

Trabajo Integrador

CARRERA: Ingeniería en Sistemas de la Información.

MATERIA: Paradigmas y Lenguajes de Programación III.

COMISIÓN: “U” (única).

PROFESOR: Mgter. Encina Agustín.

ESTUDIANTE: Vera, Romina.

FECHA: 18-11-2025.



Introducción

El presente informe describe el desarrollo de "Digital Point", un sitio web de comercio electrónico orientado a la venta de productos tecnológicos.

El proyecto integra frontend, backend en PHP, base de datos MySQL (sobre XAMPP) y un conjunto de páginas especializadas: catálogo, carrito, proceso de pago, perfil de usuario, preguntas frecuentes, blog, panel de administración y páginas legales (términos y condiciones, políticas de privacidad y cookies).

Además del funcionamiento básico de una tienda online, el sitio incorpora diseño responsivo, animaciones suaves, componentes reutilizables (breadcrumbs, tarjetas, modales, toasts) y secciones específicas para gestionar usuarios, pedidos, contenidos del blog y ofertas desde un entorno de administración.



Desarrollo

Descripción del proyecto

El proyecto implementa una tienda online completa con páginas PHP para la portada, listados de productos, fichas de detalle, ofertas, blog, secciones institucionales y flujo de compra (carrito, checkout y proceso de pago), reutilizando componentes comunes mediante includes. La capa de JavaScript modulariza la lógica de interacción: gestión de carrito, filtrado y ordenamiento de productos, sliders, validaciones de formularios, contadores regresivos y manejo de sesiones en el lado del cliente. La arquitectura CSS se organiza en una hoja global y estilos por página, asegurando una estética uniforme, responsiva y alineada con el prototipo de diseño. La base de datos MySQL almacena usuarios, productos, categorías, ofertas, pedidos y detalles de compra, permitiendo el funcionamiento del e-commerce y la gestión desde el panel de administración.

Objetivos del Sistema

El objetivo general del proyecto es diseñar e implementar un prototipo funcional de plataforma de e-commerce que integre diseño de interfaz, lógica de negocio y persistencia de datos, utilizando tecnologías web estándar. En particular, se busca transformar el prototipo visual en una solución navegable que permita recorrer el catálogo, gestionar un carrito de compras, simular el proceso de pago y administrar contenidos (ofertas y blog) desde un backend sencillo, aplicando buenas prácticas de programación modular en PHP, JavaScript y CSS, y un modelo de datos relacional en MySQL.



Análisis de Requerimientos

Requerimientos Funcionales

Gestión de Usuarios:

- Registro de nuevos usuarios mediante correo electrónico y contraseña.
- Inicio de sesión seguro para usuarios registrados.
- Panel de control personal con información de cuenta.

Navegación y catálogo de productos:

- Visualización del catálogo organizado por categorías principales.
- Subcategorización especializada.
- Sistema de búsqueda con filtros avanzados por precio, relevancia y categoría.
- Opciones de vista alternativa (lista vertical o cuadrícula).
- Botones de navegación rápida y acceso a productos destacados.

Sistema de carrito de compras:

- Añadir productos al carrito desde el catálogo o página de detalle.
- Modificar cantidades de productos en el carrito.
- Eliminar productos individuales del carrito.
- Actualizar el carrito con cambios realizados.
- Vaciar completamente el carrito.
- Mostrar subtotal, costos de envío y total actualizado.
- Funcionalidad "Seguir comprando" con redirección al catálogo.



Proceso de Pago:

- Formulario de pago con datos personales completos.
- Selección de método de envío (tarifa estándar o retiro en local).
- Cálculo automático de costos de envío según datos ingresados.
- Resumen detallado de la orden antes de confirmar la compra.

Atención al cliente y contacto:

- Sección "Contactarnos" con formulario interactivo.
- Múltiples canales de comunicación disponibles.
- Sección informativa "About Us" y "Términos y Condiciones".
- Sistema de suscripción para ofertas y promociones.
- Funcionalidad "Seguir comprando" con redirección al catálogo.

Requerimientos No Funcionales

Rendimiento:

- Tiempo de respuesta del buscador inferior a 3 segundos.
- Tiempo de carga de páginas no superior a 4 segundos.
- Optimización para diferentes dispositivos y velocidades de conexión.

Usabilidad:

- Interfaz intuitiva adaptable a usuarios de diferentes niveles técnicos
 - Diseño responsivo para múltiples dispositivos.
 - Navegación clara y consistente en todas las secciones.
-

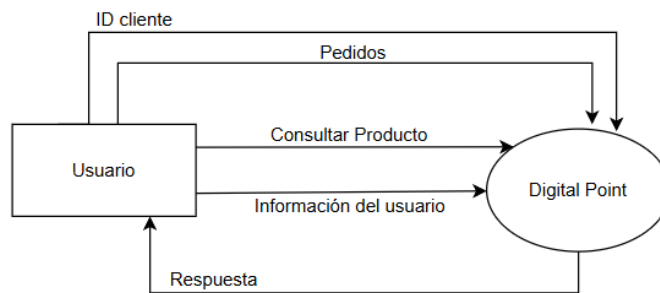


Modelado del sistema

Diagrama de flujo de datos nivel 0

Figura 1

Diagrama de flujo de datos nivel 0 del sistema Digital Point



Nota. Elaboración Propia.

