéMás, Izipay, Niubiz, Bsale o Tango, muchos de ellos presentan barreras significativas para su adopción por parte de microempresas peruanas: altos costos de licencia, necesidad de conexión constante a internet, interfaz compleja o no adaptada al modelo de negocio local, y poco acompañamiento en la implementación.

En respuesta a esta problemática, algunos negocios han optado por desarrollar sus propias soluciones a medida o utilizar herramientas básicas como Excel. Sin embargo, estas alternativas suelen carecer de integración en tiempo real, automatización del stock y capacidad de análisis histórico, funciones clave para una gestión moderna y competitiva.

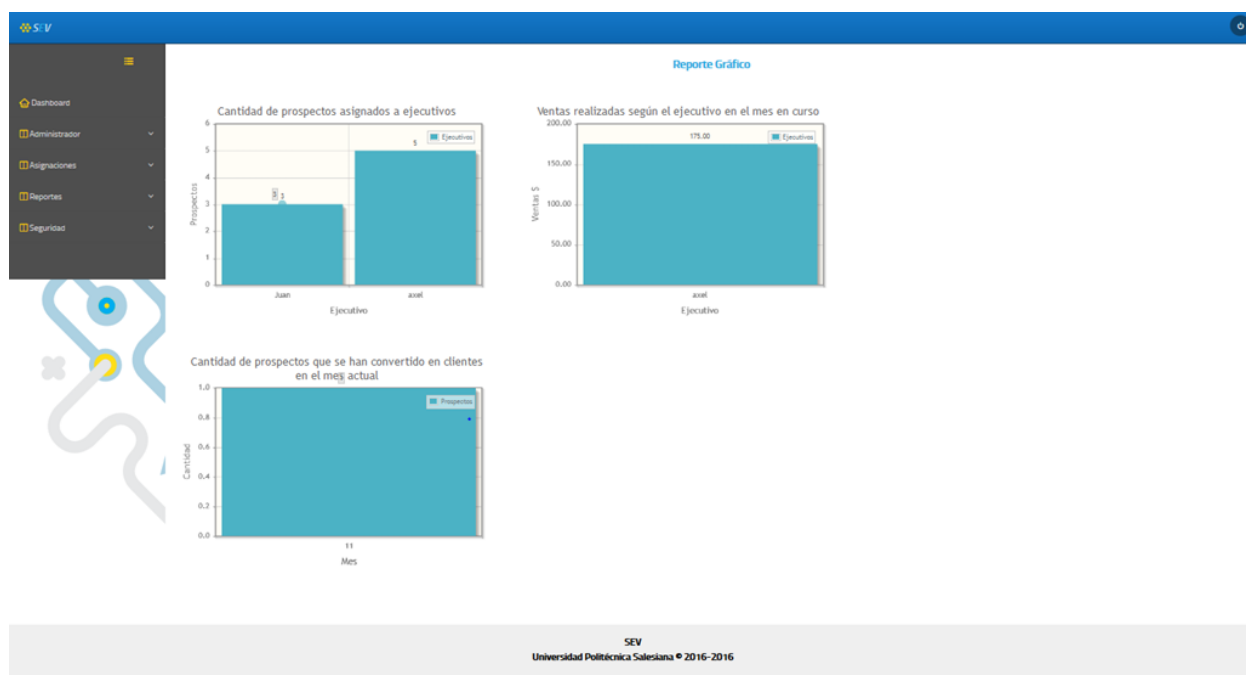
Este proyecto se diferencia al enfocarse específicamente en las necesidades de una pastelería real del medio local, priorizando una solución ligera, personalizable, intuitiva y que pueda ser utilizada sin requerimientos técnicos avanzados. Así, se busca aportar no solo a la mejora operativa del negocio en cuestión, sino también a generar un modelo replicable para otras pequeñas empresas peruanas con realidades similares.

A continuación, se mostrará algunos de los proyectos que existen en la actualidad que sirven como referencia para el desarrollo del sistema deseado:

### **Desarrollo e Implementación de Sistema de Administración de Ventas para Humane Escuela de Negocios para optimizar los procesos de gestión de ventas de servicios académicos**

Este proyecto se encontró con un problema similar al que nosotros nos encontramos, teniendo que lidiar con información que es ingresada de forma manual por sus diferentes usuarios generando un inconveniente para la verificación de la veracidad de información. Mediante el uso de lenguajes de programación como Java, frameworks como Java Server Faces y la metodología MVC (Modelo Vista Controlador) se desarrolló un sistema que con una interfaz de usuario simple e intuitiva se pueda agilizar el proceso de ventas manteniendo al mismo tiempo un control más exacto de la información requerida.

**Figura 2 Ventana de Reporte Grafico de Sistema de Administración de Ventas para Humane Escuela de Negocios**



Latorre, A., & Castañeda, J. (2016). *Desarrollo e implementación de sistema de administración de ventas para Humane Escuela de Negocios*. Universidad Politécnica Salesiana, Ecuador.

### Implementación de un sistema automatizado para el control de ventas y facturación en la panadería Tijerino del municipio de la Trinidad

Este trabajo, en particular, se usó como referencia gracias a la amplia similitud entre ambos negocios. Ambas empresas pertenecen a un contexto familiar que, al utilizar procesos totalmente manuales, reducían la fiabilidad de estos. La panadería registraba ventas y disponibilidad de tanto mercadería como insumos en libretas, lo cual generaba muchos errores, pérdidas económicas e incluso falta de stock. El nuevo sistema permitió registrar las ventas, controlar mucho mejor el inventario e incluso generar reportes de la entrada y salida de dinero. Gracias a esto, nuestro trabajo pudo alinearse y seguir el ejemplo de este proyecto, pues se desarrolla en el mismo entorno gastronómico y los problemas a solucionar son casi los mismos.

**Figura 3 Formulario de Clientes de Panadería Tijerino**

**Explorador de Clientes**

BUSCAR CLIENTES

**Datos del Cliente**

Código: 8

Nombre:

Apellidos:

Email:

DNI:

Teléfono:

Estado: disponible

Dirección:

**Estado clientes**

☒ Disponibles ☐ No Disponibles

ID	Nombres	Apellidos	Dirección	Teléfono	Email	Documento	Estado
1	Público	en general					disponible
2	Ana	Trujillo	caridad	5555-4729			disponible
3	Antonio	Moreno	caridad	7555-3932			disponible
4	Jony	Thomas Hardy	canta rana	8555-7788		74589652	disponible
5	Douglas	Martinez	Colombia	75845544			disponible
6	Cristina	Hernandez	Caridad	87544322			disponible
7	Pedro	Frédérico Lopez	San Antonio	88.60.15.31		56462123	disponible

**AGREGAR** **GUARDAR** **EDITAR** **seleccionar** **CANCELAR**

Ilustración 9: Formulario clientes.

González, C., & Tijerino, J. (2020). *Implementación de un sistema automatizado para el control de ventas en la Panadería Tijerino*. Universidad Nacional Autónoma de Nicaragua.

### **Diseño y aplicación de sistema para punto de venta**

Por otro lado, este proyecto se centra más en el diseño y organización del sistema, específicamente uno de ventas tal como el que se desarrolla en este informe. Aunque no se relaciona con el mismo ámbito culinario, este es aplicado a negocios familiares o minoristas tal como la pastelería Vlady. El autor detalla la construcción del sistema enfocado, especialmente, en la estructura de la base de datos. Tablas como productos, ventas, clientes y demás lo que facilita la obtención de reportes y el análisis de los datos. Esto nos ayudó a mantener la organización y tener una idea clara sobre la estructura de un modelo o sistema de ventas, aspectos que se han considerado para el diseño de este sistema.

### **Figura 4 Relación de tablas uno a uno**



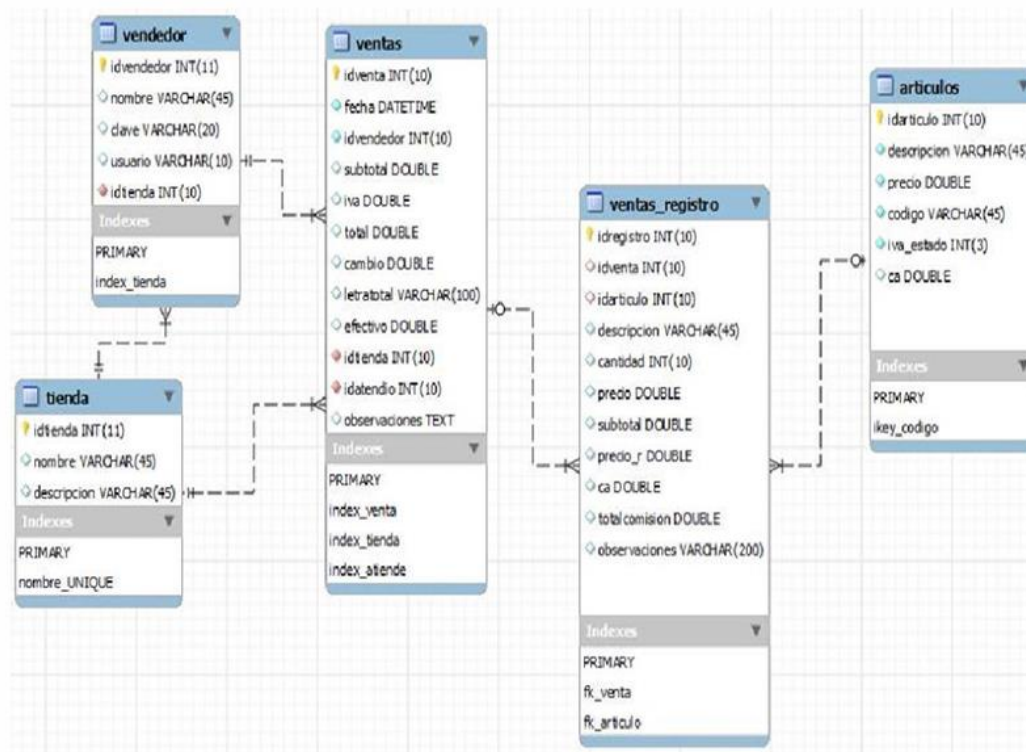


Figura 3.7. Relación de tablas de uno a uno.

García, L. A. (2014). *Diseño y aplicación de sistema para punto de venta*. Instituto Politécnico Nacional, México.

## Capítulo 02 – Marco Teórico

### 2.1 Marco teórico de programación orientada a objetos

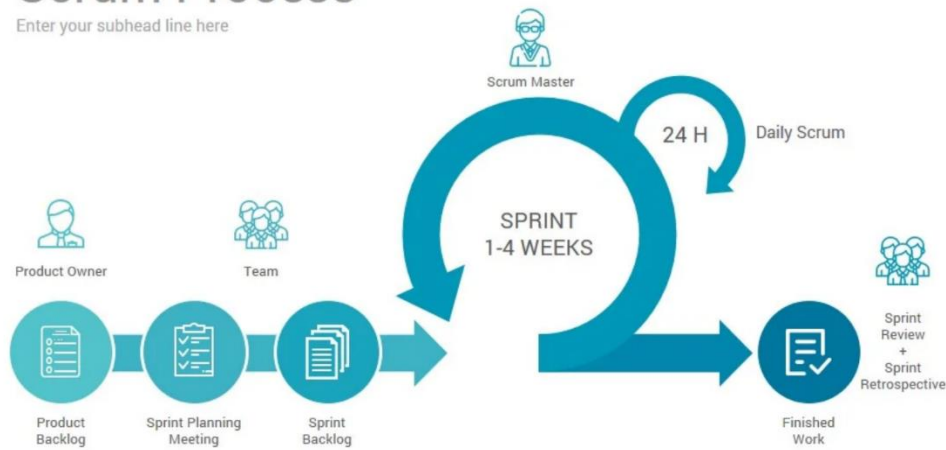
#### 2.1.1 Metodología

#### SCRUM

#### Figura 5 – Proceso Scrum

# Scrum Process

Enter your subhead line here



Fuente: <https://webdesigncusco.com/que-es-scrum-y-para-que-sirve/>

## Planificación – Product Backlog

En esta fase se construye y mantiene el Product Backlog, una lista priorizada de tareas y funcionalidades necesarias para el proyecto. El Product Owner, en conjunto con el equipo de desarrollo, identifica los elementos más importantes a trabajar.

