

Trabajo Práctico N°2

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: 07-10-2025.



Introducción

Digital Point es un e-commerce especializado en notebooks y monitores desarrollado como prototipo avanzado plenamente funcional. La solución implementa HTML5 semántico, CSS3 responsivo con Grid, Flexbox y sistema de diseño basado en custom properties (tema oscuro con acento rosa #e8c5d8), y JavaScript vanilla modular para interactividad, validaciones y persistencia mediante localStorage/sessionStorage.

El proyecto cubre extremo a extremo el recorrido del usuario: portada con slider y destacados, catálogo con filtrado dinámico, fichas de producto con galería interactiva, carrito con cupones y cálculo automático de IVA (21%) y envío, checkout con validación en vivo, proceso de pago con tres métodos, sistema de autenticación con perfil y historial de pedidos, además de secciones institucionales, ofertas con contador regresivo, FAQs con buscador, política de privacidad con configuración de cookies y términos legales.

El diseño adopta enfoque mobile-first e incorpora prácticas de accesibilidad mediante estructura jerárquica, atributos ARIA y navegación por teclado. El sitio está desplegado en Netlify (<https://digital-point.netlify.app/>) con dominio propio registrado (digital-point.shop), preparado para HTTPS y CDN sin cambios de código.

El objetivo principal es validar navegación, contenidos y diseño con usuarios finales en entorno equivalente a producción, estableciendo bases técnicas y contratos de interfaz para futura integración backend con servicios reales de autenticación, catálogo, stock y pagos.



Desarrollo

Descripción del proyecto

Digital Point constituye una tienda virtual moderna y responsiva que presenta una estructura organizacional clara. La plataforma cuenta con una franja superior informativa que incluye horarios de atención, dirección, contacto y teléfono. La estructura principal incorpora el logotipo corporativo, categorías de productos diferenciadas (notebooks y monitores), sección de ofertas especiales, buscador avanzado, carrito de compras y área de perfil de usuario.

El sistema permite la navegación intuitiva a través de categorías específicas como notebooks para uso diario y gaming, así como monitores de oficina y gaming. La implementación incluye funcionalidades avanzadas de búsqueda con filtros por precio, relevancia y categoría, además de opciones de visualización alternativas.

Sección 1: Resumen de Trabajo Práctico N°1

En el primer trabajo práctico se desarrolló Digital Point, una plataforma de comercio electrónico especializada en notebooks y monitores. El proyecto inició con análisis de requerimientos diferenciando usuarios registrados con acceso completo de usuarios no registrados limitados a visualización y suscripción, estableciendo tiempos de respuesta inferiores a tres segundos y diseño responsivo.

El modelado empleó diagramas UML incluyendo flujo de datos, casos de uso diferenciando funcionalidades, y diagrama de clases modelando estructura completa con atributos, métodos y relaciones entre entidades.



La implementación visual inició con prototipo en Figma definiendo paleta de colores con tema oscuro y acentos rosa, tipografías e iconografía. El frontend desarrolló cinco páginas usando HTML5 y CSS3. La portada incluye sección hero, categorías, galería con calificaciones, testimonios y formulario de suscripción. El listado en tabla usa flexbox mostrando productos horizontalmente con controles de ordenamiento. El listado en cajas implementa CSS Grid responsivo con tarjetas verticales. La ficha de producto presenta layout de dos columnas con galería interactiva e información detallada. El formulario de compra separa productos del resumen con totales. El proceso de pago incluye formulario de datos y opciones de envío junto a resumen del pedido.

Las tecnologías empleadas fueron HTML5 para estructura semántica, CSS3 con Grid, Flexbox, custom properties y transiciones. El proyecto constituyó un prototipo enfocado en diseño visual y experiencia de usuario con funciones backend simuladas, estableciendo bases para futura implementación completa.

Objetivos del Sistema

Objetivo General: Desarrollar una plataforma de comercio electrónico especializada en notebooks y monitores que proporcione a los usuarios una experiencia de compra sencilla, segura y eficiente.

Objetivos Generales:

- Implementar una interfaz intuitiva con categorización jerárquica de productos.
- Desarrollar un sistema de búsqueda eficiente con filtros avanzados por precio, relevancia y categoría.



