



GA8-220501096-AA1-EV01 DESARROLLAR SOFTWARE A PARTIR DE LA
INTEGRACIÓN DE SUS MÓDULOS COMPONENTES

ANGÉLICA CAÑÓN CALDERÓN
APRENDIS- FICHA 2977518

CENTRO AMBIENTAL Y ECOTURÍSTICO DEL NORORIENTE AMAZÓNICO
SENA REGIONAL GUAINÍA
TECNÓLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE – ADSO
INSTRUCTORA: YINNA PAOLA VARON MOREIRA

NORORIENTE AMAZÓNICO
22 DE NOVIEMBRE DE 2025

INTRODUCCIÓN

La presente evidencia de desempeño tiene como finalidad demostrar el proceso de desarrollo e integración de los módulos componentes de una aplicación web denominada DigiCompra, cumpliendo con los requerimientos funcionales y no funcionales definidos para el sistema.

El proyecto DigiCompra corresponde a una aplicación web orientada a la gestión de compras, en la cual se integran los módulos de frontend, backend, base de datos y seguridad, aplicando buenas prácticas de programación, uso de frameworks, control de versiones y arquitectura por capas.

Durante el desarrollo se aplicaron los conocimientos adquiridos en el componente formativo, incluyendo análisis de requerimientos, diagramas UML, codificación modular, pruebas unitarias y configuración de ambientes de desarrollo y prueba.

Alcance

El proyecto **DigiCompra** consiste en el desarrollo de una aplicación web orientada a la gestión de compras, la cual integra sus módulos componentes bajo una arquitectura cliente-servidor. El sistema permite a los usuarios registrarse, iniciar sesión y visualizar un catálogo de productos, mientras que el administrador puede gestionar la información de los productos mediante operaciones CRUD (crear, consultar, actualizar y eliminar).

La aplicación integra un frontend desarrollado para la interacción con el usuario, un backend encargado de la lógica del negocio y una base de datos relacional para la persistencia de la información. Asimismo, se implementan mecanismos básicos de seguridad para el manejo de usuarios y la protección de datos.

El alcance del proyecto se limita a un entorno web y a un ambiente de desarrollo y pruebas, sin incluir pasarelas de pago en línea, notificaciones externas ni aplicaciones móviles.

Objetivos

Objetivo General

Desarrollar una aplicación web funcional mediante la integración de sus módulos componentes, aplicando buenas prácticas de programación, una arquitectura por capas y control de versiones, de acuerdo con los requerimientos del sistema DigiCompra.

Objetivos Específicos

1. Analizar y aplicar los requerimientos funcionales y no funcionales del sistema DigiCompra.
2. Diseñar la estructura del sistema utilizando diagramas UML como apoyo al desarrollo del software.
3. Codificar los módulos del frontend y backend utilizando el lenguaje y frameworks seleccionados.
4. Implementar una API REST para la comunicación entre la interfaz de usuario y la base de datos.
5. Configurar el servidor y la base de datos para el correcto funcionamiento del sistema.
6. Aplicar mecanismos básicos de seguridad para la gestión de usuarios y la protección de la información.
7. Gestionar el proyecto mediante un repositorio de control de versiones, documentando los cambios realizados.

Requerimientos de Software Funcionales

FUNCIONALIDAD	TIPO (esencial, ideal, opcional)
RF01 – Inicio de sesión	Esencial
RF02 – Gestionar artículos	Esencial
RF02.1 Registrar producto	Esencial
RF02.2 Consultar producto	Esencial
RF02.3 Actualizar producto	Esencial
RF02.4 Inactivar / borrar producto	Esencial
RF03 – Gestionar Trabajadores	Ideal
RF03.1 Agregar trabajador	Ideal
RF03.2 Inactivar/Eliminar trabajador	Opcional
RF04 – Gestionar Clientes	Esencial
RF04.1 Registrar Cliente	Esencial
RF04.2 Consultar Cliente	Esencial
RF04.3 Actualizar Cliente	Esencial
RF04.4 Inactivar / Borrar Cliente	Opcional
RF05 – Gestionar pedidos	Esencial
RF05.1 Hacer pedido	Esencial
RF05.2 Consultar pedido	Esencial
RF05.3 Actualizar pedidos	Ideal
RF06 – Buscar productos	Esencial
RF07 – Visualizar productos	Esencial
RF08 – Carrito de compras (hacer el pedido)	Esencial
RF08.1– Eliminar productos	Esencial
RF08.2 – Aumentar cantidad	Esencial
RF09 – Hacer pagos	Ideal
RF010 – Monitorear el pedido (cliente)	Ideal
RF011 – Ayuda-Guía de uso de la plataforma	Ideal
RF012 – Notificaciones	Esencial
RF012.1 – SMS	Ideal
RF012.2 – Correo	Esencial
RF012.4 – En plataforma	Esencial.
RF013 – Atención al usuario	Esencial
RF013.1-Enciar mensaje	Esencial
RF013.2 Realizar llamada	Esencial
RF014- Gestionar Servicio	Esencial
RF014.1- Consultar Servicio	Esencial
RF014.2- Solicitar Servicio	Esencial