A diferencia de métodos tradicionales, Scrum permite que el Product Backlog evolucione constantemente, adaptándose según se adquiere conocimiento durante el desarrollo. También es crucial la colaboración con los stakeholders para asegurar que las expectativas del cliente y las necesidades del negocio estén bien representadas. La comunicación continua es clave para que todo el equipo tenga una comprensión compartida de los objetivos.

## Ejecución – Sprint

El Sprint es el núcleo operativo de Scrum. Es un ciclo de trabajo de duración fija (generalmente entre 1 y 4 semanas), durante el cual el equipo desarrolla un producto funcional que podría ser entregado al cliente.

Antes de iniciar el Sprint se realiza una Sprint Planning, donde el equipo selecciona tareas del Product Backlog, estimando qué pueden completar durante ese periodo. Al final del Sprint, se llevan a cabo dos reuniones fundamentales: la Sprint Review, donde se presenta el trabajo realizado, y la Sprint Retrospective, en la que el equipo analiza cómo mejorar su proceso de trabajo en el futuro.

## **Control y Monitorización – Daily Scrum y Burn Down Chart**

Durante el Sprint, el equipo se reúne diariamente en una reunión breve conocida como Daily Scrum (o daily stand-up). Esta reunión permite sincronizar esfuerzos, detectar impedimentos y ajustar tareas rápidamente.

El Burn Down Chart es una herramienta visual que muestra cuánto trabajo queda por hacer en relación con el tiempo restante. Es útil para monitorear el avance real frente al planificado y tomar decisiones correctivas si es necesario. El uso de herramientas digitales complementarias puede mejorar aún más la visibilidad del progreso.

## **Revisión y Adaptación – Sprint Review y Retrospective**

Al finalizar el Sprint, se realiza una Sprint Review para evaluar el producto entregado, obtener retroalimentación de los stakeholders y, si es necesario, ajustar el Product Backlog. Luego, el equipo realiza una Sprint Retrospective para reflexionar sobre su desempeño, identificar problemas y proponer acciones de mejora.

Esta fase refuerza el principio de mejora continua, fomentando una cultura de aprendizaje, adaptación y colaboración. El objetivo es que cada nuevo Sprint sea más eficiente y productivo que el anterior.

El enfoque de la metodología SCRUM ayuda a gestionar el trabajo de forma ágil a través de eventos, artefactos y procesos que guían al equipo a cumplir sus objetivos.

Fuente: <https://www.apd.es/metodologia-scrum-que-es/>

### **1. Sprint:**

Ciclo de trabajo repetitivo, de unas 2 a 4 semanas, donde el equipo desarrolla un aumento del sistema que se entrega y se prueba. Todos los eventos de scrum se desarrollan dentro de este ciclo, cada uno tiene un objetivo definido y tareas tomadas de Product Backlog.

### **2. Sprint Planning:**

Antes de empezar cada sprint, se hace una reunión de planificación donde el equipo define:

- El objetivo de Sprint.
- Las funciones que desarrollara.
- La organización del trabajo, dividiendo tareas grandes en subtareas más sencillas.

### **3. Daily SCRUM:**

Se realiza una reunión corta para sincronizar los esfuerzos del equipo. Se responden estas preguntas en la reunión:

- ¿Que hice ayer para avanzar hacia el objetivo del sprint?
- ¿Qué hare hoy?
- ¿Hay algo que me bloquee?

### **4. Sprint Review:**

Al terminar el sprint, el equipo muestra el incremento del sistema a los stakeholders, también incluyendo al representante de la pastelería Vlady. Sirve para mostrar lo desarrollado, recibir retroalimentación valiosa por parte de los interesados y revisar si las metas del producto se están cumpliendo en base a las expectativas del negocio.

### **5. Sprint Retrospective:**

Al terminar la revisión, el equipo hace una retrospectiva para reflexionar sobre el Sprint que a terminado. Se identifican los aspectos que funcionaron bien, los que se podrían mejorar y se proponen ajustes que puedan aumentar la eficiencia del equipo en futuros Sprints. Esta práctica continua de mejora es fundamental para mantener un desarrollo ágil y adaptable.

### **6. Product Backlog:**

Es una lista priorizada de todos los requerimientos y mejoras necesarias para el sistema. El Product Owner de la pastelería es quien se encarga de mantener este backlog organizado y actualizado, tomando en cuenta las necesidades operativas del negocio y la retroalimentación recibida tras cada sprint.

### **7. Sprint Backlog:**

Conjunto de elementos seleccionados del product backlog que el equipo decide desarrollar durante un sprint específico. Esta lista se convierte en la guía diaria de trabajo del equipo, y puede ajustarse a medida que se avanza el sprint, siempre que no se comprometa el objetivo planteado.

Tabla 13 : Comparativa de Metodologías

Aspecto	SCRUM	Kanban	Waterfall
Flexibilidad en Cambios	<b>Máxima flexibilidad:</b> Gracias a los <b>Sprints</b> , se pueden realizar cambios rápidos en cada ciclo, lo que es clave para adaptarse a la demanda creciente de la pastelería.	Flexible, pero <b>sin estructura de ajuste continuo</b> : No hay fases definidas, lo que puede dificultar ajustes rápidos como los de SCRUM.	<b>Baja flexibilidad:</b> Cambios solo en fases específicas del proyecto, lo que hace difícil adaptarse rápidamente a nuevas demandas o problemas.
Visibilidad y Control	<b>Alta visibilidad:</b> Las <b>Daily Standups</b> y revisiones de <b>Sprints</b> aseguran que todos los miembros estén alineados y que el proyecto se pueda ajustar rápidamente.	Buena visibilidad del flujo de trabajo, pero <b>menos estructura</b> en términos de seguimiento constante y ajustes.	<b>Muy baja visibilidad:</b> No se pueden ver avances hasta el final del proyecto, lo que hace difícil ajustar el proceso a tiempo.
Colaboración del Equipo	<b>Colaboración continua:</b> El equipo trabaja de manera constante en ciclos cortos, lo que promueve la interacción y mejora la comunicación en todas las áreas.	Colaboración más <b>suave</b> y menos estructurada; menos reuniones formales que garantizan que todos estén alineados en todo momento.	<b>Colaboración limitada:</b> La mayoría de la colaboración se da al inicio y al final del proyecto, lo que puede llevar a desalineaciones durante el proceso.
Tiempo de Implementación	<b>Rápido y optimizado:</b> Los <b>Sprints cortos</b> permiten implementaciones más ágiles, lo que asegura que la pastelería pueda obtener beneficios tangibles más rápido, como la automatización del registro de pedidos.	<b>Flujo constante</b> , pero menos enfocado en entregas rápidas. Los cambios importantes pueden tardar más en ser implementados.	<b>Lento y rígido:</b> La falta de flexibilidad hace que el proceso de implementación sea más largo y complicado, lo que no es ideal para negocios con un ritmo acelerado como el de Vlady.
Adaptación a Crecimiento	<b>Ideal para negocios en expansión:</b> SCRUM permite ajustar el sistema rápidamente a medida que la demanda de la pastelería crece, asegurando que la tecnología soporte el crecimiento de manera fluida.	<b>Adaptable, pero menos estructurado:</b> Aunque se puede ajustar el flujo de trabajo, no se maneja tan bien el crecimiento rápido de manera estructurada como SCRUM.	<b>No adecuado para el crecimiento rápido:</b> Waterfall es adecuado solo para proyectos con requisitos fijos; su rigidez no permite adaptarse bien a cambios constantes o crecimiento acelerado.

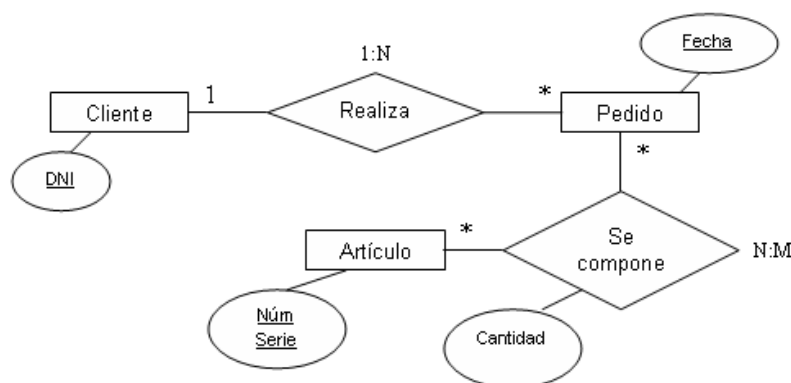
Fuente: Elaboración propia

### 2.1.2. Modelo

#### Modelo ER(Entidad-Relación)

Es un concepto más amplio que se refiere a la metodología o el esquema conceptual que define las entidades, sus atributos y las relaciones entre ellas. El modelo puede ser representado visualmente a través de un diagrama, pero el modelo en sí también puede estar presente en forma de una descripción abstracta que no necesariamente tiene que estar representada gráficamente.

**FIGURA 6 - MODELO ER**



Fuente: [https://es.wikipedia.org/wiki/Modelo\\_entidad-relaci%C3%B3n](https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n)

## Diagrama Entidad-Relación

Un diagrama entidad-relación ilustra cómo se conectan diferentes entidades, tales como personas, objetos o conceptos. Este tipo de modelo conceptual de datos es útil para que desarrolladores y diseñadores comprendan las relaciones entre los elementos clave dentro de un software.

Los diagramas ER fueron creados por el informático Peter Chen en la década de 1970 y son comúnmente utilizados por las empresas para el diseño de bases de datos. Sin embargo, no solo se limitan a representar datos; también pueden mostrar las interacciones entre otros componentes de una organización, como roles dentro de la empresa (por ejemplo, la relación entre un product manager y un desarrollador), bienes tangibles (productos o servicios) o intangibles (como el product backlog).

Este diagrama es un tipo de diagrama de flujo, proporcionando un modelo lógico que describe cómo los datos se trasladan de una entidad a otra. Gracias a su formato sencillo, los desarrolladores y diseñadores pueden obtener una visión clara de la estructura del sistema.

Usar un diagrama entidad-relación te permite comprender mejor la organización de tus datos. Al tener una representación visual, es más fácil identificar las relaciones entre las entidades y tomar decisiones más informadas sobre su gestión. Si deseas obtener más detalles sobre cómo funcionan estos diagramas, o incluso crear uno tú mismo, puedes hacerlo fácilmente de forma online.

## Símbolos en un diagrama entidad-relación

Un diagrama ER utiliza una variedad de símbolos para representar las entidades y sus relaciones. Algunos de los más comunes son:

- **Rectángulos:** Representan las entidades principales, y dentro de estos pueden aparecer entidades secundarias (también llamadas entidades débiles) representadas con rectángulos dobles.
- **Óvalos:** Se utilizan para mostrar los atributos de las entidades, es decir, las características que las definen.

- Cuando un ovalo está subrayado es un atributo clave.
  - Los atributos compuestos se unen a través de líneas a óvalos secundarios.
  - Los óvalos dobles representan atributos multivaluados.
- **Rombos:** Indican las relaciones entre las diferentes entidades.
  - **Líneas y flechas:** Se utilizan para conectar las entidades y mostrar cómo interactúan entre sí.
  - **Cardinalidades:** Indican la cantidad de datos que se transfieren de una entidad a otra.
  - **Claves foráneas:** Una clave foránea es un atributo en una entidad que hace referencia a la clave primaria de otra entidad.

## Entidades

Una entidad es cualquier objeto del mundo real o conceptual que puede representarse en una base de datos. Cada entidad tiene un conjunto de atributos que describen sus características. Por ejemplo, en una universidad, las entidades podrían ser Estudiante, Profesor, Curso, etc. En términos técnicos, una entidad suele representarse como una tabla en una base de datos relacional.

## Atributos

Un atributo es una propiedad o característica que describe una entidad. En una base de datos relacional, los atributos se representan como columnas de una tabla. Por ejemplo, si la entidad es "Estudiante", algunos atributos pueden ser "Nombre", "Apellido", "Edad" y "Correo electrónico". Los atributos pueden tener restricciones, como tipos de datos, valores únicos o no nulos, y pueden ser claves primarias o foráneas.

## Relaciones

En el contexto de bases de datos, una relación describe cómo dos o más entidades están conectadas entre sí. Las relaciones permiten modelar la interacción entre conjuntos de datos. Estas pueden ser de distintos tipos:

### Clave primaria

Una clave foránea consiste en una o más columnas dentro de una tabla, cuyos valores deben coincidir con los de la clave primaria de otra tabla. Para que se pueda insertar una fila con un valor específico en la clave foránea, debe existir previamente una fila en la tabla relacionada que tenga ese mismo valor como clave primaria.