- Incorporar funcionalidades completas del carrito de compras con gestión de cantidades y cálculo automático.
- Desarrollar un sistema de checkout con cálculo automático de costos de envío
- Implementar sistema de reseñas de clientes para cada producto.
- Crear panel de control personalizado para gestión de cuenta del usuario.
- Desarrollar sistema de suscripción para notificaciones de ofertas especiales.

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:

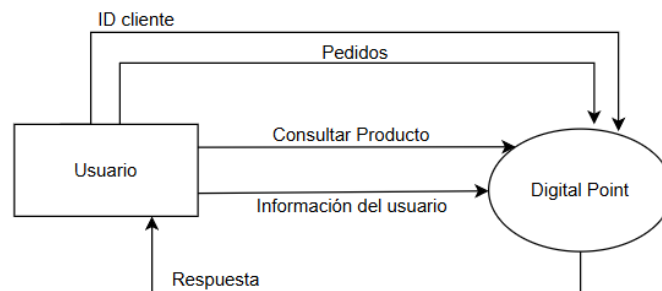
- Interfaz intuitiva adaptable a usuarios de diferentes niveles técnicos
- Diseño responsivo para múltiples dispositivos.
- Navegación clara y consistente en todas las secciones.

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:



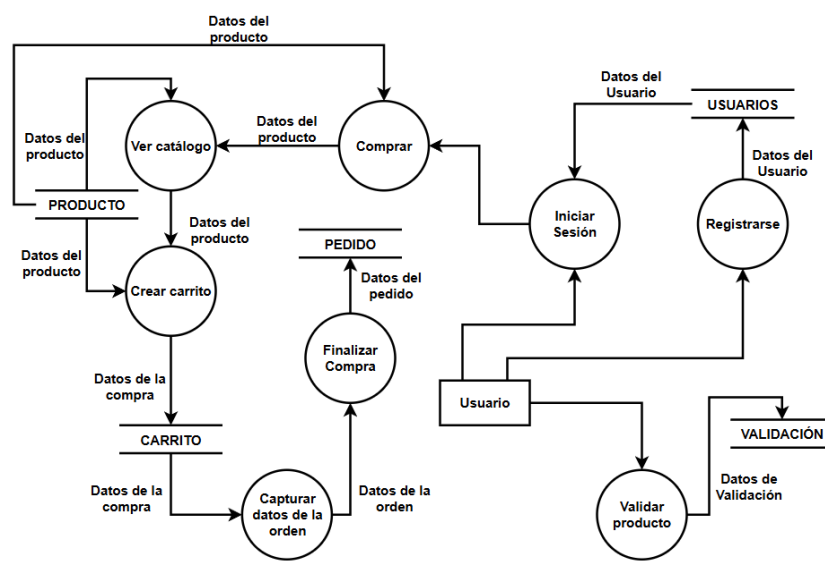
- Información del usuario: datos de registro, inicio de sesión o perfil.
- Consultar producto: búsquedas o solicitudes para visualizar el catálogo.
- Pedidos: solicitudes de compra con productos seleccionados.
- Salidas hacia el usuario:
 - ID del cliente: confirmación de registro o autenticación.
 - 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.

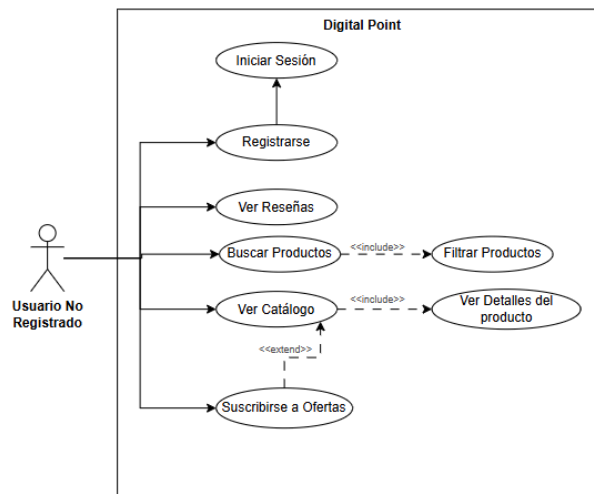
7. Capturar datos de la orden.

- 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.