En el DFD Nivel 0 se representa al sistema Digital Point como un único proceso central que interactúa con el Usuario, quien es la entidad externa principal.

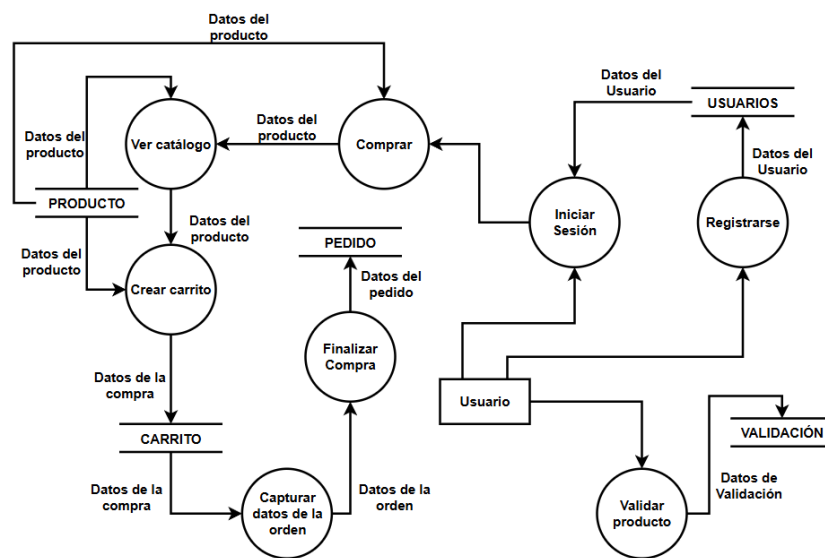
- Entradas al sistema:
 - Información del usuario: datos de registro, inicio de sesión o perfil.
 - Consultar producto: búsquedas o solicitudes para visualizar el catálogo.
 - Pedidos: solicitudes de compra con productos seleccionados.
- Salidas hacia el usuario:
 - ID del cliente: confirmación de registro o autenticación.
 - Respuesta: confirmaciones de acciones (registro, compra, suscripción, etc.).
 - Producto entregado: confirmación de que la orden fue procesada.

Este nivel resume el alcance del sistema: el usuario consulta, se registra y compra; mientras que Digital Point responde con confirmaciones y entrega virtual de los pedidos.

Diagrama de flujo de datos de las funciones del usuario

Figura 2

Diagrama de flujo de de funciones del usuario



Nota. Elaboración Propia.

En este diagrama de flujo de datos, se desglosan en detalle las acciones que puede realizar el usuario dentro del sistema.

1. Registrarse / Iniciar sesión.
 - El usuario envía sus datos de registro o login.
 - El sistema los valida y guarda en el almacén de USUARIOS.
 - Se genera un ID de cliente que luego se usa para sus compras.
2. Ver catálogo.



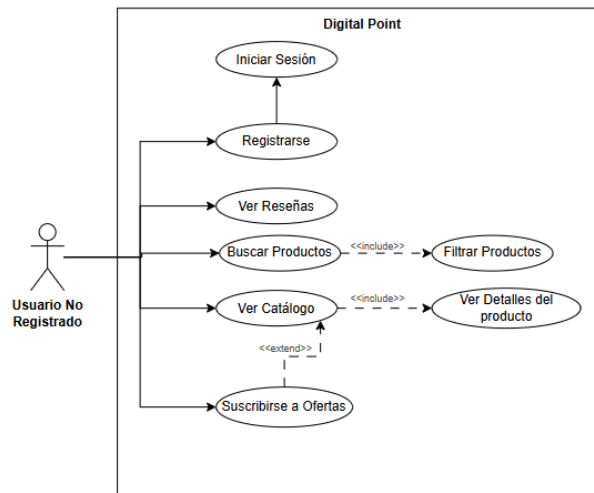
- El usuario solicita consultar productos.
 - El sistema obtiene la información desde el almacén de PRODUCTOS.
 - Se devuelven datos como nombre, precio y disponibilidad.
3. Crear carrito.
- El usuario selecciona productos para añadirlos al carrito.
 - La información se almacena en el repositorio de CARRITO.
4. Comprar.
- El usuario confirma la compra de productos seleccionados.
 - Se genera un pedido y se almacena en PEDIDO.
5. Finalizar compra.
- El sistema cierra el carrito y convierte los productos seleccionados en una orden definitiva.
 - Se vinculan datos del usuario y de los productos.
6. Validar producto.
- Antes de confirmar, el sistema verifica la disponibilidad o datos de los productos.
 - Los resultados son devueltos al proceso de compra.
7. Capturar datos de la orden.
- Se ingresan datos del pago y se almacenan en el repositorio de PAGO.

Diagrama de caso de uso del usuario no registrado

Figura 3



Caso de uso: Usuario no registrado



Nota. Elaboración Propia.

Descripción

Este caso de uso describe las interacciones del usuario no registrado con la plataforma Digital Point, donde puede visualizar información básica sin necesidad de crear una cuenta.