### Clave foránea

Una clave foránea es un campo o grupo de campos en una tabla que hace referencia a los valores de la clave primaria de otra tabla. Para insertar un registro con un valor determinado en dicha clave, debe existir previamente un registro en la tabla vinculada que contenga ese mismo valor como clave primaria.

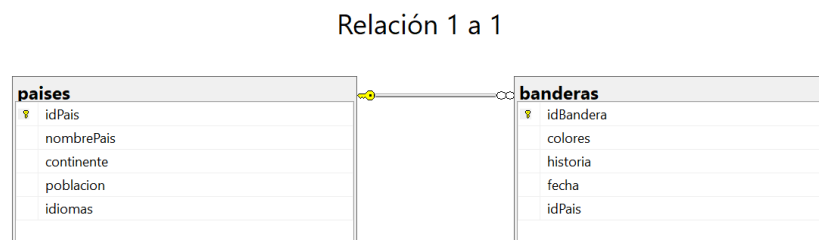
### Tipos de Relaciones

En las bases de datos, existen tres tipos principales de relaciones entre tablas: uno a uno, uno a muchos y muchos a muchos. Para determinar qué tipo de relación corresponde en cada caso, es importante comprender bien los datos, las tablas y las reglas de negocio que las definen.

Al crear tablas en una base de datos, como MySQL o SQL Server, estas relaciones se definen mediante el uso de claves primarias y claves externas.

### Relación uno a uno en bases de datos

**FIGURA 7 – RELACION 1 A 1**



Fuente: <https://sqllearning.com/es/introduccion-sql-server/tipo-relaciones/>

En una relación de este tipo, cada registro de una tabla se asocia con uno y solo un registro en la tabla relacionada. Es decir, cada valor de la clave primaria en una tabla se vincula con un solo valor en la otra tabla, o bien no se vincula a ningún registro.

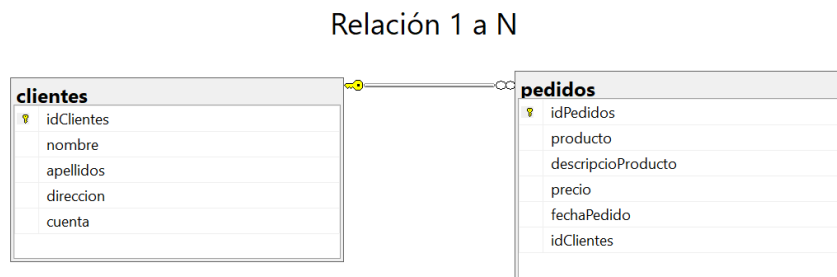


Este tipo de relación generalmente está determinado por reglas de negocio y no suele surgir de manera natural de los datos. Es menos común en la práctica, ya que en muchos casos es más eficiente combinar ambas tablas en una sola.

Ejemplo: Supongamos que tenemos dos tablas. La tabla A contiene los nombres de los clientes, y la tabla B almacena la información de contacto. En este caso, cada entrada de la tabla A corresponde a un único registro en la tabla B.

### Relación uno a muchos en bases de datos

**FIGURA 8 – RELACION 1 A MUCHOS**



Fuente: <https://sqllearning.com/es/introduccion-sql-server/tipo-relaciones/>

En una relación de uno a muchos, un solo registro de la primera tabla puede estar relacionado con múltiples registros en la segunda tabla. Sin embargo, cada registro de la segunda tabla se vincula solo a un registro en la primera tabla. Este tipo de relación es el más común en las bases de datos modernas.

Ejemplo: Una tabla podría contener información básica de los clientes, mientras que otra tabla registra los pedidos de esos clientes. En este caso, un cliente puede tener varios pedidos, pero cada pedido solo está asociado a un cliente específico.

### Relación muchos a muchos en bases de datos

**FIGURA 9 – RELACION MUCHOS A MUCHOS**

## Relación N a N



Fuente: <https://sqllearning.com/es/introduccion-sql-server/tipo-relaciones/>

Las relaciones de muchos a muchos son las más complejas, ya que en ellas, cada registro de la primera tabla puede estar relacionado con varios registros en la segunda tabla, y viceversa. Para gestionar este tipo de relación, se utiliza una tercera tabla, conocida como tabla de unión o tabla de enlace, que conecta las dos tablas principales y facilita el manejo de las relaciones.

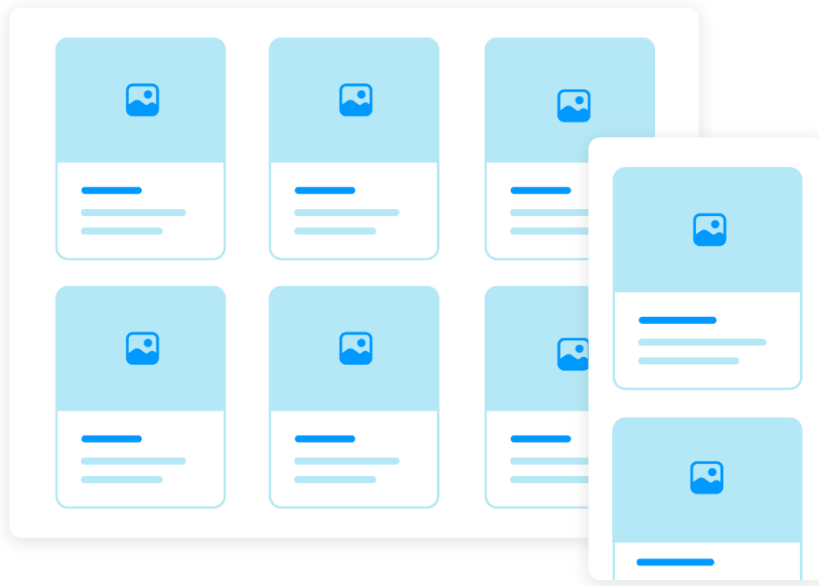
Ejemplo: Supongamos que una tabla almacena los estudiantes y sus nombres, y otra tabla contiene los cursos. Un estudiante puede estar inscrito en varios cursos, y a su vez, cada curso puede tener varios estudiantes inscritos.

### 2.1.3 Patrones de GUI

#### 1. Cards

Las cards son ideales para mostrar visualmente productos, pedidos o clientes, agrupando información clave como nombre, imagen, precio y disponibilidad en un solo contenedor. En el contexto de la Pastelería Vlady, este patrón facilita que los vendedores puedan identificar rápidamente los productos sin necesidad de leer listas extensas, lo que agiliza el proceso de atención en un entorno de alta rotación y pedidos personalizados, mejorando la precisión y reduciendo tiempos de búsqueda.

## FIGURA 10 EJEMPLO DE CARDS

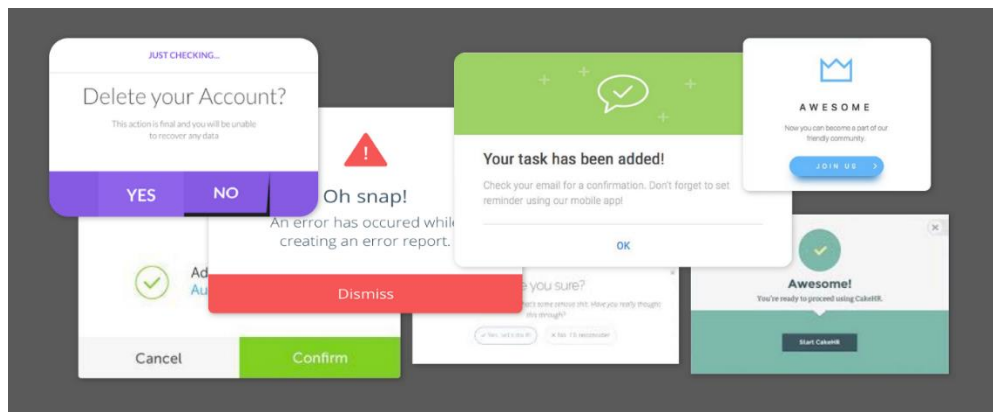


Fuente: <https://www.justinmind.com/es/ui-diseno/tarjetas>

## 2. Modales (Modals)

Los modales ofrecen una forma eficiente de confirmar acciones importantes como el registro de una venta, la edición de un pedido o la eliminación de un cliente, sin salir de la vista principal. En este negocio, se necesita responder rápidamente a los pedidos y mantener la fluidez operativa, los modales nos evitan principalmente interrupciones innecesarias.

**FIGURA 11 EJEMPLO DE MODALS**



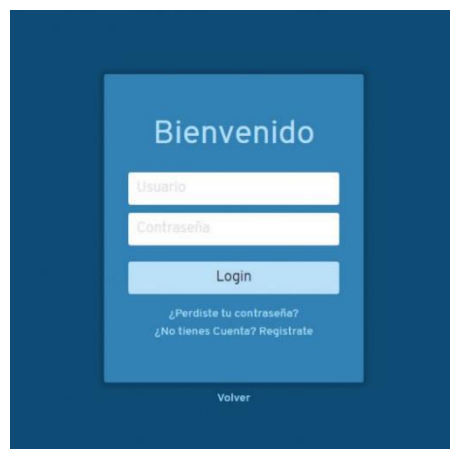
Fuente: <https://medium.com/wikiux/pr%C3%A1cticas-recomendadas-para-modales-superposiciones-ventanas-de-di%C3%A1logo-2b21f9e43f45>

### 3. Breadcrumbs

Los breadcrumbs proporcionan una guía de navegación más sencilla de entender que permite al usuario ubicarse fácilmente dentro del sistema, especialmente útil al consultar reportes, gestionar pedidos o revisar historial de clientes. En nuestro sistema, el administrador puede explorar múltiples niveles de información, este patrón mejora la orientación y la usabilidad, reduciendo la pérdida de contexto y facilitando un acceso eficiente a los datos requeridos.

## FIGURA 12 EJEMPLO DE BREADCRUMBS

Ejemplo de Login



The image shows a login form titled "Bienvenido" (Welcome) on a dark blue background. The form is a light blue rectangle containing the following elements: a "Usuario" (Username) input field, a "Contraseña" (Password) input field, a "Login" button, and two links: "¿Perdiste tu contraseña?" (Forgot your password?) and "¿No tienes Cuenta? Regístrate" (Don't have an account? Register). Below the form, there is a "Volver" (Return) link.

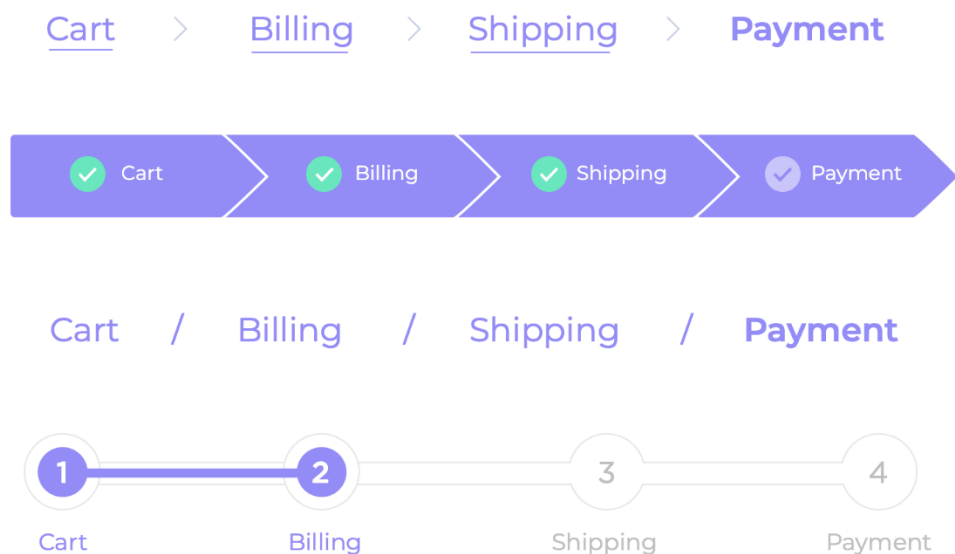
Fuente: <https://www.videojuegosydesarrollo.com/wp-content/uploads/2020/01/login-blue-tones-768x538.jpg>