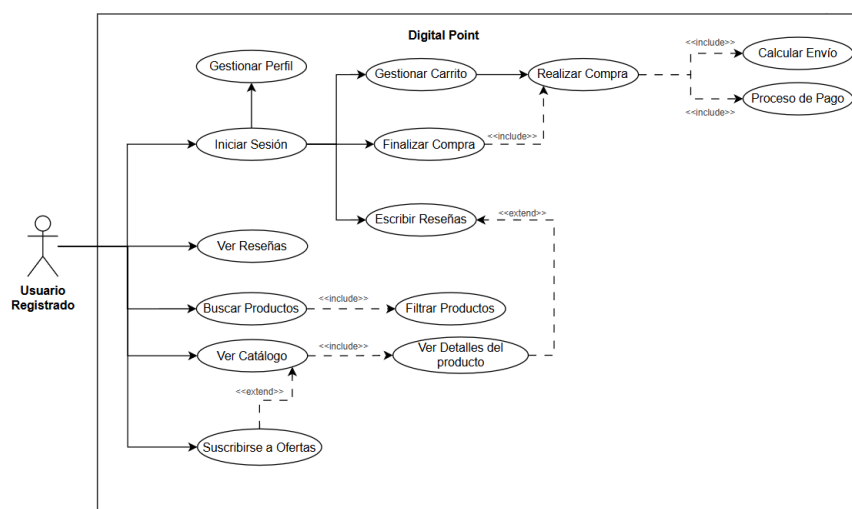


3	El usuario puede buscar productos y filtrarlos.	4	El usuario puede acceder al catálogo y ver detalles de los productos.
5	El usuario, si desea, puede suscribirse a ofertas ingresando su correo.	6	El usuario decide registrarse y crear una cuenta.
7	El usuario queda habilitado para iniciar sesión.		

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.



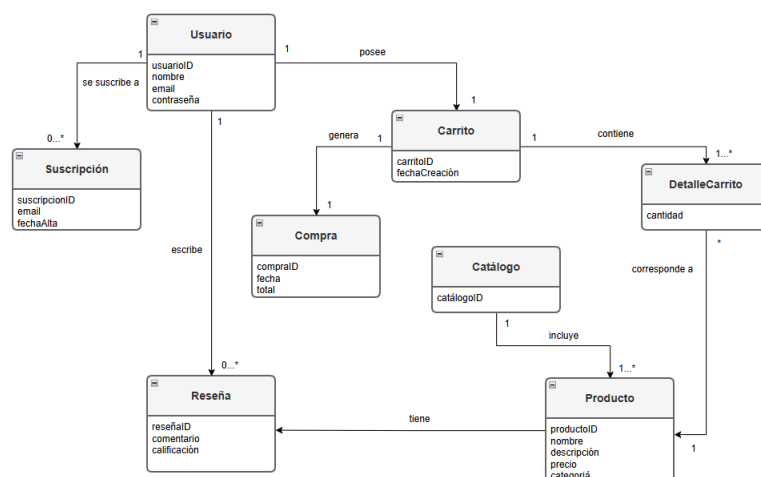
Curso Normal.

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 según sus intereses.	4	El usuario visualiza detalles de los 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.