Requisitos no funcionales

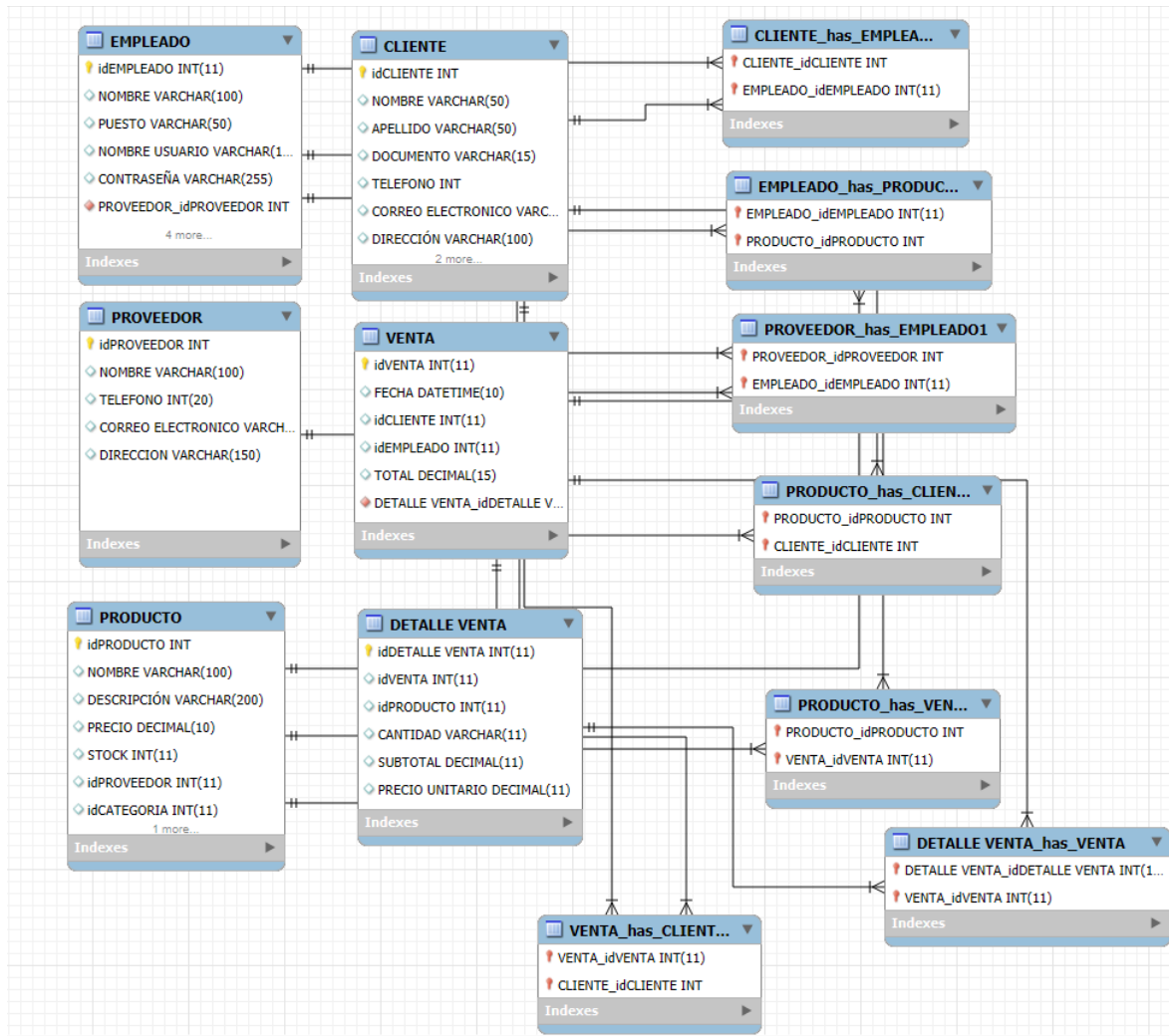
Usabilidad	
RNF01 – Información del producto	Toda la información sobre medicamentos (dosis, efectos secundarios, precauciones) debe presentarse de manera clara, concisa y fácil de entender para el usuario.
RNF02 - Búsqueda	El proceso de búsqueda y filtrado de productos debe ser sencillo y eficiente.
RNF03 – Navegación intuitiva	La estructura de la página debe ser lógica y fácil de seguir, permitiendo a los usuarios encontrar rápidamente los productos que buscan.
RNF04 - Glosario	Los términos médicos deben explicarse de forma sencilla o incluirse un glosario.
RNF05 - Pago	La información sobre precios, formas de pago y envío debe ser transparente y fácil de encontrar.
RNF06 - Información actualizada	Toda la información sobre los productos farmacéuticos debe ser precisa y estar actualizada.
RNF07 - Atención al usuario	Se debe ofrecer la posibilidad de consultar a un farmacéutico en línea para resolver dudas.
Confiabilidad	
RNF08 -Disponibilidad de la Pagina	La aplicación debe tener una disponibilidad del 95%.
RNF09 - Tolerancia a Fallos	Si se produce algún fallo no controlado en la página el usuario podrá seguir navegando en otras áreas y reportar el incidente en el apartado de contáctenos.
RNF10 - Almacenamiento local	El portal Web podrá guardar algunos datos temporales de navegación por el sitio Web.

RNF11 - Almacenamiento remoto	El portal Web almacena datos temporales mediante sesiones, además guardará datos en la base de datos
Seguridad	
RNF12 – Datos encriptados	La información personal y médica de los usuarios debe estar protegida mediante encriptación y otras medidas de seguridad.
RNF13 – Privacidad de datos	Se debe garantizar la privacidad de los datos de los usuarios, cumpliendo con las normativas de protección de datos.
RNF14 - Compra segura	El proceso de compra debe ser seguro y confiable, utilizando protocolos de pago seguros.
RNF15 - Permisos de administrador y usuario	Implementar un sistema de permisos que restrinja el acceso a los datos según el rol del usuario (administrador, cliente).
RNF16 - Detección de anomalías	Implementar sistemas de detección de anomalías para identificar comportamientos inusuales que puedan indicar un ataque.
RNF17 - Escaneo frecuente	Realizar escaneos de vulnerabilidades de forma regular para identificar y corregir cualquier debilidad en la aplicación y la infraestructura.
RNF18 - Firewall	Implementar un firewall para proteger la red de la farmacia de ataques externos.
RNF19 - Contraseñas	Se deben exigir contraseñas que combinen mayúsculas, minúsculas, números y caracteres especiales.
Eficiencia	
RNF20- Tiempo de respuesta	El tiempo de respuesta de la aplicación para el acceso a las funcionalidades básicas no debe ser mayor a 2 segundos
RNF21– Tiempo de respuesta	El tiempo de respuesta del sistema para el proceso de consulta a la base de datos remota no debe ser mayor a 5 segundos
RNF22 – Capacidad de la pagina	La aplicación debe soportar al menos 10.000 usuarios concurrentes sin degradación del servicio.
RNF23 - Optimización multimedia	Se deben optimizar las imágenes y otros elementos multimedia para reducir el tiempo de carga.

RNF24 - Monitoreo de la pagina	Implementar herramientas de monitoreo para supervisar el rendimiento de la aplicación en tiempo real y detectar cualquier falla en la página.
Portabilidad.	
RNF25 – Soporte para navegadores	El portal Web deberá verse correctamente en los navegadores más usados por los usuarios como son Edge, Google Chrome, Mozilla, Safari, y Opera
RNF26 – Multi dispositivo	Adaptar el diseño de la página a diferentes tamaños de pantalla (ordenadores, tablets, Smartphone) para garantizar una visualización óptima.
RNF27 - Funcionamiento en Sistemas Operativos	La página Deberá funcionar eficazmente en los sistemas operativos más comunes (Windows, macOS, Linux, iOS, Android).
Mantenibilidad	
RNF28 – Nombres limpios	Utilizar nombres claros y descriptivos para variables, funciones y clases.
RNF29 - Comentarios	Agregar comentarios explicativos para documentar el código y facilitar su comprensión.
RNF30 - Modularidad	Dividir el código en módulos o componentes reutilizables.
RNF31 - Cohesión y acoplamiento	Cada módulo debe tener una responsabilidad única y depender lo menos posible de otros módulos.
Soportabilidad y operatividad	
RNF32 – Actualizaciones	El sistema será fácil de desplegar en un servidor web que tenga las especificaciones que se den como el lenguaje de programación, base de datos, capacidad de disco.
RNF33 - Documentación	El código fuente debe estar bien documentado y seguir estándares de programación.