**FIGURA 13 EJEMPLO DE CALENDARIO**

May 2014						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
April 27 Sleep	28 English class Music class	29 Math class Biology class Physical experiment John's birthday party	30 Computer theory Art class	May 1 English class History class	2 Physics class Math class Sports class	3
4 Play badminton with John	5 English class Music class Yoga class	6 Math class Biology class	7 Computer theory Art class	8 English class History class Mum's birthday	9 Physics class Math class Sports class	10 Short trips
11 Short trips	12 English class Music class	13 Math class Biology class Physical experiment	14 Computer theory Art class	15 English class History class	16 Physics class Math class Sports class	17
18 Play badminton with John	19 English class Music class Yoga class	20 Math class Biology class	21 Computer theory Art class	22 English class History class	23 Physics class Math class Sports class	24 Costume party
25 Sleep	26 English class Music class	27 Math class Biology class Physical experiment	28 Computer theory Art class	29 English class History class	30 Physics class Math class Sports class	31
June 1 Play badminton with John	2 English class Music class Yoga class	3 Math class Biology class	4 Computer theory Art class	5 English class History class	6 Physics class Math class Sports class	7

Fuente: <https://www.edrawsoft.com/template/calendar-plan.png>

**FIGURA 14 EJEMPLO 3 DE BREADSCUMB**



Fuente: <https://ecdisis.com/que-son-las-breadcrumbs/>

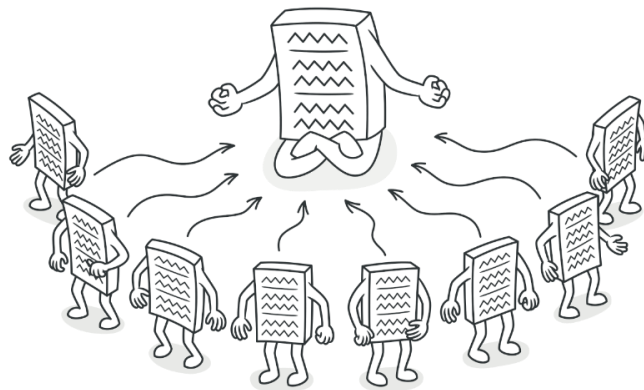
## 2.1.4 Patrones de desarrollo

### Singleton

El patrón Singleton es necesario en el sistema de ventas para la Pastelería Vlady porque permite garantizar una única instancia de conexión a la base de datos, la cual es compartida por todos los módulos del sistema, como el registro de ventas, la consulta de stock, la generación de reportes y el manejo de clientes. Esto evita la creación innecesaria de múltiples conexiones, optimiza el uso de recursos, mejora el rendimiento del sistema y centraliza la configuración de acceso a la base de datos.

Ejemplo:

**FIGURA 15 SINGLETON IMAGEN**

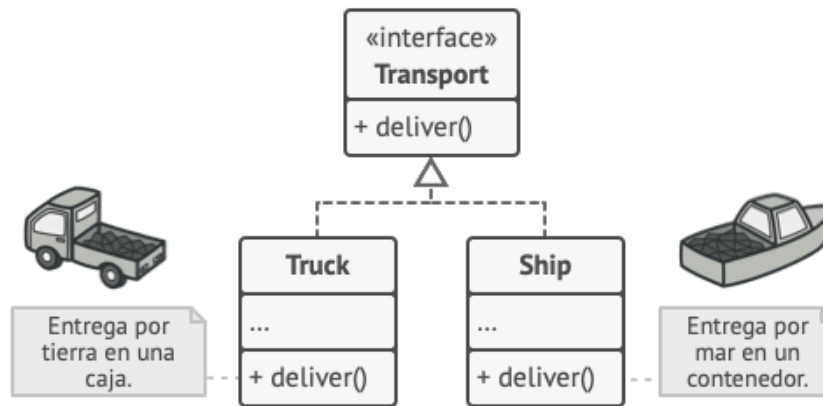


Fuente: <https://refactoring.guru/images/patterns/content/singleton/singleton-2x.png?id=accb2cc7594f7a491ce01dddf0d2f876>

### **Factory Method**

En nuestra aplicación, gestionamos diferentes tipos de empleados, como recepcionistas, asistentes y administradores. Implementar un CRUD separado para cada tipo de empleado sería un proceso tedioso y repetitivo. Para simplificar este trabajo, decidimos implementar el patrón Factory Method. Con este patrón, podemos crear empleados a partir de una interfaz común, permitiendo definir qué tipo de empleado se va a crear sin necesidad de tener un método de creación específico para cada tipo.

**FIGURA 16 FACTORY METHOD**

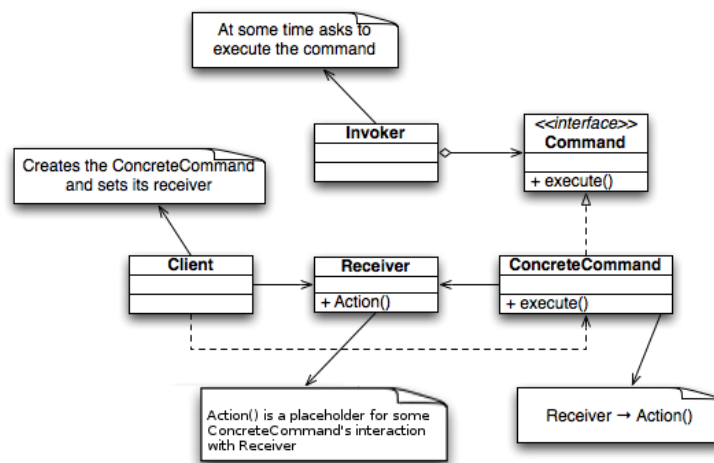


Fuente: <https://refactoring.guru/es/design-patterns/factory-method>

## Command

El patrón Command encapsula una solicitud como un objeto, lo que permite parametrizar acciones, almacenar historial o deshacer operaciones. Se aplica en casos como registrar acciones como "crear venta", "cancelar pedido" o "emitir reporte", que podrían necesitar ser repetidas o revertidas. Su justificación radica en permitir la implementación de historial de acciones, auditoría y la capacidad de deshacer operaciones en el futuro, mejorando el control y la flexibilidad del sistema.

**FIGURA 17 EJEMPLO DE COMMAND**



[https://upload.wikimedia.org/wikipedia/commons/8/8e/Command\\_Design\\_Pattern\\_Class\\_Diagram.png](https://upload.wikimedia.org/wikipedia/commons/8/8e/Command_Design_Pattern_Class_Diagram.png)

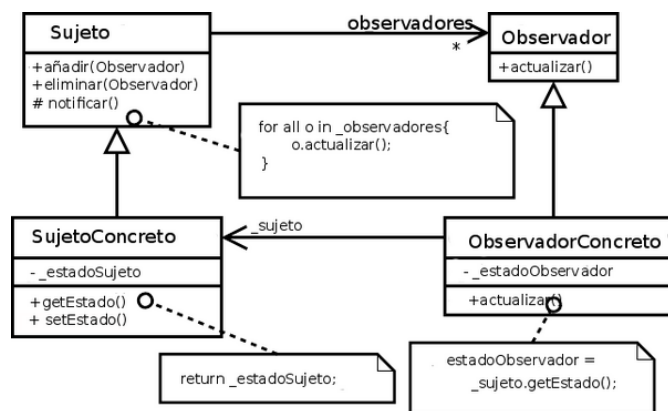
## Observer

El patrón Observer define una dependencia entre objetos de manera que, cuando uno cambia, los demás son notificados automáticamente. Se aplica, por ejemplo, cuando se registra una venta y se actualizan automáticamente el módulo de stock y los reportes de ingresos. Este patrón justifica su uso al mantener la coherencia entre módulos sin acoplarlos directamente, lo que facilita la modularidad y mejora la escalabilidad del sistema.

Patrones de diseño / Design patterns. (s. f.-b). <https://refactoring.guru/es/design-patterns>

Ejemplo:

**FIGURA 18 EJEMPLO DE OBSERVER**



Fuente:

<https://upload.wikimedia.org/wikipedia/commons/thumb/9/97/EstructuraPatronObservador.png/600px-EstructuraPatronObservador.png>

## DAO

El patrón de diseño de objetos de acceso a datos o patrón DAO es un patrón estructural fundamental comúnmente utilizado en aplicaciones Java. El patrón de diseño DAO se utiliza para separar las API o las operaciones de acceso a datos de bajo nivel de las lógicas comerciales de alto nivel. Esto generaba un acoplamiento estrecho y varias desventajas:

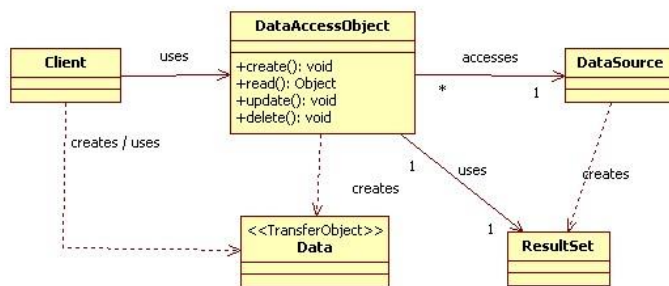


- **Mantenimiento difícil:** Los cambios en la estructura de la base de datos o en los métodos de acceso requieren modificaciones en toda la aplicación, lo que hace que el mantenimiento sea engorroso.
- **Reutilización limitada:** el código de acceso a los datos era específico de una base de datos particular y no podía reutilizarse fácilmente en otras partes de la aplicación.
- **Capacidad de prueba reducida:** probar la lógica empresarial se volvió más complejo debido a las operaciones de acceso a datos entrelazadas.

Fuente: <https://dev.to/ghazanfarofficial01/dao-design-pattern-2g53>

Ejemplo:

**FIGURA 19 EJEMPLO DE DAO**



Fuente: [https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhriRXg9Xqj-PiYZG9lpZ-OEHcRw0lqnapuyCbl8ndFyEOpj6JE-trHO57hvS3fTZiylvbHeEJo05CZAIZKf91klnu6xo43lZG3WdO\\_VyR2NSWY2Ck3oon5cbATXQsSV0PH-ZcdrL2gy2D/s800/DAO-Class-Diagram.jpg](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhriRXg9Xqj-PiYZG9lpZ-OEHcRw0lqnapuyCbl8ndFyEOpj6JE-trHO57hvS3fTZiylvbHeEJo05CZAIZKf91klnu6xo43lZG3WdO_VyR2NSWY2Ck3oon5cbATXQsSV0PH-ZcdrL2gy2D/s800/DAO-Class-Diagram.jpg)

### 2.1.5 Lenguajes de programación / framework

Para desarrollar el sistema de ventas de la pastelería Vlady, se eligieron estos lenguajes de programación, frameworks y herramientas que se adapten mejor a las necesidades del proyecto:

**NetBeans**

NetBeans es un entorno de desarrollo integrado (IDE) gratuito y de código abierto que permite desarrollar aplicaciones web, de escritorio, móviles y corporativas, especialmente con tecnologías como Java y HTML5. NetBeans también permite integrar múltiples lenguajes de programación con los entornos y sistemas operativos correspondientes, facilitando todo el ciclo de desarrollo de software.

#### **Características de NetBeans:**

- Gratuito y de código abierto.
- Multiplataforma.
- Herramientas integradas en compilación y depuración
- Autocompletado inteligente.
- Usable en múltiples plataformas.

Fuente: <https://www.crehana.com/blog/transformacion-digital/que-es-netbeans/>

#### **FIGURA 20 NETBEANS**



Fuente: <https://www.unigermana.edu.co/caja-de-herramientas-especificas/java-netbeans-8-12>

#### **JavaScript**

JavaScript (abreviado como JS) es un lenguaje de programación de alto nivel creado en 1995 por Brendan Eich. Desde entonces, se ha consolidado como uno de los principales lenguajes utilizados en el desarrollo web, desempeñando un papel fundamental en la

construcción de páginas web interactivas y dinámicas. A pesar de su nombre, JavaScript no debe confundirse con Java, ya que son lenguajes distintos tanto en su diseño como en su propósito.

Una de las características más importantes de JavaScript es que se ejecuta directamente en el navegador del usuario, en lugar de requerir una compilación previa en el servidor. Esto permite que las páginas web puedan responder de forma inmediata a las acciones de los usuarios, haciendo posibles tareas como la validación de formularios, la creación de animaciones, la actualización de contenido en tiempo real y otras funcionalidades modernas sin necesidad de recargar la página completa. Gracias a JavaScript, las experiencias en línea son mucho más ágiles, interactivas y personalizadas.