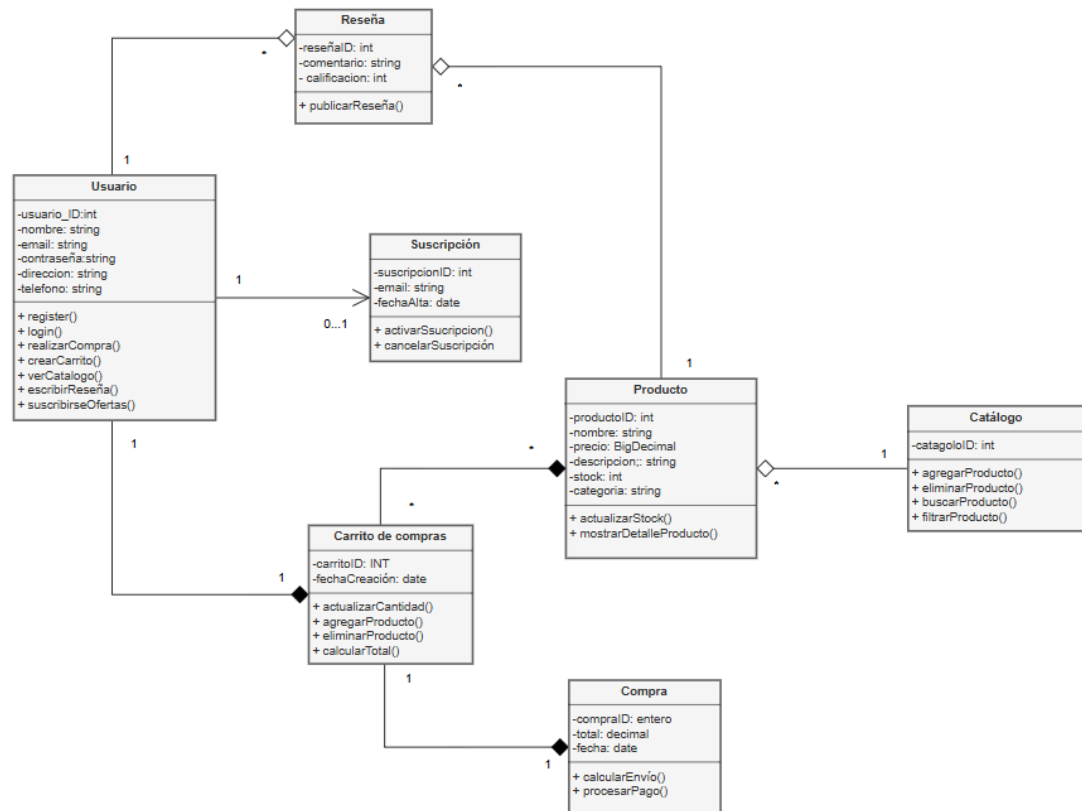
Catálogo: Agrupa y organiza productos de la tienda en categorías jerárquicas. Facilita navegación y búsqueda de productos.



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.

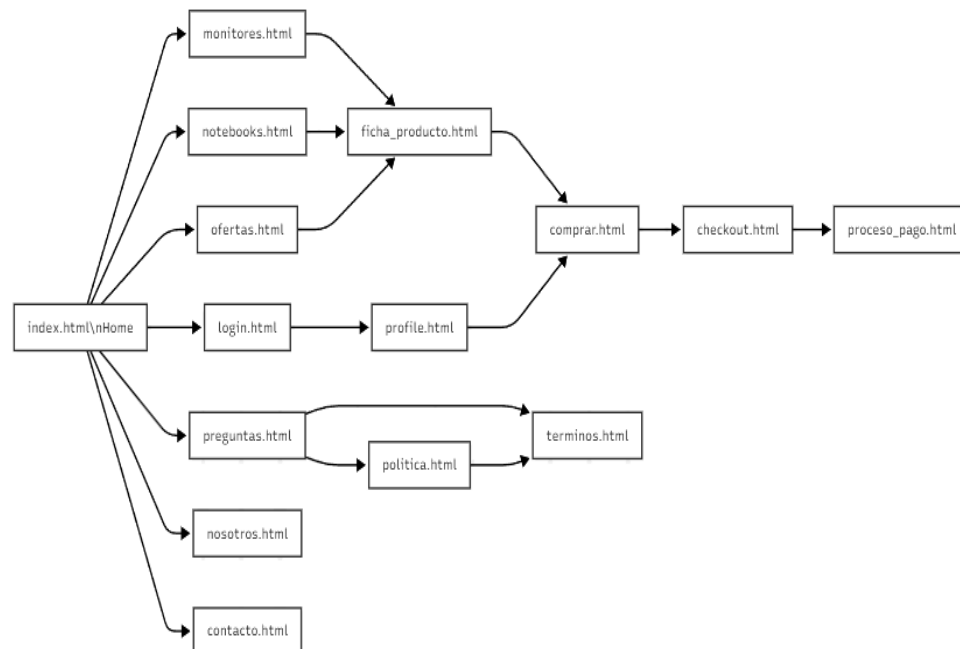


- **Reseña:** almacena comentarios y calificaciones de los usuarios sobre los productos.
 - Está en relación de agregación con Usuario (un usuario puede crear reseñas).
 - También se relaciona con Producto (un producto puede tener varias reseñas).