RNF34 - Requisitos	Documentación clara sobre los requisitos del sistema.
--------------------	---

Diagrama de entidad – relación



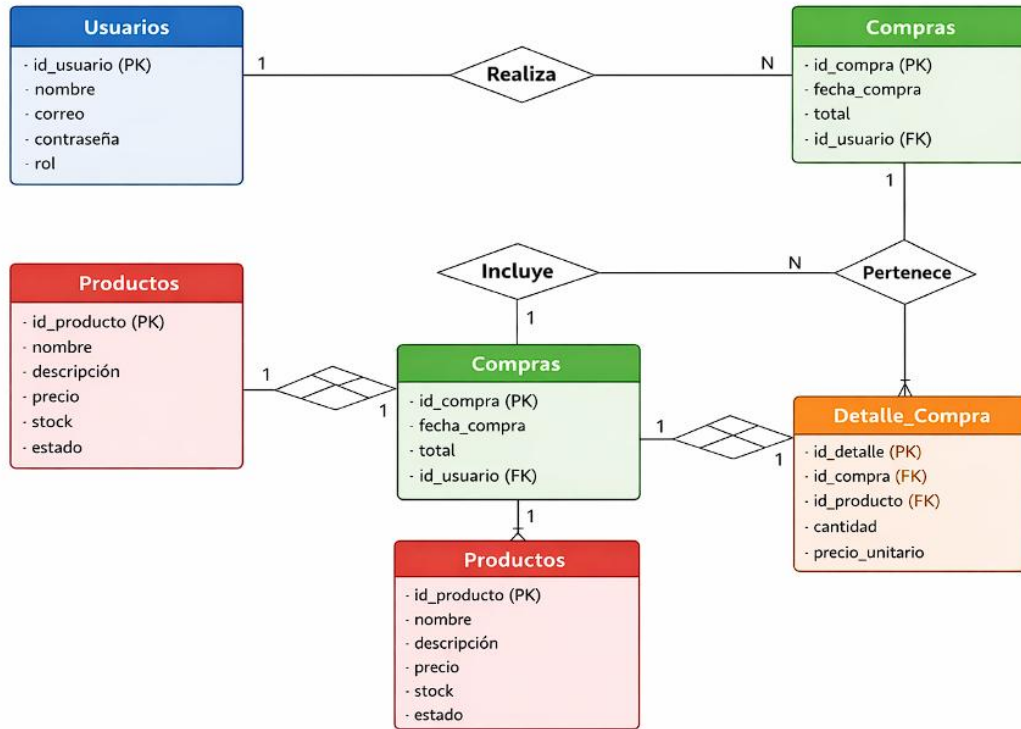


Diagrama de Componentes:

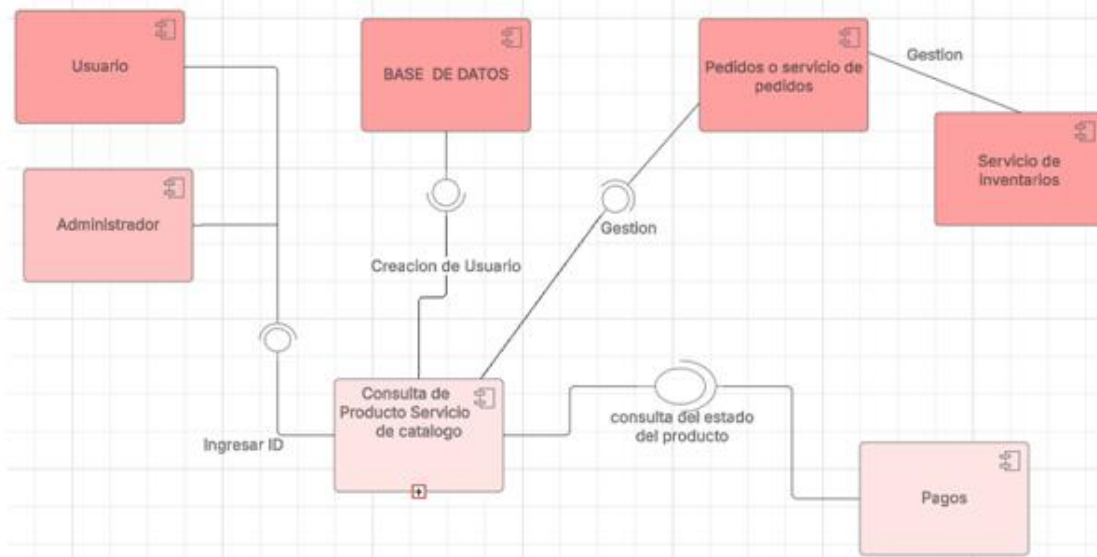


Diagrama de Componentes de DigiCompra

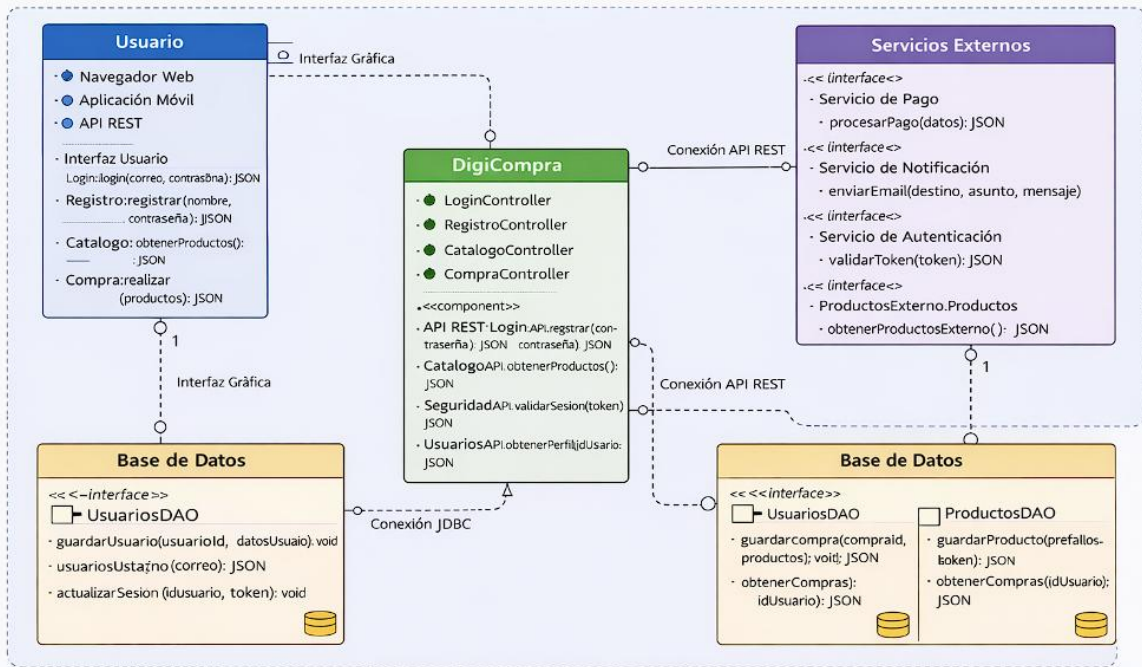
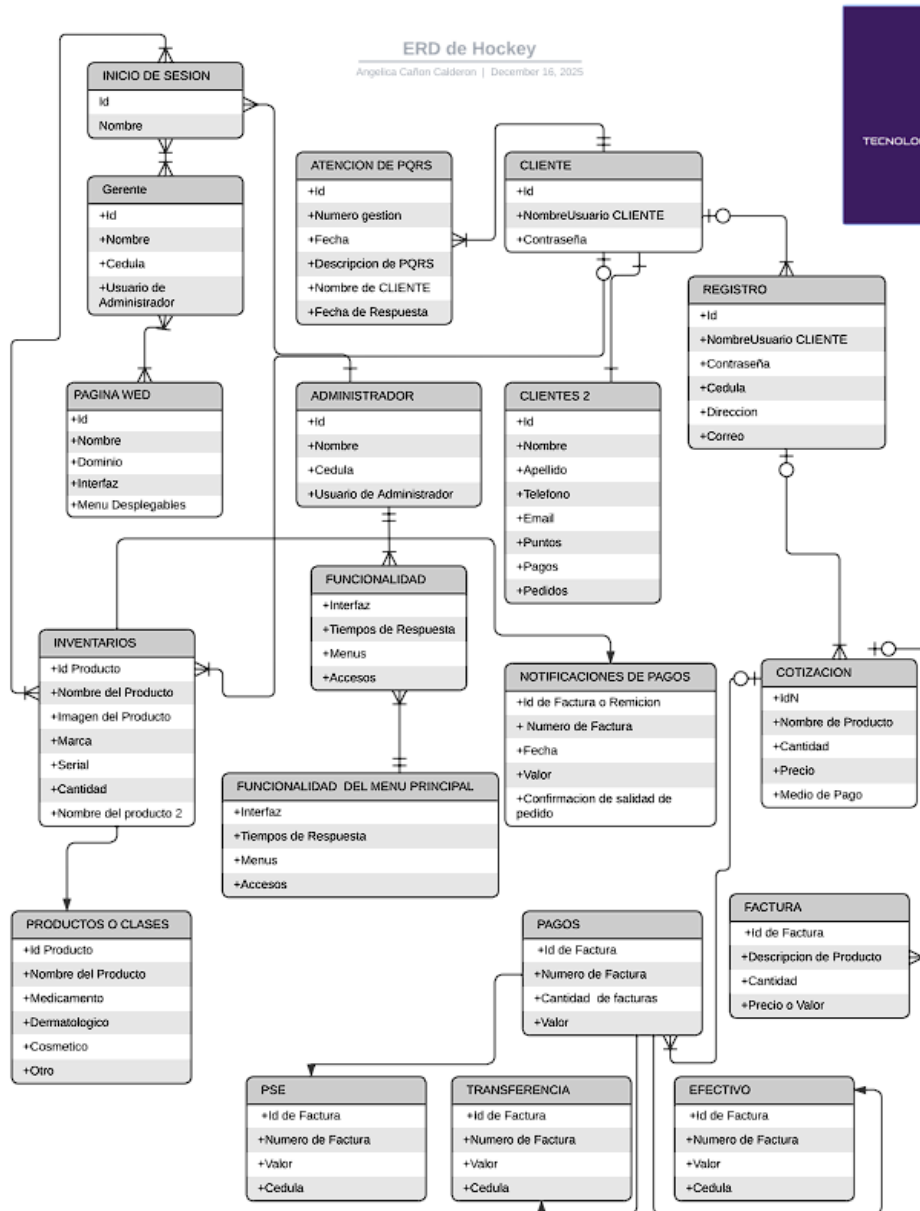