Caso de Uso.	Acciones del usuario no registrado en la plataforma.
Actores.	Usuario No Registrado.
Tipo.	Primario.
Precondición.	El sistema debe estar disponible y tener productos cargados.
Postcondición.	El usuario puede registrarse, iniciar sesión o continuar navegando sin acceso a funcionalidades avanzadas (compra, gestión de carrito).



Autor.	Vera, Romina.
---------------	---------------

Propósito.

Permitir a los usuarios no registrados visualizar información básica y motivarlos a registrarse.

Resumen.

El usuario no registrado puede ver reseñas, buscar y filtrar productos, acceder al catálogo, ver detalles del producto y suscribirse a ofertas. Si lo desea, puede registrarse y luego iniciar sesión.

Curso Normal.

1	El usuario accede a la plataforma sin cuenta registrada.	2	El usuario puede visualizar reseñas disponibles.
3	El usuario puede buscar productos y filtrarlos.	4	El usuario puede acceder al catálogo y ver detalles de los productos.
5	El usuario, si desea, puede suscribirse a ofertas ingresando su correo.	6	El usuario decide registrarse y crear una cuenta.

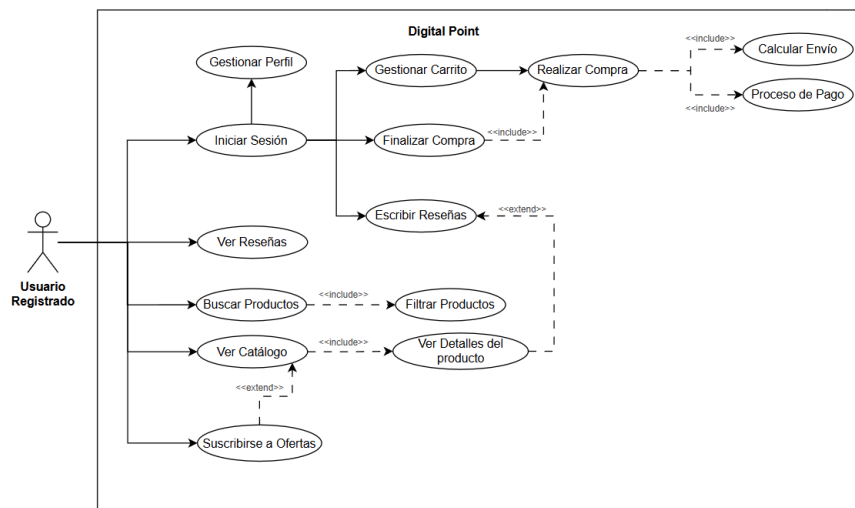


7	El usuario queda habilitado para iniciar sesión.
---	--

Diagrama de caso de uso de usuario registrado

Figura 4

Caso de uso: Usuario registrado



Nota. Elaboración Propia.

Descripción

Este caso de uso describe las acciones que puede realizar un usuario registrado en Digital Point, incluyendo la compra completa de productos.

Caso de Uso.	Acciones del usuario registrado en la plataforma.
---------------------	---



Actores.	Usuario Registrado.
Tipo.	Primario.
Precondición.	El usuario debe estar registrado en el sistema e iniciar sesión.
Postcondición.	El usuario puede realizar la compra completa de productos.
Autor.	Vera, Romina.

Propósito.

Permitir a los usuarios registrados acceder a todas las funcionalidades, incluyendo la gestión de compras.

Resumen.

El usuario registrado inicia sesión, gestiona su perfil, visualiza productos, agrega productos al carrito, finaliza la compra y realiza el proceso de pago.

Curso Normal.

1	El usuario inicia sesión en la plataforma.	2	El usuario puede acceder al catálogo y seleccionar productos.
3	El usuario busca y filtra productos	4	El usuario visualiza detalles de los

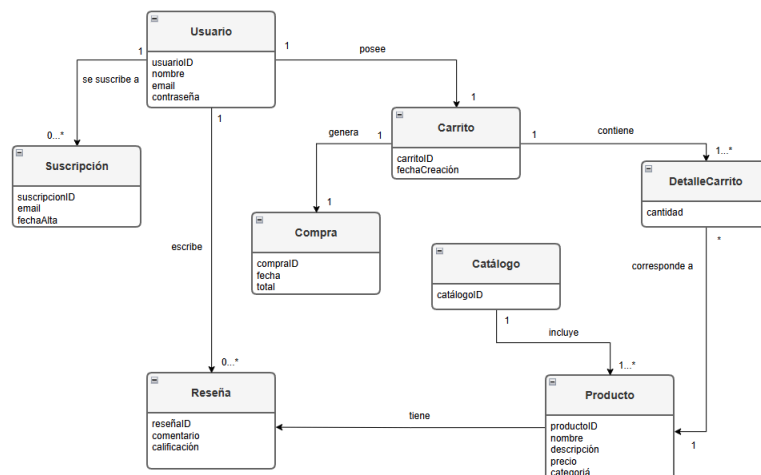


	según sus intereses.		productos elegidos.
5	El usuario crea un carrito de compras y añade productos.	6	El usuario finaliza la compra desde el carrito.
7	El sistema calcula el costo de envío.	8	El usuario completa el formulario de pago correspondiente.