Además, para facilitar aún más el trabajo de los desarrolladores, han surgido los llamados frameworks de JavaScript. Un framework es una estructura o plantilla de código que ofrece herramientas y componentes predefinidos, permitiendo desarrollar aplicaciones de manera más rápida, organizada y eficiente. Algunos de los frameworks más conocidos y utilizados en la actualidad son React, Vue.js, Ember y Singular. Estos frameworks no solo simplifican la escritura del código, sino que también proporcionan patrones sólidos y buenas prácticas para construir aplicaciones web modernas.

Fuente: <https://www.unir.net/revista/ingenieria/que-es-javascript/>

### **Características de JavaScript**

- Simplicidad: Estructura sencilla que lo hace fácil de aprender e implementar.
- Velocidad: Su ejecución es más rápida que otros lenguajes y favorece la detección de errores.
- Versatilidad: Es compatible con lenguajes como PHP, Java, etc.
- Popularidad: Tiene varios foros y recursos que ayudan a quienes empiezan con habilidades y conocimientos limitados.

Fuente: <https://ceeivalencia.emprenemjunts.es/?op=8&n=28660>

### **Node.js**

## **FIGURA 21 NODEJS**



Fuente: <https://agilesparks.com/courses/technical-excellence-courses/nodejs-development/>

Es un entorno de tiempo de ejecución de código abierto y multiplataforma que permite ejecutar código JavaScript fuera de un navegador web. Es decir, Nodejs permite ejecutar JavaScript en el servidor. Fue creado por Ryan Dahl en 2009 y se basa en el motor JavaScript V8 de Google Chrome.

Además, es especialmente adecuado para aplicaciones web en tiempo real y de alto rendimiento, como servidores de aplicaciones, API REST, servicios de chat, juegos en línea, entre otros.

### Ventajas de Node.js

- **Eficiencia y escalabilidad:**  
Utiliza un modelo de entrada/salida no bloqueante y basado en eventos, permitiendo manejar muchas solicitudes al mismo tiempo de manera eficiente.
- **Unificación del lenguaje:**  
Permite escribir tanto el frontend como el backend en JavaScript, facilitando la comunicación entre equipos y mejorando la productividad en el desarrollo.
- **Amplio ecosistema de módulos:**  
Gracias a npm, Node.js ofrece miles de paquetes y módulos que permiten reutilizar código y acelerar el desarrollo de aplicaciones.
- **Comunidad activa:**  
Cuenta con abundante documentación, soporte en línea y una gran variedad de herramientas y frameworks que facilitan el trabajo de los desarrolladores.

Fuente: <https://www.solbyte.com/blog/nodejs/>

## SQL

FIGURA 22 SQL



Fuente: <https://arturonavarro.com/data/potencia-tu-analisis-de-ventas-con-sql-guia-practica-con-ejemplos/>

Es un lenguaje que permite gestionar información de manera inteligente, eficiente y rentable, abarcando desde Recursos Humanos hasta la gestión de clientes.

SQL no solo es fácil de aprender, sino que también permite crear, eliminar y consultar bases de datos rápidamente. Esto lo convierte en una herramienta esencial para las empresas que buscan optimizar su toma de decisiones y mejorar su rendimiento.

### Ventajas de SQL

- **Centralización de datos:** Administra toda la información crítica desde un sistema centralizado, lo que aumenta la seguridad y accesibilidad.
- **Gestión de clientes:** Optimiza la relación con tus clientes y mejora tus campañas de marketing a través de una gestión eficaz de datos.
- **Seguimiento de inventario:** Monitorea el inventario en tiempo real y ajusta tus políticas de compras de manera eficiente

- **Planificación estratégica:** Con un panel intuitivo, visualiza datos clave para planificar el crecimiento de tu empresa de forma segura.

Fuente: <https://blog.epayco.com/5-beneficios-de-sql-para-manejar-bases-de-datos/>

## 2.1.6 Gestor de Base de Datos

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad gracias a que está basado en código abierto. Fue desarrollado originalmente por la empresa MySQL AB, adquirida posteriormente por Sun Microsystems en 2008, y esta a su vez fue comprada por Oracle Corporation en 2010. Oracle ya contaba con su propio motor de almacenamiento, InnoDB, ampliamente utilizado con MySQL. Aunque MySQL se distribuye como software de código abierto, también dispone de una versión comercial gestionada por Oracle, lo que le otorga una doble licencia: una gratuita bajo código abierto y otra de carácter comercial.

Sus principales características son:

1. **Arquitectura** **Cliente-Servidor**  
Funciona mediante una separación entre cliente y servidor para mejorar el rendimiento. Los clientes pueden consultar, modificar y gestionar datos.
2. **Compatibilidad** **con SQL**  
Soporta el lenguaje SQL, lo que facilita la migración desde otros sistemas de bases de datos.
3. **Vistas**  
Desde la versión 5.0, permite crear vistas personalizadas, útiles especialmente en bases de datos grandes.
4. **Procedimientos** **almacenados**  
Permite encapsular lógica de negocio dentro de la base de datos, mejorando la eficiencia.
5. **Desencadenantes** **(Triggers)**  
Automatiza acciones al producirse ciertos eventos, como actualizar o validar datos.
6. **Transacciones**  
Asegura la integridad de los datos mediante la ejecución completa o la cancelación de todas las operaciones involucradas. Protege ante fallas como cortes de energía.

Fuente: <https://openwebinars.net/blog/que-es-mysql/>

Elegimos MySQL ya que da mejor equilibrio entre rendimiento, facilidad de uso, escalabilidad y soporte. Además, el ser compatible con múltiples lenguajes y plataformas facilita la integración con otras partes del sistema, y su respaldo comercial aseguran estabilidad a largo plazo. A diferencia de SQLite, que es demasiado limitado, o PostgreSQL, que puede ser más complejo, MySQL es la solución más práctica y eficiente para nuestro contexto.

### Cuadro comparativo:

Tabla 14 : Comparación SQL's

Característica	MySQL	PostgreSQL	SQLite
Licencia	Código abierto y versión empresarial	Código abierto	100% libre
Rendimiento	Rápido en consultas de lectura y escritura	Ideal para operación de alta complejidad y volumen	Rápido y ligero, pero limitado en afluencia
Facilidad de uso	Fácil de usar	Más complejo en su configuración y administración	Ideal para proyectos pequeños
Escalabilidad y arquitectura	Ideal para aplicaciones web	Más orientado al análisis de datos	No es ideal para entornos multiusuario
Soporte empresarial	Tiene soporte técnico profesional	Soporte empresarial, limitado	Se orienta para apps locales o móviles

Fuente: Elaboración propia

### 2.1.7 Herramientas adicionales:

#### HTML

El HTML (cuyas siglas en inglés significan *HyperText Markup Language*, traducido como *Lenguaje de Marcado de Hipertexto*) es el lenguaje principal utilizado para la creación y visualización de páginas web y aplicaciones en internet. Aunque hoy en día estamos tan acostumbrados a navegar por la red que no siempre pensamos en ello, cada página o aplicación que vemos está construida gracias a este lenguaje.

Este lenguaje funciona principalmente a través del uso de etiquetas, que se escriben entre los signos < y >. Las etiquetas son fundamentales porque describen el contenido de cada sección de una página web: pueden definir títulos, párrafos, imágenes, enlaces, tablas, listas y muchos otros elementos que forman parte de un sitio web.

En esencia, el HTML organiza el contenido de una página web para que los navegadores puedan interpretarlo correctamente y presentarlo de manera entendible para los usuarios. Aunque no se trata de un lenguaje de programación en el sentido clásico (no contiene lógica de programación como condiciones o bucles), el HTML es indispensable en el desarrollo web, ya que establece la base sobre la cual se construyen y diseñan las páginas que vemos en internet.

### **Ventajas de HTML**

- Es el lenguaje web por excelencia.
- Es de fácil uso y entendimiento.
- Es idóneo para aprender gracias a su marco limpio y consistente.
- Es gratuito y se ejecuta en todos los navegadores.
- Es fácil de integrar a lenguajes de backend como PHP y Node.js

Fuente: <https://ondho.com/diccionario-de-marketing/term/html/>

**FIGURA 23 HTML**



Fuente: <https://soltech.net/html-5-what-is-it-and-why-use-it/>

### **CSS**

El CSS (Cascading Style Sheets), en español "Hojas de Estilo en Cascada," es un lenguaje de programación utilizado en el desarrollo web para definir la presentación y el diseño de una página o sitio web. A menudo, se considera como el complemento perfecto para el lenguaje de marcado HTML, ya que permite separar la estructura del contenido de su aspecto visual.



El CSS sirve para definir cómo se verá una página web. En otras palabras, controla aspectos como colores, fuentes, márgenes, tamaños de texto, posicionamiento de elementos y más. Sin CSS, una página web sería una serie de elementos desordenados y poco atractivos. El CSS permite dar vida a la web y crear diseños atractivos y funcionales.

#### Ventajas de CSS

- **Consistencia:** Puedes definir estilos específicos que se apliquen a múltiples páginas, asegurando que todos los elementos tengan una apariencia uniforme.
- **Flexibilidad:** Puede crear diseños flexibles y adaptables a diferentes tamaños de pantalla y dispositivos, lo que es esencial en la era de la web móvil.
- **Eficiencia:** Reduce la duplicación de código y el tamaño de los archivos, lo que resulta en tiempos de carga más rápidos y una mejor eficiencia en la gestión del sitio web.

Fuente: <https://axarnet.es/blog/que-es-css>

#### FIGURA 24 CSS



Fuente: <https://1000marcas.net/css-logo/>

**CANVA**

Canva es una plataforma de diseño gráfico en línea que permite crear fácilmente contenidos visuales como presentaciones, posters, publicaciones para redes sociales, infografías y más, sin necesidad de conocimientos avanzados en diseño.

**Ventajas:**

- **Fácil de usar:** interfaz intuitiva con función de arrastrar y soltar.
- **Plantillas predefinidas:** gran variedad de diseños personalizables.
- **Colaboración en línea:** permite trabajo en equipo en tiempo real.
- **Accesible desde la web:** no requiere instalación.
- **Versión gratuita funcional:** incluye muchas herramientas sin costo.

**FIGURA 25 CANVA**



Fuente: <https://www.pcmag.com/reviews/canva>

**GitHub**

GitHub es una plataforma de desarrollo colaborativo que aloja proyectos en la nube utilizando el sistema de control de versiones llamado Git. Ayuda a los desarrolladores a almacenar y administrar el código llevando un registro de cambios. Generalmente el código es abierto, lo que permite realizar proyectos compartidos y mantener el seguimiento detallado de su progreso. La plataforma GitHub también funciona como red social conectando a los desarrolladores con los usuarios. Como usuario puedes descargar programas o aplicaciones, y de la misma manera puedes aportar a su desarrollo ofreciendo mejoras y discutir las cuestiones que te interesan en foros temáticos.

Fuente: <https://ebac.mx/blog/que-es-github>

**FIGURA 26 GITHUB LOGO**



Fuente: <https://ecoding.dev/p/basico-de-github-575>

**Draw.io**

**FIGURA 27 DRAW.IO**



Fuente: <https://apps.microsoft.com/detail/9mvvszk43qqw?hl=es-ES&gl=ES>

Draw.io es una herramienta de diagramación, de diagrama de flujo, de proceso, entre otras muchas funciones. Es una herramienta gratuita con la que se puede dibujar cualquier tipo de mapas mentales, mapas conceptuales, esquemas o diferentes representaciones gráficas, como diagrama de jerarquía o conjuntos. Principales características:

- **Integración:** Se puede utilizar en versión online o de escritorio.
- **Plantillas y librerías:** La herramienta dispone de una serie de plantillas organizadas por tipos que hacen el trabajo más sencillo y más rápido.

- **Sencillez de uso:** la interfaz de Draw.io es sencilla, cuenta con las herramientas básicas para poder desarrollar la experiencia.
- Compartir y cooperar: Crea archivos que pueden ser editados por varias personas al mismo tiempo.