Diagrama de clases



[illegible]

```

graph LR
    subgraph DigiCompra
        direction TB
        IS1([Iniciar Sesión])
        GA1([Gestionar Administradores])
        IG1([Gestionar Gerentes])
        GI1([Gestionar Inventario])
        GP1([Gestionar Productos])
        VC1([Ver Cotizaciones])
        RC1([Realizar Compras])
        PF1([Pagar Factura])
        PC1([Pebar Compras])
        PF2([Pagar Factura])
    end

    Gerente --> IS1
    Administrador --> GA1
    Administrador --> IG1
    Empleado --> IG1
    Empleado --> GI1
    Cliente --> AT([Atender PQRs])
    Cliente --> VC2([Ver Cotizaciones])
    Cliente --> RC2([Realizar Compras])

    IS1 --> IS2([Iniciar Sesión])
    GA1 --> GA2([Gestionar Administradores])
    IG1 --> IG2([Gestionar Gerentes])
    GI1 --> GI2([Gestionar Inventario])
    GP1 --> GP2([Gestionar Productos])
    VC1 --> VC3([Ver Cotizaciones])
    RC1 --> RC3([Realizar Compras])
    PF1 --> PF3([Pagar Factura])
    PC1 --> PC2([Pebar Compras])

    IS2 --> GA2
    GA2 --> IG2
    IG2 --> GI2
    GI2 --> GP2
    GP2 --> VC3
    VC3 --> RC3
    RC3 --> PF3
    PF3 --> PC2
    PC2 --> PF3

    IG2 -.->|<<include>>| RPSE([Pago PSE])
    IG2 -.->|<<include>>| TB([Transferencia Bancaria])
    IG2 --> PE([Pago en Efectivo])

    RC3 -.->|<<include>>| APQR([Atención PQRs])
    PF3 --> APQR2([Atención PQRs])
  
```

Diagrama de Paquetes – DigiCompra

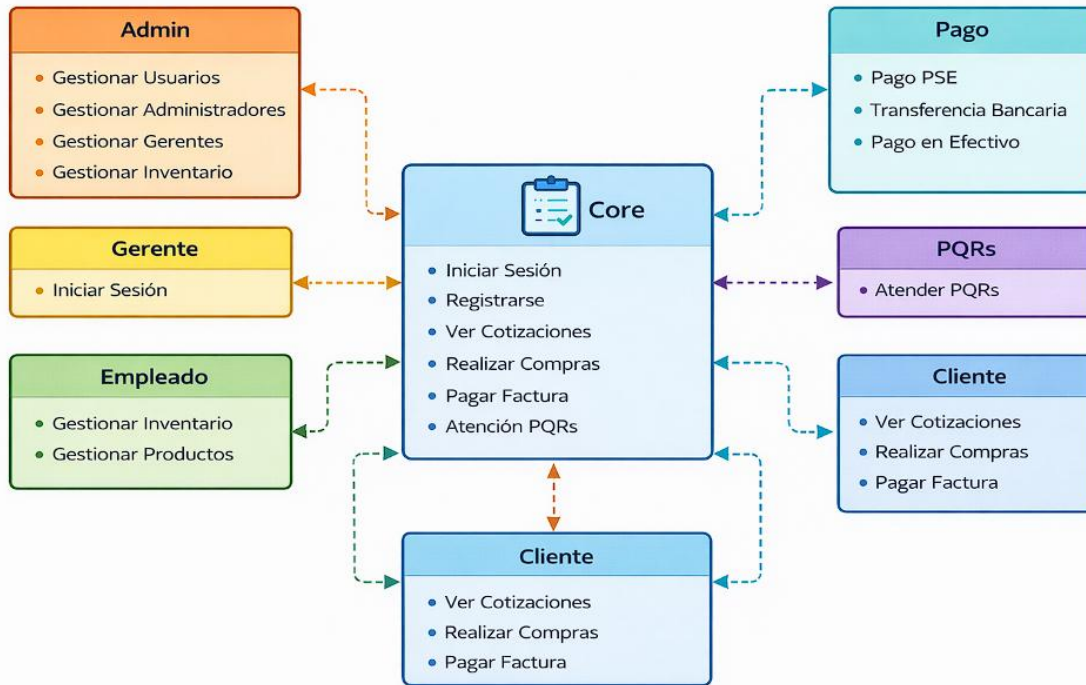
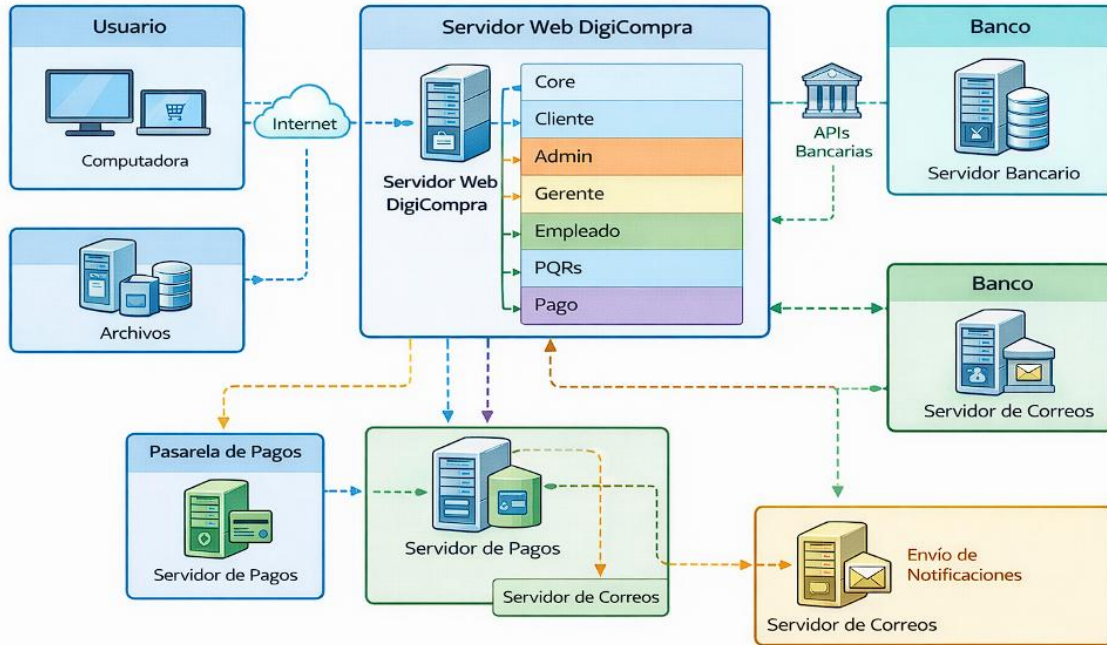