Modelo de dominio del sistema

Figura 5

Modelo de dominio del sistema



Nota. Elaboración Propia.

El modelo de dominio de Digital Point representa las entidades principales del sistema de comercio electrónico y las relaciones que existen entre ellas. A continuación, se detallan las clases identificadas y sus interacciones:



Usuario: Entidad central que representa personas que interactúan con la plataforma. Contiene atributos básicos como usuarioID, nombre, email y contraseña. Establece relaciones con múltiples entidades del sistema.

Suscripción: Almacena información de usuarios registrados para notificaciones y promociones, incluyendo suscripcionID, email y fechaAlta. Relación opcional con Usuario (1 a 0..*).

Carrito: Representa conjuntos de productos seleccionados antes de finalizar compra. Incluye carritoID y fechaCreación. Mantiene relación de composición con Usuario (1 a 1) y se relaciona con Producto a través de DetalleCarrito.

DetalleCarrito: Entidad asociativa que relaciona Carrito y Producto, incluyendo atributo cantidad. Permite gestionar productos específicos dentro del carrito.

Producto: Representa artículos ofrecidos para venta con atributos como productoID, nombre, descripción, precio y categoría. Se organiza dentro de Catálogo y puede recibir múltiples Reseñas.

Compra: Corresponde a transacciones finalizadas por usuarios. Incluye compraID, fecha y total. Se origina en un carrito mediante relación 1 a 1.

Reseña: Contiene valoraciones y comentarios de usuarios sobre productos, con atributos reseñaID, comentario y calificación. Establece relaciones con Usuario y Producto.

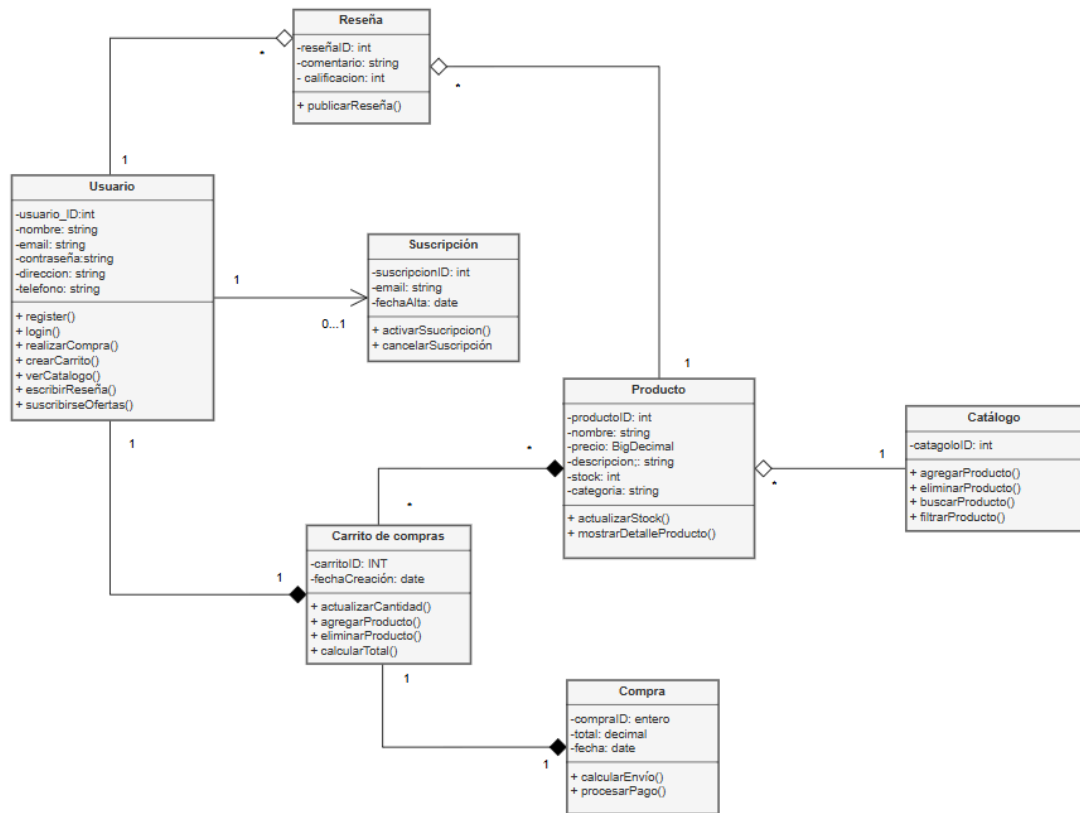
Catálogo: Agrupa y organiza productos de la tienda en categorías jerárquicas. Facilita navegación y búsqueda de productos.

Diagrama de clase del sistema

Figura 6



Diagrama de clases del sistema



Nota. Elaboración Propia.

El diagrama de clases modela la estructura principal del sistema de comercio electrónico **Digital Point**, representando las entidades, sus atributos, métodos y las relaciones que existen entre ellas.

- **Usuario:** es la clase base que contiene los datos esenciales de los clientes. Permite registrarse, iniciar sesión, realizar compras, crear carritos, ver el catálogo, escribir reseñas y suscribirse a ofertas.
 - Se relaciona de forma compuesta con Carrito de compras (cada usuario tiene un carrito activo que depende de él).