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,

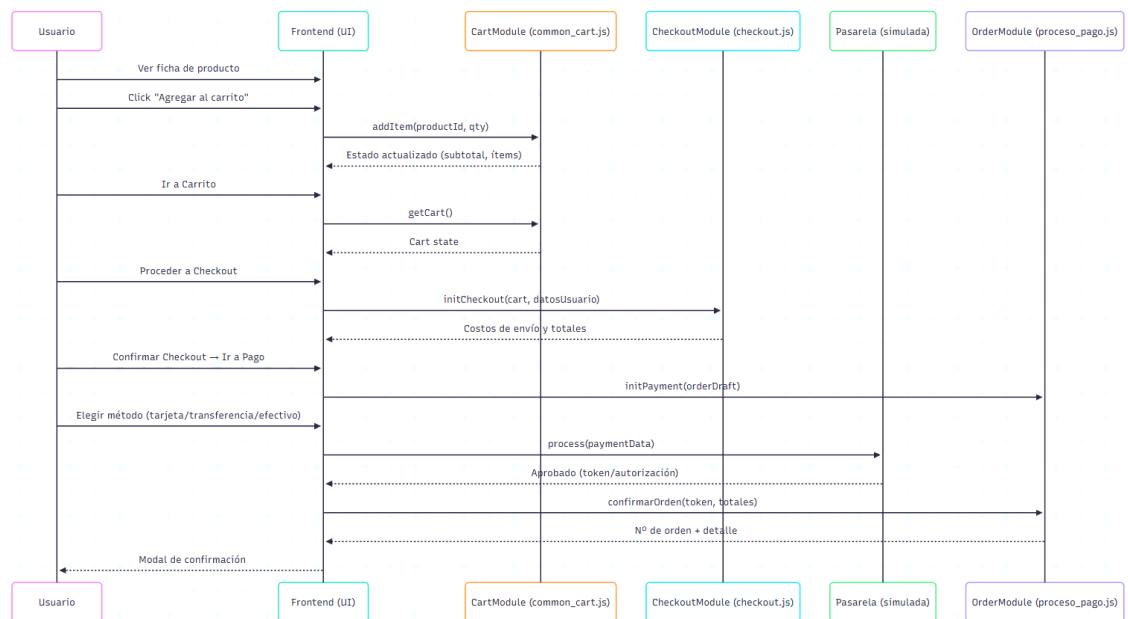


con retorno posible al Carrito para continuar la compra. El bloque de soporte y legales parte de Preguntas y enlaza a Política y Términos (y de Política a Términos), concentrando la referencia normativa. La arquitectura prioriza claridad, mínima fricción y reutilización de plantillas y componentes.