Diagrama de Despliegue – DigiCompra



Componentes del proyecto

Página Web DIGICOMPRA

1. Título del proyecto

Desarrollo de la página web DIGICOMPRA para la compra y venta de productos en línea.

2. Planteamiento del problema

Muchas personas no cuentan con una plataforma digital sencilla y segura para realizar compras en línea. Esto limita el acceso a productos y reduce las oportunidades de comercio digital.

3. Justificación

El proyecto DIGICOMPRA es importante porque permite a los usuarios comprar productos de forma rápida y segura desde cualquier lugar. Además, impulsa el uso de la tecnología y el comercio electrónico.

4. Objetivos

Objetivo general

Crear una página web llamada DIGICOMPRA que facilite la compra y venta de productos en línea.

Objetivos específicos

- Diseñar una interfaz amigable para el usuario.
- Implementar un catálogo de productos.
- Desarrollar un sistema de carrito de compras.
- Garantizar la seguridad de los datos del usuario.

5. Alcance

El proyecto incluye el diseño y desarrollo de la página web DIGICOMPRA con funciones básicas de comercio electrónico. No incluye aplicación móvil ni entrega física de productos.

6. Funcionalidades principales

- Registro e inicio de sesión de usuarios
- Visualización de productos
- Carrito de compras
- Proceso de pago
- Confirmación de compra

7. Metodología

El proyecto se desarrollará por etapas: análisis, diseño, desarrollo, pruebas y publicación.

8. Recursos

Humanos: desarrollador web y administrador

Tecnológicos: computadora, internet, software de desarrollo

Financieros: dominio, hosting y mantenimiento

9. Cronograma

Las actividades se organizarán en un periodo determinado, distribuidas por fases.

10. Resultados esperados

- Página web DIGICOMPRA funcionando correctamente
- Usuarios registrados
- Compras realizadas de forma segura

Metodología utilizada

Página Web DIGICOMPRA

Para el desarrollo de la página web DIGICOMPRA se utilizó una **metodología de desarrollo por fases**, que permite organizar el trabajo de manera ordenada y eficiente, garantizando el correcto funcionamiento del sistema.

1. Análisis de requisitos

En esta fase se identificaron las necesidades de los usuarios y los objetivos de la página web. Se definieron las funcionalidades principales, como el registro de usuarios, el catálogo de productos, el carrito de compras y el sistema de pago.

2. Diseño del sistema

Se diseñó la estructura de la página web, incluyendo la interfaz gráfica, la navegación y la distribución de los contenidos. Se buscó que el diseño fuera atractivo, intuitivo y fácil de usar para los usuarios.

3. Desarrollo

En esta etapa se realizó la programación de la página web utilizando tecnologías web como HTML, CSS y JavaScript, junto con un lenguaje de servidor y una base de datos. Se implementaron las funcionalidades definidas en la fase de análisis.