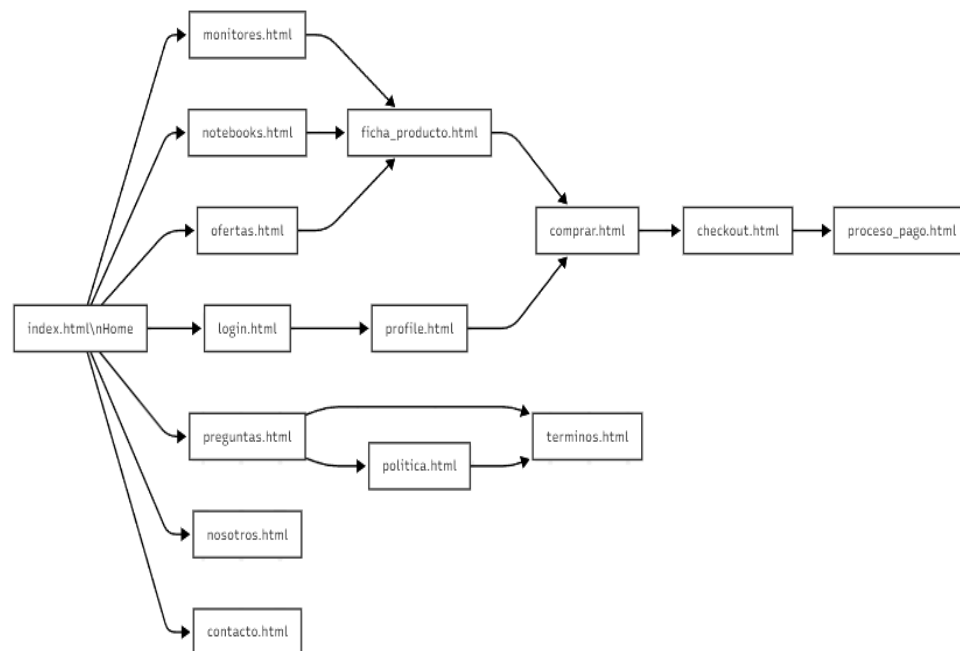


- Además, mantiene una relación de dependencia opcional con Suscripción (un usuario puede tener 0 o 1 suscripción a ofertas).
- **Suscripción:** almacena información sobre las notificaciones y promociones que recibe el usuario. Permite activar o cancelar la suscripción.
- **Carrito de Compras:** representa el conjunto de productos seleccionados por el usuario. Incluye operaciones como agregar, eliminar y actualizar productos, además de calcular el total de la compra.
 - Está en composición con Producto: el carrito depende de los productos que contiene y cada producto dentro del carrito existe sólo como parte de él.
 - Se asocia de forma 1 a 1 con Compra, ya que de un carrito se genera una única transacción.
- **Producto:** representa los artículos a la venta Tiene operaciones para actualizar el stock y mostrar el detalle.
 - Pertenece a un Catálogo (relación de agregación, ya que los productos pueden existir fuera del catálogo).
 - Se relaciona con Reseña, ya que un producto puede recibir múltiples valoraciones de usuarios.
- **Catálogo:** organiza los productos de la tienda en categorías, contiene opciones para agregar, eliminar, buscar y filtrar productos.
- **Compra:** representa la confirmación de un carrito.
- **Reseña:** almacena comentarios y calificaciones de los usuarios sobre los productos.
 - Está en relación de agregación con Usuario (un usuario puede crear reseñas).
 - También se relaciona con Producto (un producto puede tener varias reseñas).

Mapa del sitio

Figura 7

Mapa del sitio



Nota. Elaboración Propia.

El mapa del sitio ubica a Home como hub central hacia Monitores, Notebooks, Ofertas, Preguntas, Nosotros y Contacto. Los listados (Monitores, Notebooks y Ofertas) conducen a una ficha de producto unificada, desde la cual el flujo de compra progresa linealmente: Carrito → Checkout → Pago. La autenticación se resuelve en Login → Perfil, con retorno posible al Carrito para continuar la compra. El bloque de soporte y legales parte de Preguntas y enlaza a Política y Términos (y de Política a Términos), concentrando la