**Fuente:** [https://intef.es/observatorio\\_tecno/draw-io-mucho-mas-que-mapas-mentales/](https://intef.es/observatorio_tecno/draw-io-mucho-mas-que-mapas-mentales/)

## StarUML

StarUML es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture), que en un principio era un producto comercial y que hace cerca de un año paso de ser un proyecto comercial (anteriormente llamado plastic) a uno de licencia abierta GNU/GPL. Algunas de sus aplicaciones son:

- Diagrama de casos de uso
- Diagrama de clase
- Diagrama de secuencia
- Diagrama de colaboración.

**FIGURA 28 STARUML**



**Fuente:** <https://hl4rny.tistory.com/446>

## Excel

Excel es una potente aplicación de hoja de cálculo desarrollada por Microsoft que permite a los usuarios crear, organizar y analizar datos en formato de tabla, lo que facilita tareas como cálculos matemáticos, compresión de estadísticas, creación de gráficos y gestión de información financiera.

Este programa tiene una interfaz intuitiva basada en celdas, donde los usuarios pueden ingresar datos, así como aplicar fórmulas y funciones para realizar cálculos automáticos. Su versatilidad y capacidad para manejar grandes conjuntos de información la han convertido en una herramienta esencial en campos de finanzas, contabilidad, planificación empresarial y análisis de datos. Estas utilidades, sin duda, contribuyen a su continua relevancia en el mundo empresarial.

Fuente: <https://blog.hubspot.es/marketing/para-que-sirve-excel>

**FIGURA 29 EXCEL LOGO**



Fuente: <https://pixabay.com/da/illustrations/excel-logo-excel-logo-genereret-8033473/>

## **Word**

El software o procesador de textos de Microsoft Windows (el más popular y utilizado del mundo), que básicamente te permite escribir y automatizar textos que se pueden editar, guardar, compartir e imprimir, entre otras maravillosas funciones. Este famoso editor de texto cuenta con una versión de escritorio, una versión web y una aplicación para dispositivos móviles con sistema Android y iOS.

Características:

- Automatizar y escribir textos.
- Crear presentaciones y documentos desde cero.
- Agregar texto, imágenes, gráficos y vídeos.
- Acceder a documentos desde un equipo, tableta o teléfono.
- Compartir documentos y trabajar con otras personas.
- Controlar y revisar cambios.

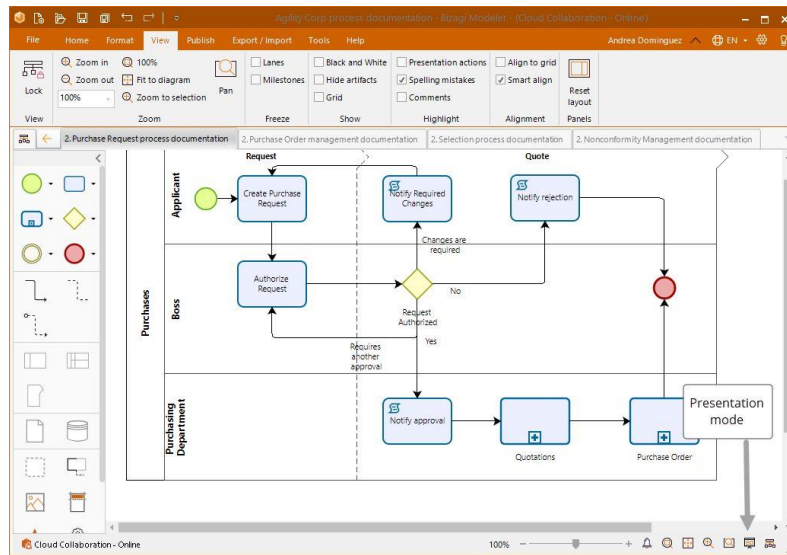
Fuente: <https://www.crehana.com/blog/negocios/word-caracteristicas-y-funciones/>

**FIGURA 29 WORD LOGO**



Fuente: <https://apps.apple.com/es/app/microsoft-word/id586447913>

**Bizagi Modeler**



Fuente: [https://help.bizagi.com/platform/es/index.html?presentation\\_actions.htm](https://help.bizagi.com/platform/es/index.html?presentation_actions.htm)

Bizagi Modeler es un software de modelado de procesos de negocio (BPMN) diseñado para mapear, diagramar y documentar flujos de trabajo de forma visual. Esta herramienta gratuita permite visualizar y optimizar procesos empresariales sin necesidad de programación. Compatible con el estándar BPMN 2.0, Bizagi Modeler ayuda a identificar oportunidades de mejora y a documentar claramente cada paso del proceso.

Está pensado tanto para analistas y gestores de procesos como para usuarios sin experiencia técnica. Gracias a su interfaz intuitiva de arrastrar y soltar, cualquier persona puede elaborar modelos de flujo sin conocimientos de programación. De hecho, es descrito como “un aliado perfecto tanto para expertos en BPM como para principiantes” por su facilidad de uso. Bizagi Modeler se utiliza ampliamente en empresas y organizaciones que necesitan documentar y analizar sus procesos de negocio de manera clara y colaborativa.

### Características:

- **Edición visual BPMN 2.0** – Soporta el estándar BPMN 2.0 al 100%, con notación gráfica para modelar procesos de negocio
- **Interfaz intuitiva (Drag & Drop)** – Permite arrastrar y soltar elementos para construir diagramas sin escribir código.
- **Documentación automática** – Genera informes y documentación profesional de los procesos modelados (Word, PDF, HTML).
- **Colaboración en equipo** – Ofrece funciones para compartir modelos con el equipo (en la nube o local) y sincronizar cambios en línea.

- **Simulación integrada** – Simula el flujo de procesos en tiempo real para predecir su desempeño antes de implementarlos.
- **Plantillas y símbolos predefinidos** – Cuenta con una amplia biblioteca de plantillas, símbolos y guías que facilitan la creación de diagramas precisos.
- **Multi-idioma** – Está disponible en varios idiomas (incluido el español), lo que facilita su adopción por equipos globales.
- **Gratis y sin límites** – La versión básica es completamente gratuita, sin restricciones de uso ni de número de usuarios.

Fuente: <https://www.bizagi.com/es/plataforma/modeler>

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

### 2.2.1 Fórmulas

#### Indicadores de gestión:

Los indicadores de gestión son métricas numéricas utilizadas para cuantificar, evaluar y monitorear el desempeño de una organización, sus procesos o servicios. Estos indicadores transforman datos en información valiosa para una adecuada toma de decisiones, optimización de recursos y mejora continua.

Principales ventajas:

- Permiten monitorear el avance hacia los objetivos establecidos.
- Proporcionan datos concretos para fundamentar la toma de decisiones estratégicas y operativas.
- Ayudan a identificar tendencias, patrones y problemas que requieren atención.
- Posibilitan el seguimiento del impacto de las acciones implementadas.

Entre los indicadores que usaremos para nuestro proyecto están:

#### Venta Promedio por Cliente

**Descripción:** Representa el monto promedio que gasta un cliente haciendo una transacción. Es útil para ver el comportamiento de compra y planificar estrategias que aumenten la ganancia por venta promedio.



**Ventajas:**

- Permite identificar oportunidades para promociones o ventas cruzadas.
- Ayuda a evaluar la fidelidad del cliente y su valor económico.
- Facilita proyecciones de ingresos.

**Formula:**

$$\text{Promedio Diario} = \frac{\text{Ingresos diarios}}{\text{Numero de ventas diarias}}$$

$$\text{Promedio Semanal} = \frac{\text{Ingresos semanales}}{\text{Numero de ventas semanales}}$$

**Stock de Productos**

**Descripción:** Calcula la cantidad de productos disponibles para la venta. Este indicador permite garantizar que hallan productos disponibles y planificar reposiciones.

**Ventajas:**

- Evita faltantes y pérdida de ventas por falta de productos.
- Permite optimizar compras y producción.
- Mejora la satisfacción del cliente al garantizar disponibilidad.

**Formula:**

$$\text{Stock disponible} = \text{Stock Inicial} - \text{Cantidad Vendida}$$

**Tiempo de Atención Promedio**

**Descripción:** Calcula el tiempo promedio que tarda un cliente en ser atendido desde que realiza el pedido hasta que finaliza la compra.

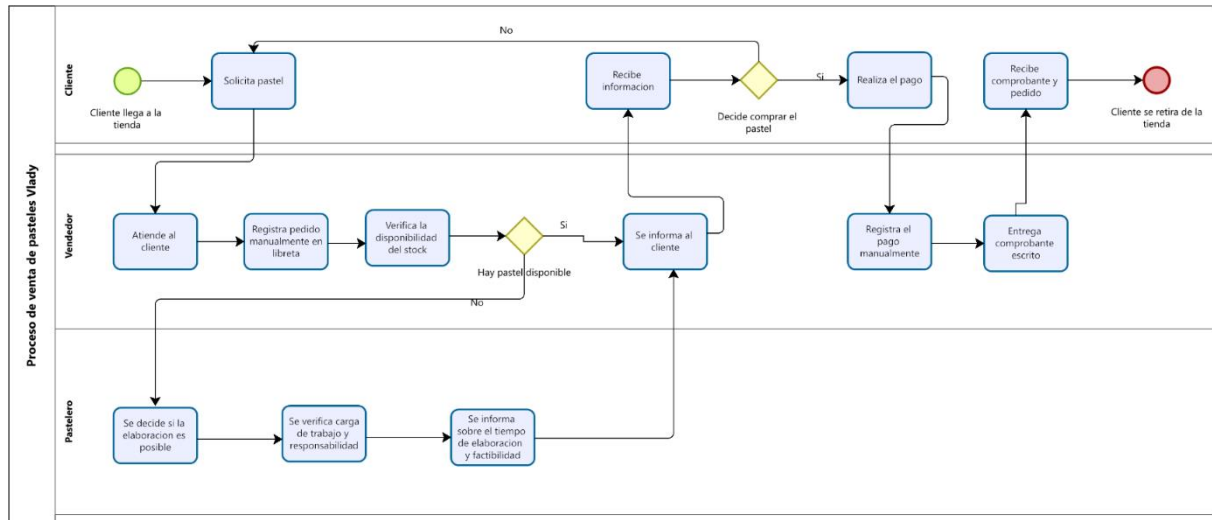
**Ventajas:**

- Evalúa la eficiencia del servicio al cliente.
- Permite identificar cuellos de botella en la atención.
- Mejora la experiencia del cliente al reducir tiempos de espera.

**Formula:**

$$\text{Tiempo promedio diario} = \frac{\Sigma(\text{Tiempo de venta diario por cliente})}{\text{Numero de ventas diarias}}$$

## Ingresos por Ventas



**Descripción:** Muestra el total de ingresos generados por la venta de productos en un periodo determinado. Evalúa la rentabilidad del negocio.

### Ventajas:

- Permite medir el desempeño financiero del negocio.
- Facilita proyecciones de crecimiento.
- Ayuda a evaluar la efectividad de campañas de ventas y marketing.