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

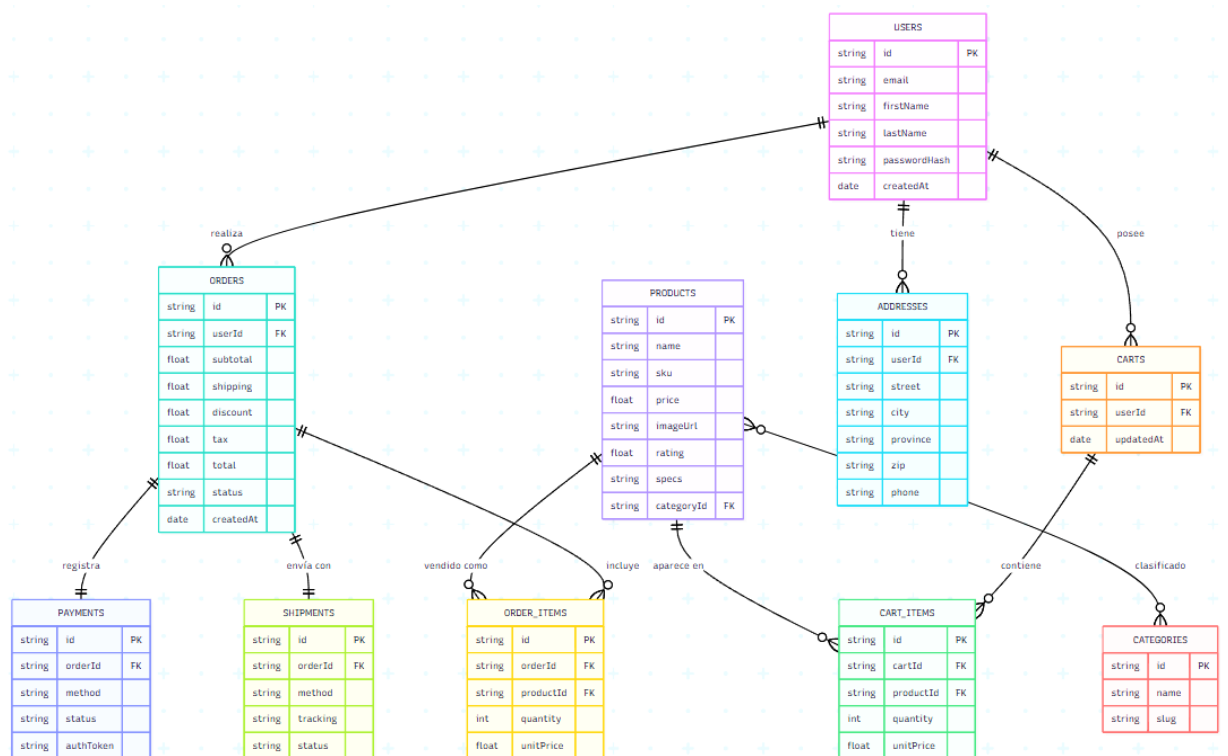


con *OrderModule*, el usuario elige el método y la pasarela simulada aprueba (token). Con esa autorización, *OrderModule* confirma la orden y devuelve número y detalle; la UI muestra el modal de confirmación y cierra el flujo.

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

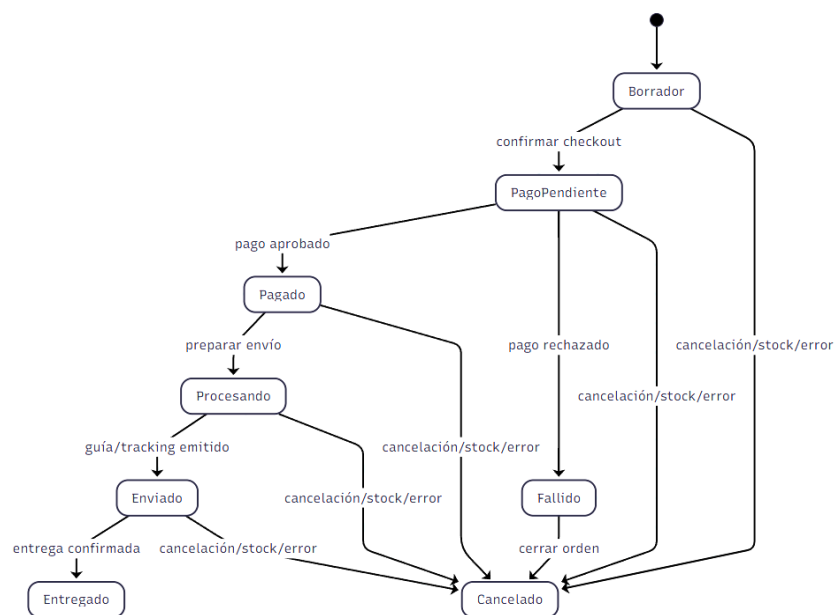


cardinalidades: un usuario puede tener varios carritos y direcciones; un carrito contiene ítems que referencian productos; una orden incluye ítems, registra un pago y un envío; los productos se clasifican por categorías y aparecen tanto en carritos como en órdenes. Las claves primarias (PK) y foráneas (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.

El prototipo incluye definición de paleta de colores, tipografías, iconografía, layouts responsivos y flujos de interacción usuario-sistema. Constituye la base visual para el desarrollo frontend y garantiza consistencia en la experiencia de usuario a través de todas las funcionalidades de la plataforma.

Link de acceso al prototipo:

https://www.figma.com/proto/UjSsWgPhU6mdR65ToAemVI/Untitled?node-id=1-5261&t=x_a0of9l3hg7c0VXG-1



Sección 2: Detalles de mejoras

Estructura de páginas desarrolladas

La implementación frontend reúne páginas coherentes con estética uniforme, componentes reutilizables y módulos JavaScript comunes. La *portada* presenta hero full-screen con slider, accesos a categorías, grilla de destacados con precio y valoración, beneficios, testimonios con carrusel y newsletter con validación.