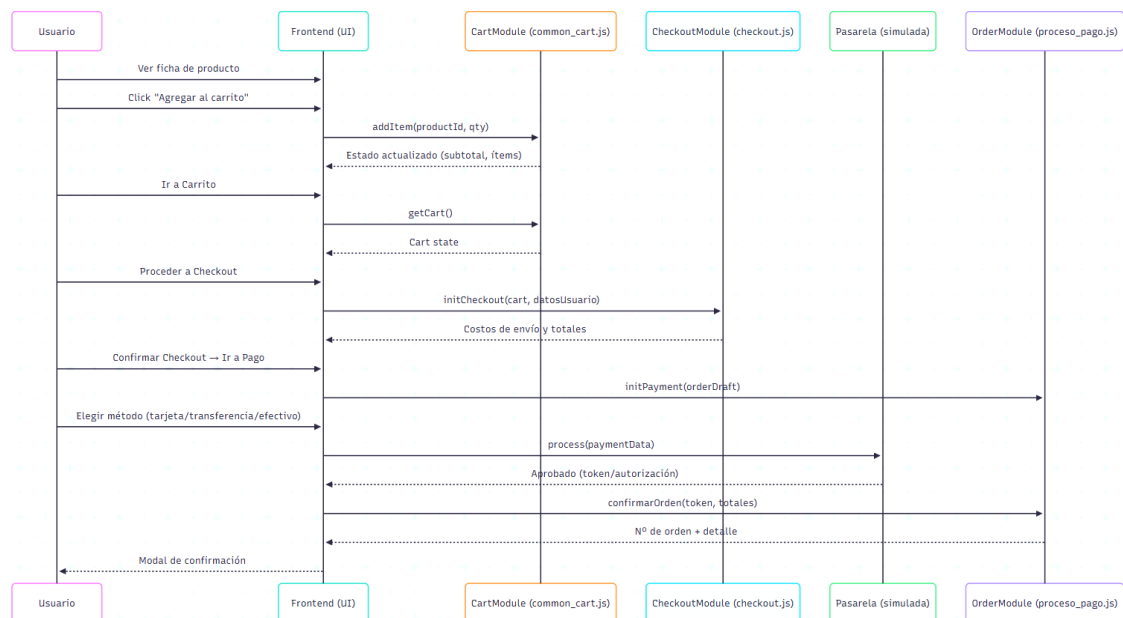


referencia normativa. La arquitectura prioriza claridad, mínima fricción y reutilización de plantillas y componentes.

Diagrama de Secuencia

Figura 8

Diagrama de secuencia: flujo de compra



Nota. Elaboración Propia.

El diagrama de secuencia muestra el recorrido desde la ficha hasta la confirmación: el usuario agrega un producto, la UI actualiza el carrito con *CartModule*, consulta el estado y avanza a checkout con *CheckoutModule*, que calcula envíos y totales. Luego se inicia el pago con *OrderModule*, el usuario elige el método y la pasarela simulada aprueba (token). Con esa

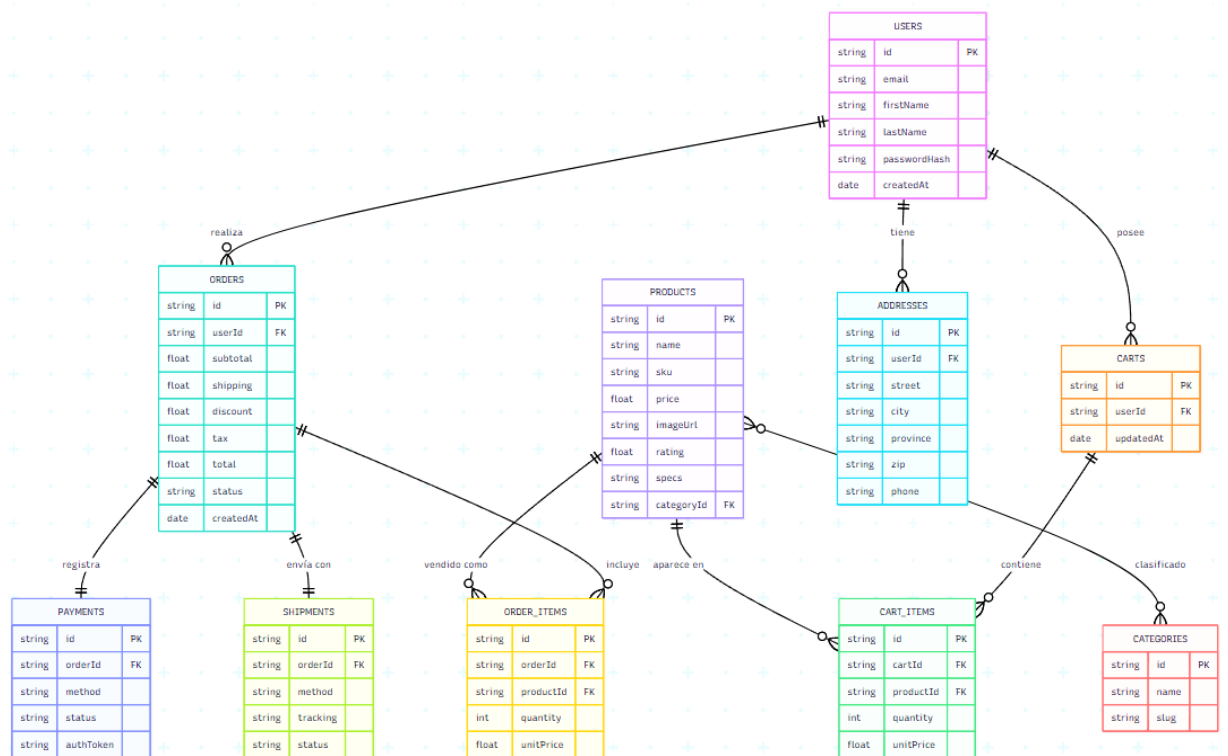


autorización, *OrderModule* confirma la orden y devuelve número y detalle; la UI muestra el modal de confirmación y cierra el flujo.

Modelo de datos (ER)

Figura 9

Modelo de datos (ER)



Nota. Elaboración Propia.