4. Pruebas

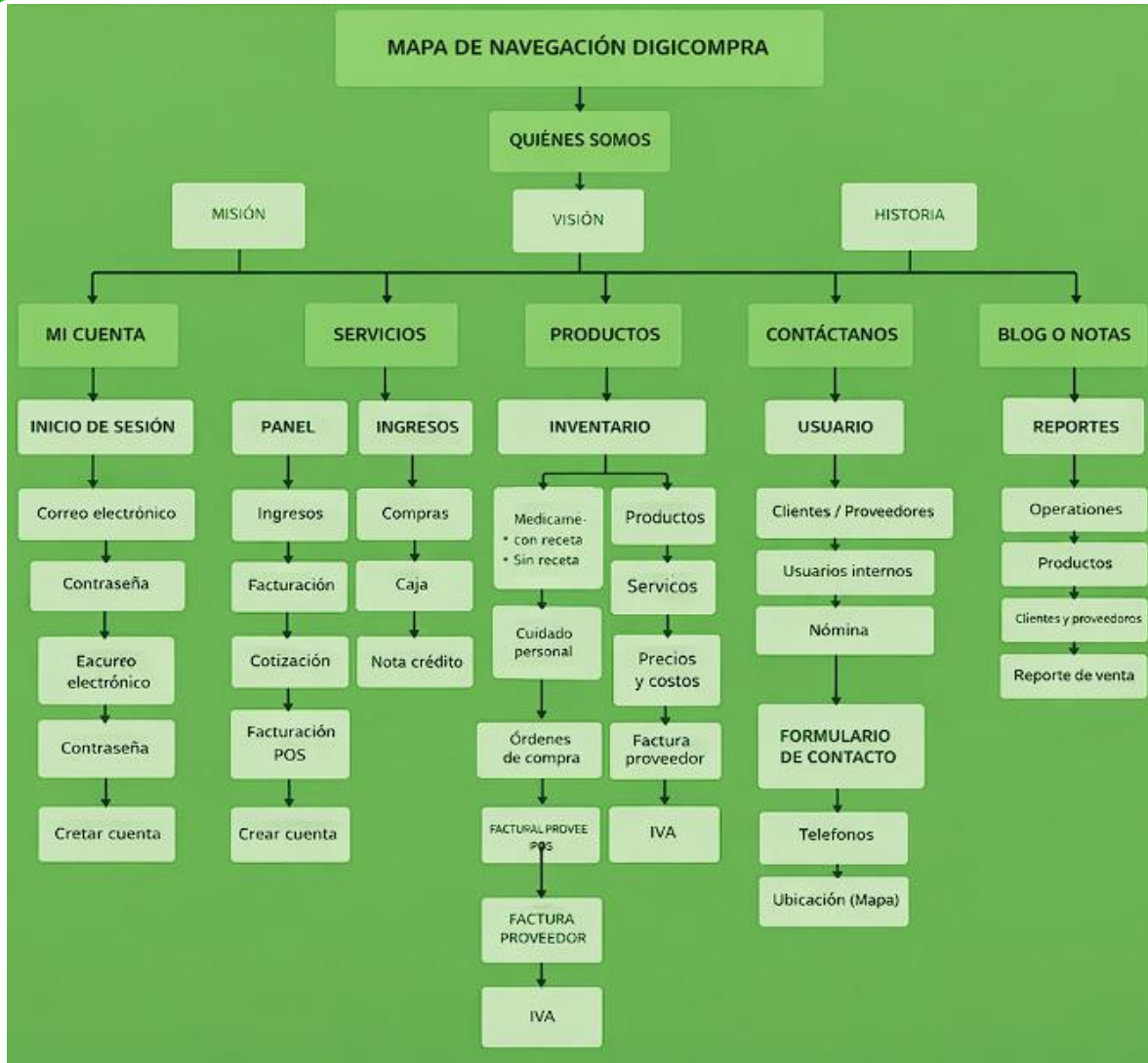
Se realizaron pruebas para verificar el correcto funcionamiento de cada sección de la página web, asegurando que el registro de usuarios, el carrito de compras y el proceso de pago funcionen correctamente y sin errores.

5. Implementación

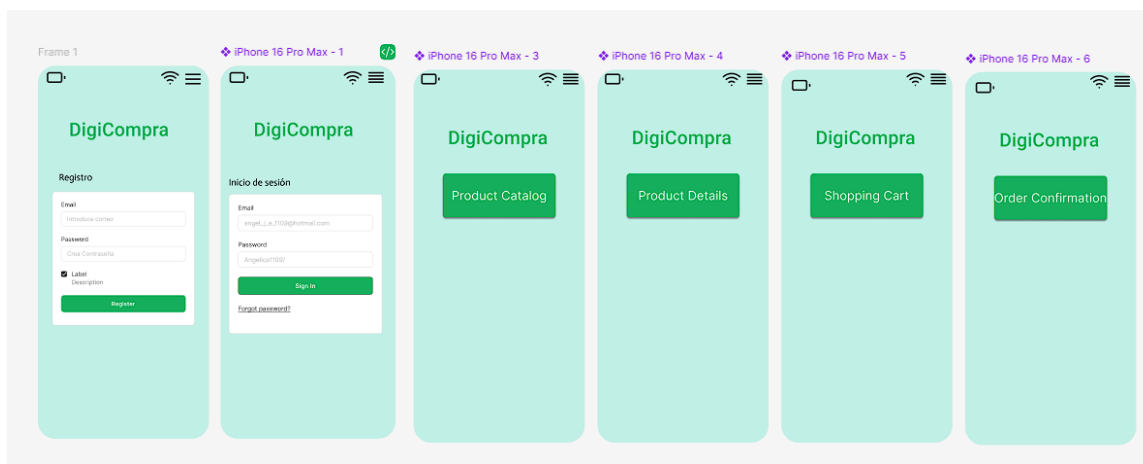
Una vez finalizadas las pruebas, la página web DIGICOMPRA fue publicada en un servidor web, quedando disponible para su uso por los usuarios.

6. Mantenimiento

Se contempla el mantenimiento continuo de la página web para corregir errores, actualizar productos y mejorar la seguridad y el rendimiento del sistema.



codificación del Fron-end -Prototipo



Lenguaje de etiquetado HTML con CSS y JAVASCRIPT

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Sobre nosotros - Digicompra</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <div class="container">
10     <header>
11       <h1>Digicompra</h1>
12     </header>
13
14     <h2>Sobre nosotros</h2>
15
16     <div class="card">
17       <p>
18         En Digicompra nos dedicamos a ser una plataforma digital que hace que comprar productos sea
19         rápido, seguro y accesible. Nuestro objetivo es ofrecerte una experiencia de usuario moderna,
20         confiable y eficiente, para que puedas encontrar todo lo que necesitas desde la comodidad de
21         tu hogar o negocio.
22       </p>
23     </div>
24   </div>
25 </body>
26 </html>
  
```