### Formula:

$$\text{Ingresos diarios} = \Sigma(\text{Precio de cada producto vendido al dia})$$

$$\text{Ingresos semanales} = \Sigma(\text{Precio de cada producto vendido a la semana})$$

## CAPÍTULO 3 - DESARROLLO DE LA SOLUCIÓN

### 3.1 Modelo de proceso de negocios

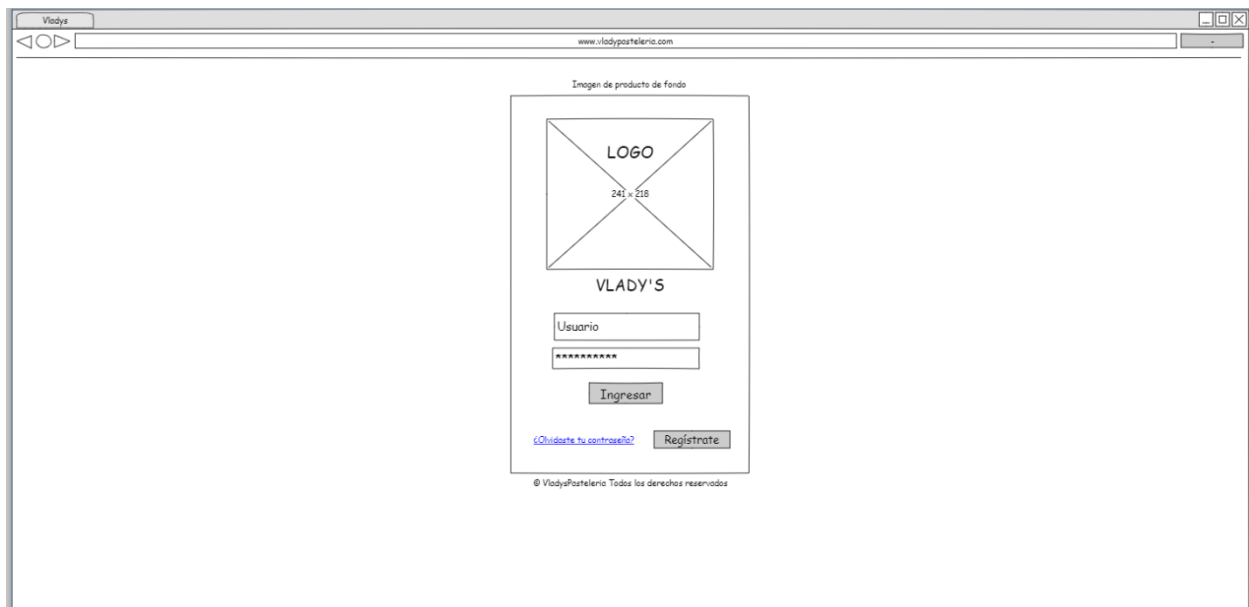
**FIGURA 30 DIAGRAMA AS IS**

Fuente: Elaboración propia

## 3.2 Prototipos

A continuación, se muestra el prototipo inicial de la interfaz de usuario, elaborado con la herramienta Pencil. Este diseño sirvió como punto de partida para definir la estructura visual y funcional del sistema, permitiendo representar de forma preliminar los principales elementos de navegación, formularios y distribución de la información. A medida que se avanzó en el desarrollo del proyecto y se refinaron los requerimientos, este prototipo fue modificado y ajustado para mejorar la experiencia del usuario y adaptarse mejor a las necesidades identificadas. El prototipo cumple un rol clave como guía para el diseño final de la interfaz.

**FIGURA 31 PROTOTIPO LOGUEO**



Fuente: Elaboración propia

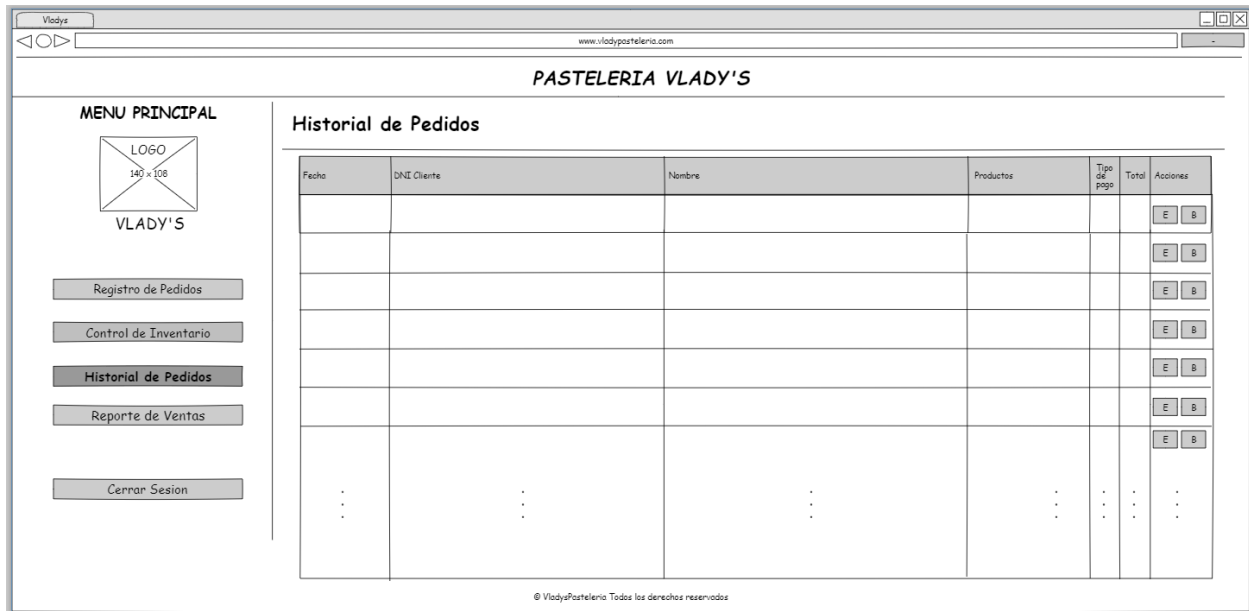
**FIGURA 32 PROTOTIPO VENTAS**

Fuente: Elaboración propia

**FIGURA 33 PROTOTIPO INVENTARIO**

Fuente: Elaboración propia

**FIGURA 34 PROTOTIPO HISTORIAL**



Fuente: Elaboración propia

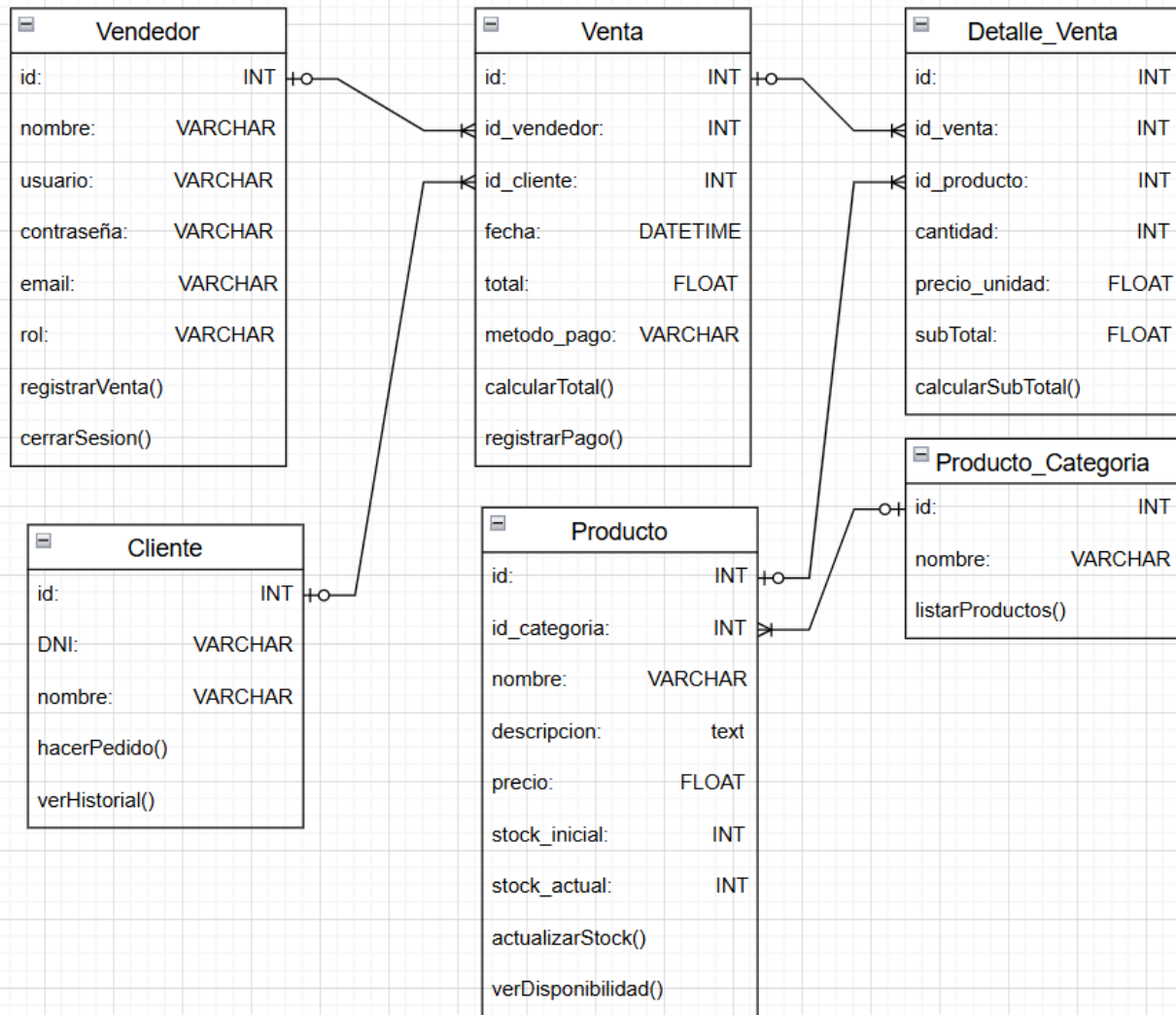
**FIGURA 35 PROTOTIPO REPORTE VENTA**



Fuente: Elaboración propia

### 3.3 Diagrama de clases:

**FIGURA 36 DIAGRAMA DE CLASES**



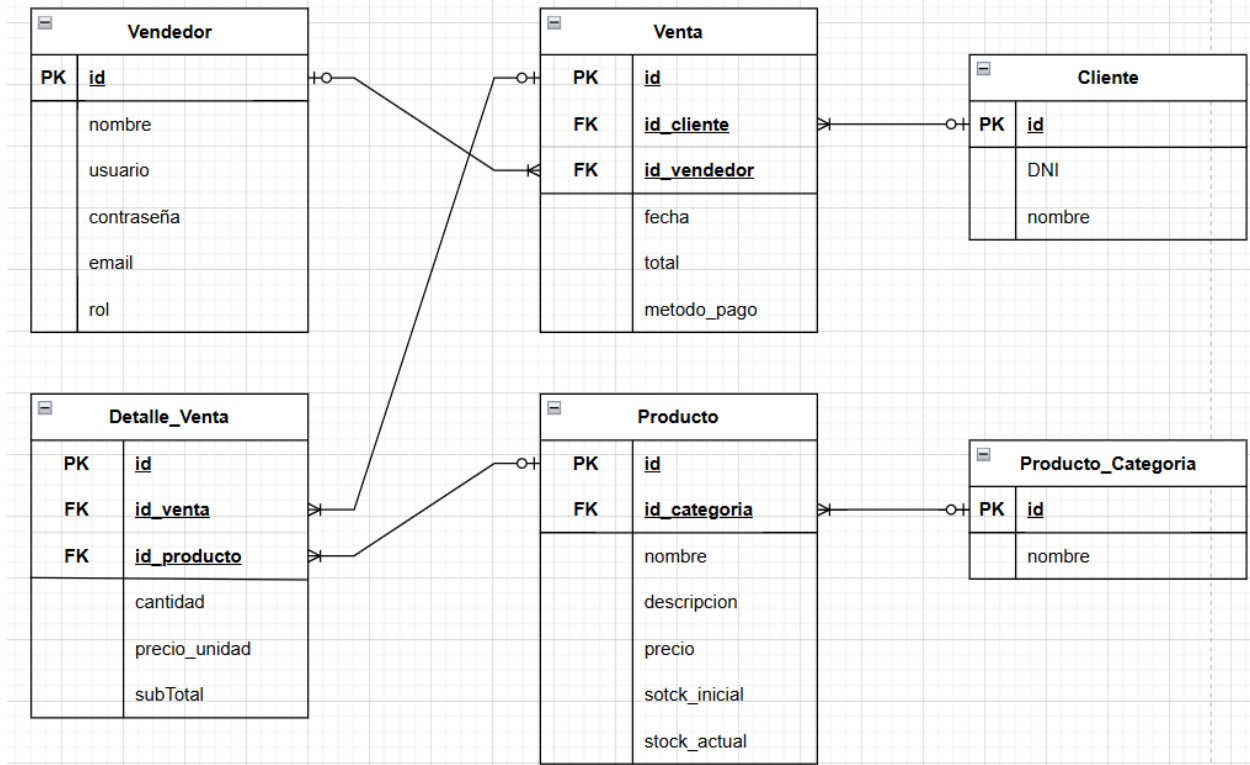
Fuente: Elaboración propia

### 3.4 Modelamiento de base de datos:

Diagrama Entidad-Relación:

A continuación, se presenta el Modelo Entidad-Relación (MER) que representa la estructura lógica del sistema propuesto. Este modelo nos permite identificar de manera clara las entidades principales, sus atributos y las relaciones entre ellas, facilitando así el diseño y la implementación de la base de datos. Este esquema servirá como base para el desarrollo de un sistema eficiente, coherente y alineado con los objetivos del proyecto.