Este diagrama ER modela los datos clave del e-commerce y cómo se relacionan para sostener todo el flujo de compra. Define entidades, sus atributos principales y las cardinalidades: un usuario puede tener varios carritos y direcciones; un carrito contiene ítems

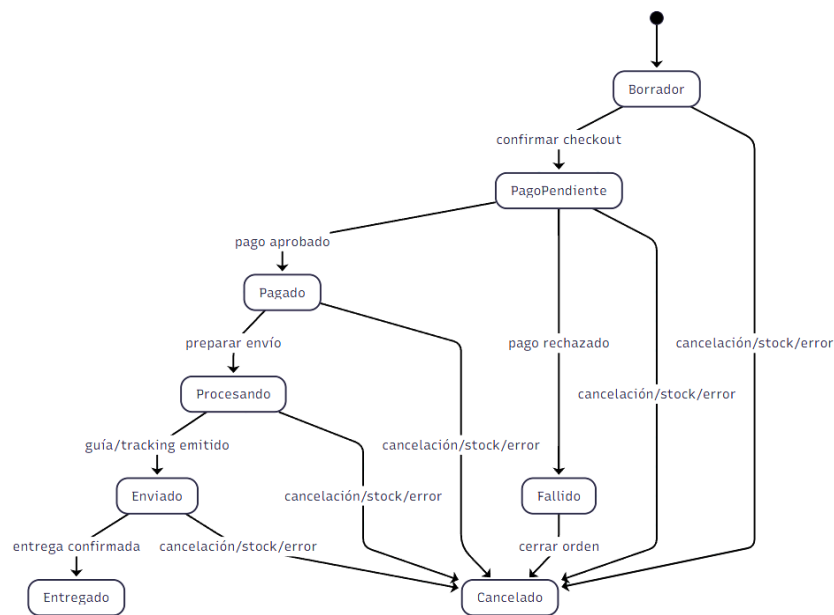


que referencian productos; una orden incluye ítems, registra un pago y un envío; los productos se clasifican por categorías y aparecen tanto en carritos como en órdenes. Las claves primarias (PK) y foráneas (FK) aseguran integridad referencial, lo que evita registros huérfanos y facilita consultas consistentes (totales, historial, stock proyectado).

Diagrama de máquina de estados del pedido

Figura 10

Máquina de estados del pedido



Nota. Elaboración Propia.

El diagrama representa la máquina de estados de una orden en Digital Point. La orden inicia en Borrador y, al confirmar el checkout, pasa a *PagoPendiente*. Si el cobro es aprobado por la pasarela, avanza a Pagado; si es rechazado, va a Fallido. Desde Pagado comienza la



preparación (Procesando), luego se despacha (Enviado) y finalmente se confirma la Entrega (Entregado).

Se contemplan transiciones hacia Cancelado por cancelación del cliente, falta de stock o errores operativos desde todos los estados relevantes, y se cierra explícitamente la orden Fallida llevándola a Cancelado.

Prototipo de diseño

Se desarrolló un prototipo completo en Figma que materializa visualmente todos los requerimientos funcionales y no funcionales establecidos. Este modelo representa la implementación visual de la estructura del sistema, desde la interfaz de usuario y navegación del catálogo hasta el proceso completo de checkout y secciones informativas corporativas.

Link de acceso al prototipo:

<https://www.figma.com/proto/UjSsWgPhU6mdR65ToAemVI/Untitled?node-id=1-5261&t=xa0of9l3hg7c0VXG-1>



Estructura de páginas desarrolladas

La capa servidor del proyecto está organizada en un conjunto de archivos PHP que sustituyen a las antiguas páginas estáticas y permiten integrar la base de datos. Las páginas públicas principales se implementan en archivos como `index.php`, `notebooks.php`, `monitores.php`, `ofertas.php` y `blog.php`, que construyen el HTML a partir de plantillas comunes e incorporan dinámicamente el catálogo, las ofertas y las entradas del blog. El flujo de compra se resuelve en `producto.php` (ficha detallada), `carrito.php`, `checkout.php` y `proceso_pago.php`, donde se recuperan productos, se calculan totales con IVA y descuentos, se validan los datos del usuario y se registran los pedidos. Las secciones institucionales y de soporte se materializan en `nosotros.php`, `preguntas.php`, `contacto.php`, `politica.php` y `terminos.php`, reutilizando la misma estructura de encabezado y pie. La lógica compartida se concentra en `includes/config.php`, que define la conexión MySQL y constantes del sistema, y en archivos de inclusión como `includes/header.php`, `includes/footer.php` y `includes/funciones.php`, que encapsulan componentes repetidos y utilidades. Sobre esta base se construye un panel de administración en archivos como `admin.php`, `admin_ofertas.php` y `admin_blog.php`, desde los cuales se gestionan ofertas, artículos, comentarios y "likes", con scripts auxiliares de alta, edición y baja que interactúan directamente con la base de datos respetando las reglas de negocio.