```

1 function login() {
2   const email = document.getElementById("email").value;
3   const pass = document.getElementById("password").value;
4
5   if (email === "" || pass === "") {
6     alert("Por favor complete todos los campos.");
7     return;
8   }
9
10  alert("Inicio de sesión exitoso");
11  window.location.href = "catalog.html";
12 }
13
14 function registerUser() {
15   const email = document.getElementById("email").value;
16   const pass = document.getElementById("password").value;
17
18   if (email === "" || pass === "") {
19     alert("Por favor complete todos los campos.");
20     return;
21   }
22
23   alert("Registro exitoso");
24   window.location.href = "login.html";
25 }
26
27 function goToProduct() {
28   window.location.href = "product.html";
29 }
30
31 function addToCart() {
32   alert("Producto agregado al carrito");
33   window.location.href = "cart.html";
34 }
35
36 function finalizeOrder() {
37   alert("Pedido confirmado");
38   window.location.href = "confirm.html";
39 }
40
  
```

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Carrito - DigiCompra</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <div class="container">
10     <header>
11       <h1>DigiCompra</h1>
12     </header>
13
14     <h2>Shopping Cart</h2>
15
16     <div class="card">
17       <p>1x Analgésico 500mg - $10.000</p>
18       <button onclick="finalizeOrder()">Confirmar pedido</button>
19     </div>
20   </div>
21   <script src="app.js"></script>
22 </body>
23 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Catálogo - DigiCompra</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <div class="container">
10     <header>
11       <h1>DigiCompra</h1>
12     </header>
13
14     <h2>Product Catalog</h2>
15
16     <div class="card">
17       <p>Aquí se mostrará el catálogo de productos.</p>
18       <button onclick="goToProduct()">Ver producto</button>
19     </div>
20   </div>
21   <script src="app.js"></script>
22 </body>
23 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Confirmación - DigiCompra</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <div class="container">
10     <header>
11       <h1>DigiCompra</h1>
12     </header>
13     <h2>Order Confirmation</h2>
14     <div class="card">
15       <p>Tu pedido ha sido confirmado exitosamente.</p>
16       <button onclick="window.location.href='index.html'">Volver al Inicio</button>
17     </div>
18   </div>
19 </body>
20 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>DigiCompra - Inicio</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <div class="container">
10     <header>
11       <h1>DigiCompra</h1>
12       <nav>
13         <a href="index.html">Inicio</a>
14         <a href="register.html">Registro</a>
15         <a href="login.html">Login</a>
16         <a href="catalog.html">Catálogo</a>
17         <a href="about.html">Nosotros</a>
18       </nav>
19     </header>
20     <div class="card">
21       <h2>Bienvenido</h2>
22       <p>
23         DigiCompra es una plataforma digital orientada a la venta de productos de droguería,
24         permitiendo al usuario consultar productos, agregar al carrito y confirmar su pedido.
25       </p>
26       <button class="btn-primary btn-block" onclick="window.location.href='login.html'">
27         Ir a inicio de sesión
28       </button>
29     </div>
30   </div>
31 </body>
32 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <title>login - DigiCompra</title>
6 <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9 <div class="container">
10 <header>
11 <h1>DigiCompra</h1>
12 </header>
13 <h2>Inicio de sesión</h2>
14 <div class="card">
15 <label for="email">Email</label>
16 <input type="email" id="email" placeholder="ejemplo@correo.com">
17 <label for="password">Password</label>
18 <input type="password" id="password" placeholder="Contraseña">
19 <button onclick="login()">Iniciar sesión</button>
20 </div>
21 <p style="text-align:center;">
22 <a href="register.html">No tienes cuenta? Registrarse</a>
23 </p>
24 </div>
25 <script src="app.js"></script>
26 </body>
27 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <title>Detalle de producto - DigiCompra</title>
6 <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9 <div class="container">
10 <header>
11 <h1>DigiCompra</h1>
12 </header>
13 <h2>Product Details</h2>
14 <div class="card">
15 <p>Nombre del producto: Analgésico 500mg</p>
16 <p>Precio: $10.000</p>
17 <button onclick="addToCart()">Agregar al carrito</button>
18 </div>
19 </div>
20 <script src="app.js"></script>
21 </body>
22 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Registro - Digicompra</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <div class="container">
10     <header>
11       <h1>Digicompra</h1>
12     </header>
13     <h2>Registro</h2>
14     <div class="card">
15       <label for="email">Email</label>
16       <input type="email" id="email" placeholder="Ingrese su correo">
17       <label for="password">Password</label>
18       <input type="password" id="password" placeholder="Cree una contraseña">
19       <label for="label">Label</label>
20       <input type="text" id="label" placeholder="Descripción">
21       <button onclick="registerUser()">Registrar</button>
22     </div>
23     <p style="text-align:center;">
24       ¿Ya tienes cuenta?
25       <a href="login.html">Iniciar sesión</a>
26     </p>
27   </div>
28   <script src="app.js"></script>
29 </body>
30 </html>

```

```

1 body {
2   margin: 0;
3   padding: 0;
4   font-family: Arial, sans-serif;
5   background-color: #f0f0f0;
6 }
7
8 .container {
9   width: 90%;
10  max-width: 420px;
11  margin: auto;
12  padding: 20px 0 40px 0;
13 }
14
15 h1, h2, h3 {
16   text-align: center;
17   color: #808080;
18 }
19
20 header {
21   text-align: center;
22   margin-bottom: 20px;
23 }
24
25 nav a {
26   margin: 0 5px;
27   color: #808080;
28   text-decoration: none;
29   font-size: 14px;
30 }
31
32 nav a:hover {
33   text-decoration: underline;
34 }
35
36 input, button {
37   width: 100%;
38   padding: 10px;
39   margin-top: 10px;
40   border-radius: 6px;
41   border: 1px solid #ccc;
42   box-sizing: border-box;
43 }
44
45 button {
46   background-color: #808080;
47   color: white;
48   font-size: 16px;

```



```

# styles.css
body
  input, button {
    width: 100%;
    padding: 12px;
    margin-top: 10px;
    border-radius: 6px;
    border: 1px solid #ccc;
    box-sizing: border-box;
  }
  button {
    background: #00a0f4;
    color: white;
    font-size: 16px;
    border: none;
    cursor: pointer;
  }
  button:hover {
    background: #0070c0;
  }
  .card {
    background: white;
    padding: 15px;
    border-radius: 10px;
    margin: 15px 0;
    box-shadow: 0 2px 5px rgba(0,0,0,0.15);
  }
  .btn-primary {
    background: #00a0f4;
    color: white;
    border: none;
  }
  .btn-block {
    width: 100%;
    display: block;
    text-align: center;
  }
  p {
    font-size: 14px;
    line-height: 1.5;
  }

```

digicompra_frontend (8)

Inicio OneDrive > videos > OneDrive > Escritorio > digicompra_frontend (8)

Buscar en digicompra_frontend (8)

Nuevo > Ordenar > Ver > Detalles

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
about.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
app.js		11/12/2025 9:14 p. m.	Archivo de origen ...	1 KB
cart.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
catalog.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
confirm.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
index.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
login.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
product.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
register.html		11/12/2025 9:14 p. m.	Chrome HTML Do...	1 KB
styles.css		11/12/2025 9:14 p. m.	Archivo de origen ...	2 KB

digicompra_frontend (8) (10 elementos)

Selecciona un único archivo para obtener más información y compartir el contenido de la nube.

Frameworks y Tecnologías Utilizadas – DigiCompra

La página web **DigiCompra** fue desarrollada como un **prototipo Front-end**, por lo que se utilizaron tecnologías ligeras y herramientas orientadas a la visualización y validación del diseño.

1. HTML5 (Lenguaje base)

- Utilizado para la **estructura de las páginas**
- Define formularios, botones, textos y enlaces
- Permite la navegación entre pantallas

- ☐ Estándar web
- ☐ No requiere framework adicional

2. CSS3 (Estilos visuales)

- Encargado del **diseño visual**
- Colores, tamaños, bordes, tipografía y espaciado
- Diseño **mobile-first** basado en Figma

- ☐ Uso de variables CSS
- ☐ Flexbox para alineación
- ☐ Sin framework CSS externo

3. JavaScript (Básico – opcional)

- Usado solo para **simular navegación**
- No hay lógica de negocio ni backend
- Permite interacción simple (redirecciones)

- ☐ JavaScript Vanilla
- ☐ Sin frameworks como React o Angular

4. Live Server (Extensión de VS Code)

- Framework de **entorno de desarrollo local**
- Permite visualizar el proyecto en el navegador

- Recarga automática al guardar cambios

- ☐ No afecta el código final
- ☐ Solo para desarrollo

5. GitHub Pages (Despliegue)

- Plataforma para **publicar el prototipo**
- Convierte el proyecto en una web accesible por enlace
- Ideal para presentaciones académicas

- ☐ Hosting estático
- ☐ Gratuito

Resumen en tabla (ideal para tu informe)

TIPO	FRAMEWORK / TECNOLOGÍA
Estructura	HTML5
Estilos	CSS3
Interactividad	JavaScript (Vanilla)
Desarrollo	Live Server (VS Code)
Despliegue	GitHub Pages
Diseño base	Figma

Configuración del Servidor y de la Base de Datos

En el desarrollo del proyecto **DigiCompra**, no se implementó un servidor backend ni una base de datos real, ya que el alcance del proyecto corresponde a un **prototipo Front-end**, cuyo objetivo principal es la validación del diseño y la experiencia de usuario.