**FIGURA 37 DIAGRAMA ENTIDAD RELACION**



Fuente: Elaboración propia

### 3.4.1 Diccionario de datos

Tabla 14: Cliente

	Columna	Tipo de Dato	Descripción	Nn
PK	id	INT(36)	Identificador del cliente	X
	DNI	VARCHAR(8)	Nro. de documento del cliente	X
	nombre	VARCHAR(50)	Nombre del cliente	X

Fuente: Elaboración propia

Tabla 15: Vendedor

	Columna	Tipo de Dato	Descripción	Nn
PK	Id	INT(36)	Identificador del vendedor	X
	nombre	VARCHAR(40)	Nombre del vendedor	X
	usuario	VARCHAR(30)	Nombre del vendedor para el login	X
	contraseña	VARCHAR(18)	Contraseña del vendedor	X
	email	VARCHAR(20)	Correo electrónico del vendedor	X

	rol	VARCHAR(20)	Rol asignado (admin, cajero, etc).	X
--	-----	-------------	------------------------------------	---

Fuente: Elaboración propia

Tabla 16: Venta

	Columna	Tipo de Dato	Descripción	Nn
PK	id	INT(30)	Identificador de la venta	X
FK	id_cliente	INT(30)	Identificador del cliente	X
FK	id_vendedor	INT(30)	Identificador del vendedor	X
	fecha	DATETIME	Fecha y hora de la venta	X
	total	Decimal(10,2)	Monto total de la venta	X
	metodo_pago	VARCHAR(20)	Metodo de pago	X

Fuente: Elaboración propia

Tabla 17: Detalle\_Venta

	Columna	Tipo de Dato	Descripción	Nn
PK	id	INT(30)	Identificador del detalle de venta	X
FK	id_venta	INT(30)	Identificador de la venta	X
FK	id_producto	INT(30)	Identificador del producto vendido	X
	cantidad	INT(30)	Numero de unidades vendidas	X
	precio_unidad	Decimal(10,2)	Precio por unidad	X
	subTotal	Decimal(10,2)	Subtotal (cantidad * precio_unidad)	X

Fuente: Elaboración propia

Tabla 18: Producto

	Columna	Tipo de Dato	Descripción	Nn
PK	id	INT(30)	Identificador del producto	X
FK	id_categoria	INT(30)	Categoría del producto	X
FK	nombre	VARCHAR(40)	Nombre del producto	X
	descripcion	TEXT	Descripción del producto	
	precio	Decimal(10,2)	Precio del producto	X
	stock_inicial	INT(20)	Stock inicial del producto	X
	stock_actua	INT(20)	Stock actual del producto	X

Fuente: Elaboración propia

Tabla 19: Producto\_Categoria

	Columna	Tipo de Dato	Descripción	Nn
PK	id	INT(30)	Identificador de la categoría	X
	nombre	VARCHAR(40)	Nombre de la categoría del producto	X

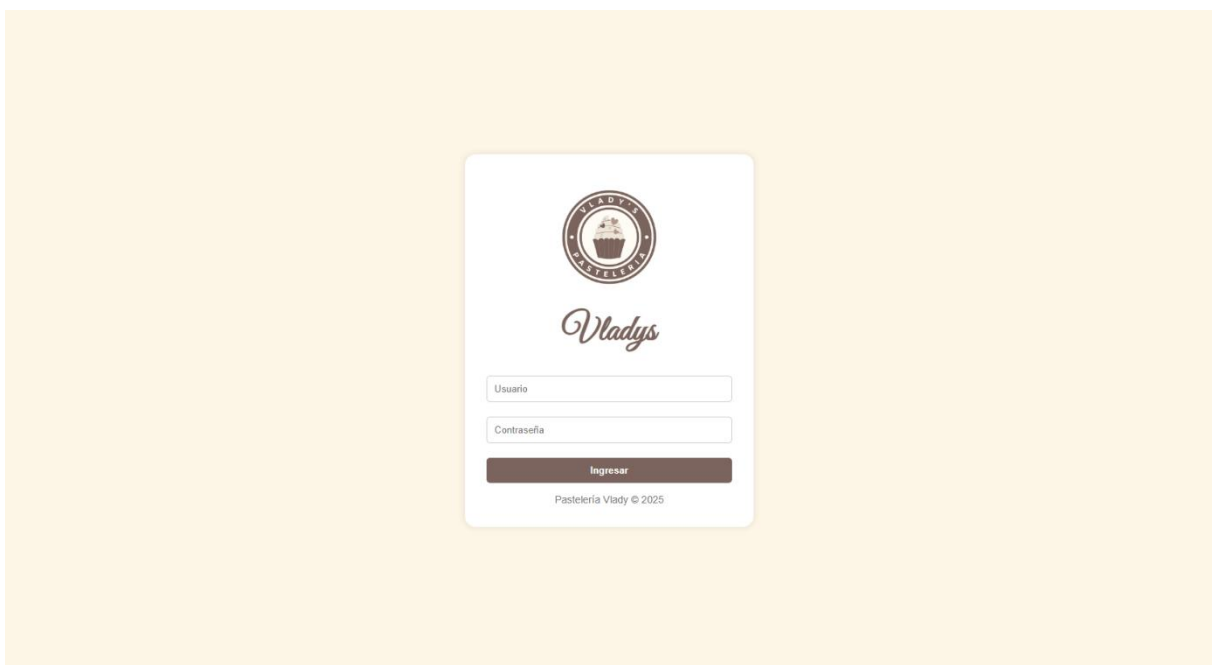
Fuente: Elaboración propia



### 3.5 Entorno visual de los formularios

A continuación, se presenta el entorno visual de los formularios diseñados para el sistema, los cuales representan cómo se verán y funcionarán las interfaces que permitirán al usuario interactuar con la base de datos. Estos formularios han sido diseñados con un enfoque en la usabilidad y la organización clara de la información, permitiendo realizar acciones como el registro, consulta, modificación y eliminación de datos de manera intuitiva. El diseño visual refleja la estructura lógica definida en el modelo entidad-relación, y fue perfeccionado tras las evaluaciones del prototipo inicial, incorporando mejoras en la disposición de campos, botones y validaciones necesarias para garantizar una experiencia de usuario eficiente y amigable.

**FIGURA 38 ENTORNO VISUAL – LOGUEO**



Fuente: Elaboración propia

**FIGURA 39 ENTORNO VISUAL – INICIO**



Fuente: Elaboración propia

**FIGURA 40 ENTORNO VISUAL – VENTA**

**Pastelería Vlady - Sistema de Gestión**

**Menú Principal**

- Inicio
- Generar Venta
- Reporte de Ventas
- Control de Inventario
- Historial de Pedidos
- Cerrar Sesión

**Registro de Pedidos**

Fecha de Venta:

DNI del Cliente:

Nombre del Cliente:

Método de Pago:

Chocolate

Red Velvet

Vainilla/MB

Tres Leches

Moka

Oreo

Total: S/ 0.00

Fuente: Elaboración propia

**FIGURA 41 ENTORNO VISUAL – REPORTE DE VENTA**



Fuente: Elaboración propia

FIGURA 42 ENTORNO VISUAL – INVENTARIO



Fuente: Elaboración propia

FIGURA 43 ENTORNO VISUAL – HISTORIAL

Pastelería Vlady - Sistema de Gestión						
Historial de Pedidos						
						<a href="#">+ Agregar Pedido</a>
Fecha	DNI Cliente	Nombre Cliente	Productos	Método de Pago	Total (\$/)	Acciones
2025-05-14	45227	Juan Pérez	Flan de vainilla (5), Pastel de chocolate (7), Brownie (5), Pie de limón (4)	Tarjeta	S/ 199.00	<a href="#">Editar</a> <a href="#">Eliminar</a>
2025-05-15	12345	Maria López	Red Velvet (2), Cheesecake de Fresa (1)	Efectivo	S/ 152.00	<a href="#">Editar</a> <a href="#">Eliminar</a>

Fuente: Elaboración propia

## Referencias Bibliográficas:

- Latorre, A., & Castañeda, J. (2016). *Desarrollo e implementación de sistema de administración de ventas para Humane Escuela de Negocios*. Universidad Politécnica Salesiana, Ecuador.
- González, C., & Tijerino, J. (2020). *Implementación de un sistema automatizado para el control de ventas en la Panadería Tijerino*. Universidad Nacional Autónoma de Nicaragua.
- García, L. A. (2014). *Diseño y aplicación de sistema para punto de venta*. Instituto Politécnico Nacional, México.
- Sandoval, H., & del Pilar, L. (2017). *Diseño de un sistema de Gestión por procesos para mejorar la productividad y competitividad de la panadería LULI*.
- IBM *InfoSphere Master Data Management*. (n.d.). <https://www.ibm.com/docs/es/imdm/11.6.0?topic=concepts-key-entity-attribute-entity-type>
- IBM *InfoSphere Data Architect*. (n.d.). <https://www.ibm.com/docs/es/ida/9.1.2?topic=entities-primary-foreign-keys>
- Zivkovic, M. (n.d.). *The Beginner's Guide to database Relationships* | Luzmo. <https://www.luzmo.com/blog/database-relationships#:~:text=There%20are%20three%20main%20types,the%20business%20rules%20behind%20them.>

- Web Design Cusco. SCRUM: Una metodología ágil para la gestión de proyectos tecnológicos. <https://webdesigncusco.com/que-es-scrum-y-para-que-sirve/>
- APD. Cómo aplicar la metodología Scrum y qué es el método Scrum. <https://www.apd.es/metodologia-scrum-que-es/>
- Wikipedia. Modelo entidad-relación. [https://es.wikipedia.org/wiki/Modelo\\_entidad-relaci%C3%B3n](https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n)
- Sqliearning. Relaciones de base de datos SQL. <https://sqliearning.com/es/introduccion-sql-server/tipo-relaciones/>
- Ecdisis. ¿Qué son las Breadcrumbs?. <https://ecdisis.com/que-son-las-breadcrumbs/>
- Refactoring Guru. Factory Method. <https://refactoring.guru/es/design-patterns/factory-method>
- Dev.to. DAO Design Patern. <https://dev.to/ghazanfarofficial01/dao-design-pattern-2g53>
- Crehana. ¿Qué es Netbeans? ¡Crea aplicaciones con Java a la velocidad de la luz!. <https://www.crehana.com/blog/transformacion-digital/que-es-netbeans/>
- Unir. JavaScript: ¿qué es y para qué sirve?. <https://www.unir.net/revista/ingenieria/que-es-javascript/>
- Agile Sparks. NodeJS Development. <https://agilesparks.com/courses/technical-excellence-courses/nodejs-development/>
- Solbyte. Node JS: ventajas e inconvenientes. <https://www.solbyte.com/blog/nodejs/>
- ePayco. SQL: La Clave para una Gestión Eficiente de Datos en tu Empresa. <https://blog.epayco.com/5-beneficios-de-sql-para-manejar-bases-de-datos/>
- Open Webinars. Qué es MySQL: Características y ventajas. <https://openwebinars.net/blog/que-es-mysql/>
- Ondho. HTML. <https://ondho.com/diccionario-de-marketing/term/html/>
- SOLTECH. HTML 5: What is it and Why use it?. <https://soltech.net/html-5-what-is-it-and-why-use-it/>
- Axarnet. Qué es el CSS y para qué sirve 【 Guía completa 】 . <https://axarnet.es/blog/que-es-css>
- European Business School. ¿Qué es un modelo CANVAS y para qué sirve? <https://www.eude.pe/blog/2021/02/23/que-es-un-modelo-canvas-y-para-que-sirve/>
- EBAC. Qué es Github y para qué sirve: una guía para principiantes. <https://ebac.mx/blog/que-es-github>
- INTEF. Draw.io: mucho más que mapas mentales. [https://intef.es/observatorio\\_tecno/draw-io-mucho-mas-que-mapas-mentales/](https://intef.es/observatorio_tecno/draw-io-mucho-mas-que-mapas-mentales/)
- HubSpot. Para qué sirve Excel, principales características y cómo funciona. <https://blog.hubspot.es/marketing/para-que-sirve-excel>
- Crehana. Funciones y características de Word, el procesador de texto más usado en el mundo. <https://www.crehana.com/blog/negocios/word-caracteristicas-y-funciones/>
- Bizagi. Bizagi Modeler. <https://www.bizagi.com/es/plataforma/modeler>