La capa de JavaScript mantiene la arquitectura modular definida en el prototipo original, ahora integrada con el backend PHP. El archivo `js/global.js` concentra utilidades transversales, formateo de precios, manejo de notificaciones y mejora de la experiencia de navegación, mientras que `js/components.js` se encarga de cargar de forma dinámica los



componentes de interfaz comunes, como encabezado y pie, asegurando coherencia entre todas las vistas. El estado comercial del sitio se administra desde módulos como `js/pages/common_cart.js` y `js/pages/productos_data.js`, que mantienen una base de datos local de productos y una API de carrito basada en `localStorage`, sincronizada con los totales que luego se persisten en MySQL al confirmar la compra. Cada vista relevante dispone de su propio script de página (por ejemplo, `js/pages/index.js`, `js/pages/monitores.js`, `js/pages/notebooks.js`, `js/pages/detalles_enrutamiento.js`, `js/pages/comprar.js`, `js/pages/checkout.js`, `js/pages/proceso_pago.js`, `js/pages/ofertas.js`, `js/pages/preguntas.js`, `js/pages/login.js` y `js/pages/nosotros.js`), que se encargan de los sliders, filtros, vistas rápida, validaciones en tiempo real, contadores regresivos y búsqueda, reforzando la interactividad del frontend sin recargar la página.

La hoja de estilos se organiza en una arquitectura CSS escalable con un núcleo en `css/global.css` y estilos específicos por página. En el archivo global se definen la paleta de colores del tema oscuro con acento rosa, el sistema de tipografías, espaciados, sombras y componentes base como tarjetas, botones, formularios, modales y avisos, junto con reglas de accesibilidad y puntos de quiebre responsivos. Las hojas de `css/pages/index.css`, `css/pages/monitores.css`, `css/pages/notebooks.css`, `css/pages/ficha_producto.css`, `css/pages/comprar.css`, `css/pages/checkout.css`, `css/pages/proceso_pago.css`, `css/pages/ofertas.css`, `css/pages/politica.css`, `css/pages/terminos.css`, `css/pages/preguntas.css` y `css/pages/login.css` ajustan la composición visual de cada sección (hero, grillas, filtros, acordeones, barras de progreso, formularios y layouts editoriales) sin romper la coherencia del sistema de diseño definido a nivel global.



En cuanto a la base de datos, el proyecto utiliza MySQL para persistir la información clave del e-commerce. El esquema contempla tablas para usuarios, productos, categorías, ofertas, pedidos y sus detalles de línea, así como estructuras auxiliares para gestionar cupones, reseñas, entradas del blog, comentarios y reacciones. Las relaciones entre tablas vinculan cada pedido con su usuario, sus productos y los importes calculados en el frontend, permiten activar o desactivar ofertas sin perder historial y centralizan el catálogo que se expone en las páginas PHP. Esta estructura relacional respalda el flujo completo de navegación, compra y administración desde el panel de backend.

Las tecnologías empleadas combinan HTML5 semántico, CSS3 responsivo y JavaScript vanilla modular con PHP y MySQL sobre servidor Apache (entorno XAMPP en desarrollo y hosting con dominio propio en producción). El frontend mantiene un enfoque mobile-first con uso de Flexbox y CSS Grid, transiciones suaves y buenas prácticas de accesibilidad, mientras que el backend en PHP aporta renderizado dinámico, acceso a datos y formularios seguros para administrar contenido. Sobre este núcleo se apoya un prototipo de diseño elaborado en Figma, que define la identidad visual, los componentes y los flujos de interacción y que se respeta en la implementación real. El proyecto Digital Point, accesible mediante el dominio registrado digital-point.shop, se presenta así como un e-commerce especializado en notebooks y monitores, con un prototipo de diseño consolidado y una implementación técnico-funcional que cubre catálogo, ofertas, blog, flujo de compra completo y panel de administración, listo para ser extendido con servicios de autenticación y pagos reales en una siguiente etapa evolutiva.



Repositorio del Proyecto

El código fuente de Digital Point se mantiene versionado en el repositorio GitHub, asegurando trazabilidad completa, control de versiones y documentación del desarrollo.

Repositorio de código fuente: https://github.com/RominaaVeraa/TIF_VeraRomina.git



Conclusión

Como resultado, Digital Point se consolida como un prototipo avanzado de tienda online, coherente con el diseño planteado en Figma e implementado con tecnologías web ampliamente utilizadas en entornos profesionales. El sitio integra navegación completa, interactividad en tiempo real y persistencia de información, acompañado por un panel de administración que facilita la gestión de ofertas y contenidos. El proyecto demuestra la viabilidad técnica del concepto, deja sentada una base sólida para incorporar métodos de pago reales y autenticación robusta en futuras iteraciones, y evidencia la integración efectiva de diseño, programación y base de datos en una misma solución.



Bibliografía

Flaticon. (s. f.). *Uicons: Interface icons*. Flaticon.

<https://www.flaticon.es/uicons/interface-icons>

Fernández, J. A. C. (s. f.). *Guía de referencia de HTML y CSS*. Universidad de Valencia. <https://www.uv.es/jac/guia/index.html>

Lenguaje HTML. (s. f.). *Lenguaje HTML*. <https://lenguajehtml.com/html/>

MDN Web Docs. (s. f.). *Funciones — Guía de JavaScript*. MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Functions>

Flaticon. (s. f.). *Entrega urgente — icono gratuito*. Flaticon. https://www.flaticon.es/icono-gratis/entrega-urgente_3142292