Configuración del Servidor

Durante la fase de desarrollo se utilizó un **servidor local** proporcionado por la extensión **Live Server** en Visual Studio Code, el cual permitió:

- Ejecutar el proyecto en un entorno web local
- Visualizar los cambios en tiempo real
- Simular el funcionamiento de una aplicación web

Posteriormente, el prototipo fue desplegado utilizando **GitHub Pages**, que actúa como un **servidor web estático**, permitiendo el acceso al sistema mediante un enlace público sin necesidad de instalación local.

Tipo de servidor utilizado:

- Servidor local (Live Server – VS Code)
- Servidor estático en producción (GitHub Pages)

Configuración de la Base de Datos

El proyecto **no utiliza una base de datos real**, debido a que:

- No se implementó un backend
- No se requiere almacenamiento persistente de información
- Los datos de usuarios, productos y compras son **simulados**

La información visualizada en el prototipo es **estática**, definida directamente en el código HTML, cumpliendo con el objetivo de demostrar el flujo del sistema sin realizar operaciones reales como registros, consultas o pagos.

Base de datos utilizada:

- ☐ No se utiliza base de datos
- ☒ Datos simulados (mock data)

Justificación técnica

La decisión de no implementar servidor backend ni base de datos se tomó para mantener el enfoque del proyecto en:

- El diseño de interfaces (UI)
- La experiencia de usuario (UX)
- La navegación entre pantallas
- La fidelidad visual respecto al prototipo diseñado en Figma

Esta metodología es común en proyectos académicos y en etapas tempranas de desarrollo de software.

Resumen

COMPONENTE	TECNOLOGÍA
Servidor local	Live Server (VS Code)
Servidor web	GitHub Pages
Base de datos	No implementada
Tipo de datos	Simulados (estáticos)

Versión corta (si el profesor pide algo breve)

El proyecto DigiCompra no cuenta con un servidor backend ni una base de datos real, ya que corresponde a un prototipo Front-end. Para la visualización se utilizó Live Server en Visual Studio Code y para el despliegue final GitHub Pages. Los datos son simulados y no se almacenan de forma persistente.

Diseño de la Base de Datos – DigiCompra

Aunque el proyecto **DigiCompra** fue desarrollado como un **prototipo Front-end**, se propone un **diseño de base de datos relacional** que permitiría, en una futura implementación, gestionar usuarios, productos y compras de manera estructurada.

El diseño está pensado para un sistema de **comercio electrónico básico**.

Entidades Principales

1Usuario

Almacena la información de los usuarios registrados.

Campos:

- id_usuario (PK)
- nombre
- email



- contraseña
- fecha_registro



2Producto

Contiene la información de los productos disponibles en la tienda.

Campos:

- id_producto (PK)
- nombre
- descripcion
- precio
- stock

3Pedido

Registra las compras realizadas por los usuarios.

Campos:

- id_pedido (PK)
- id_usuario (FK)
- fecha_pedido
- total

4Detalle_Pedido

Relaciona los pedidos con los productos comprados.

Campos:

- id_detalle (PK)
- id_pedido (FK)
- id_producto (FK)
- cantidad
- subtotal

Relaciones

- Un **usuario** puede realizar **muchos pedidos**
- Un **pedido** puede contener **muchos productos**
- Un **producto** puede estar en **muchos pedidos**

(Relación muchos a muchos resuelta con Detalle_Pedido)

Script SQL de la Base de Datos (MySQL)

-- Crear base de datos

```
CREATE DATABASE digicompra_db;
```

```
USE digicompra_db;
```

-- Tabla Usuarios

```
CREATE TABLE usuarios (  
  id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL,  
  email VARCHAR(100) NOT NULL UNIQUE,  
  contraseña VARCHAR(255) NOT NULL,  
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

-- Tabla Productos

```
CREATE TABLE productos (  
  id_producto INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL,  
  descripcion TEXT,  
  precio DECIMAL(10,2) NOT NULL,  
  stock INT NOT NULL  
);
```

-- Tabla Pedidos

```
CREATE TABLE pedidos (  
  id_pedido INT AUTO_INCREMENT PRIMARY KEY,  
  id_usuario INT NOT NULL,  
  fecha_pedido TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  total DECIMAL(10,2) NOT NULL,  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)  
);
```

-- Tabla Detalle de Pedidos

```
CREATE TABLE detalle_pedidos (  
  id_detalle INT AUTO_INCREMENT PRIMARY KEY,
```

```
id_pedido INT NOT NULL,  
id_producto INT NOT NULL,  
cantidad INT NOT NULL,  
subtotal DECIMAL(10,2) NOT NULL,  
FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido),  
FOREIGN KEY (id_producto) REFERENCES productos(id_producto)  
);
```

Tipo de Base de Datos

- **Modelo:** Relacional
- **Motor propuesto:** MySQL / MariaDB
- **Normalización:** Tercera Forma Normal (3FN)

Texto listo para el informe

El diseño de la base de datos de DigiCompra fue elaborado de forma conceptual, ya que el proyecto corresponde a un prototipo Front-end. Se propuso una base de datos relacional orientada a la gestión de usuarios, productos y pedidos, utilizando tablas relacionadas mediante claves primarias y foráneas. El diseño permitiría una futura implementación completa del sistema.

CONCLUSIÓN

El desarrollo del proyecto **DigiCompra** permitió aplicar de manera práctica los conocimientos adquiridos en el área de desarrollo web, específicamente en la creación de un **prototipo Front-end** orientado a una plataforma de comercio electrónico. A partir del diseño elaborado en **Figma**, se logró implementar una interfaz funcional utilizando **HTML5 y CSS3**, respetando colores, tamaños y estructura definidos en el prototipo visual.

Asimismo, se simuló el flujo completo de navegación del sistema, incluyendo registro, inicio de sesión, catálogo de productos, carrito de compras y confirmación de pedido, lo que facilitó la validación de la experiencia de usuario sin necesidad de implementar un backend real. El uso de herramientas como **Visual Studio Code**, **Live Server** y **GitHub Pages** permitió desarrollar, visualizar y desplegar el proyecto de manera eficiente.

Finalmente, aunque no se implementó una base de datos funcional, se propuso un **diseño de base de datos relacional** adaptado a **PostgreSQL**, dejando establecida una base sólida para una futura ampliación del sistema. En conclusión, DigiCompra cumple con los objetivos planteados como prototipo académico, demostrando un adecuado manejo de tecnologías Front-end y metodologías de desarrollo web.

REFERENCIAS

- GitHub. (2024). *GitHub Pages Documentation*.
<https://docs.github.com/en/pages>
- Figma. (2024). *Figma: The Collaborative Interface Design Tool*.
<https://www.figma.com>
- Mozilla Developer Network (MDN). (2024). *HTML5 Reference*.
<https://developer.mozilla.org/es/docs/Web/HTML>
- Mozilla Developer Network (MDN). (2024). *CSS3 Reference*.
<https://developer.mozilla.org/es/docs/Web/CSS>
- PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*.
<https://www.postgresql.org/docs/>
- Visual Studio Code. (2024). *Visual Studio Code Documentation*.
<https://code.visualstudio.com/docs>