La *ficha de producto* organiza una galería con miniaturas e información con título, precio, valoración, especificaciones, selector de cantidad y acción de compra. *Contacto* combina hero con datos institucionales, formulario con validaciones y FAQs en acordeones.

El *carrito* incorpora breadcrumbs, estados vacío/con ítems, lista de productos, resumen dinámico con subtotal, envío, descuento, IVA y total, aplicación de cupones y avance a checkout. El *checkout* consolida datos de contacto y envío con validación y métodos logísticos impactando el cálculo final.

El *listado* dispone catálogo con breadcrumb, filtros por segmento y uso, contador de resultados y ordenación por nombre, precio o valoración. *Autenticación* presenta flujos de ingreso y registro con validaciones y alternador de visibilidad de contraseña.

Monitores y notebooks abren con hero temático, filtros por uso, selector de orden, conmutador de vista, tarjetas con imagen, badges, especificaciones, valoración, precio, agregado al carrito y vista rápida en modal. La *página institucional* articula historia, compromiso y garantías mediante bloques narrativos con métricas. *Ofertas* centra atención en campañas con hero de impacto, contador regresivo y catálogo con etiquetas de descuento filtrable.



Política de privacidad expone con índice lateral propósito, principios de datos, compartición, cookies, seguridad y derechos. *Preguntas frecuentes* actúan como hub con buscador, categorías con contador y acordeones.

Proceso de pago cierra transacción con barra de progreso, columnas para métodos y resumen, formularios con máscaras, datos bancarios con copiado rápido, cálculo en tiempo real y modal con número de orden. *Perfil* centraliza datos personales, seguridad y pedidos en pestañas con validaciones. *Términos* formalizan marco contractual con índice navegable y secciones sobre aceptación, uso, compras, garantías y contacto legal.

En conjunto, el frontend ofrece una experiencia consistente, accesible y orientada a la conversión a lo largo del recorrido del usuario.

Detalle de las mejoras incorporadas mediante JavaScript

La incorporación de JavaScript transformó el sitio de prototipo estático a aplicación web interactiva mediante módulos transversales y scripts especializados.

Arquitectura y componentes: *js/global.js* implementó utilidades de formateo, sistema toast de notificaciones, manejo de scroll y listeners globales. *js/components.js* habilitó carga dinámica de header y footer mediante fetch API con función loadComponent, sincronizando navegación y contadores automáticamente.

Gestión comercial: *js/pages/common_cart.js* creó API unificada CartAPI con localStorage normalizando precios y cantidades, calculando totales en tiempo real y publicando eventos cart:updated. *js/pages/productos_data.js* estableció base de datos local PRODUCTOS_DB con 20 productos incluyendo especificaciones, imágenes múltiples, valoraciones y stock.



Páginas principales: *js/pages/index.js* agregó slider automático del hero cada 2 segundos, renderizado dinámico de destacados, validación de newsletter con FormSubmit y carrusel de testimonios. *js/pages/monitores.js* y *notebooks.js* implementaron filtrado dinámico por categorías con contador, alternancia grilla/lista, ordenación múltiple, modal de vista rápida con teclado, y navegación mediante clic en título o imagen redirigiendo a ficha. *js/pages/detalles_enrutamiento.js* habilitó carga mediante ?id=, galería interactiva con slider pausable, pestañas de contenido, selector de cantidad limitado y productos relacionados dinámicos.

Transacciones: *js/pages/comprar.js* implementó carrito con controles de cantidad, cupones de descuento predefinidos, recálculo automático con IVA 21% y envío gratis sobre \$500.000, y productos relacionados sugeridos. *js/pages/checkout.js* añadió precarga de datos del usuario, validación en vivo de campos, selección de envío actualizando costos automáticamente. *js/pages/proceso_pago.js* completó con tres métodos de pago, máscaras automáticas, copiado rápido, generación de orden única, modal de confirmación, guardado en historial del usuario consultable desde perfil y limpieza post-compra.

Funcionalidades adicionales: *js/pages/ofertas.js* implementó contador regresivo en tiempo real y slider de fondo rotando cada 2 segundos. *js/pages/politica.js* construyó índice con scroll spy, modal de cookies con persistencia y cálculo de tiempo de lectura. *js/pages/preguntas.js* añadió buscador con sugerencias, filtro por categoría con contador, acordeones accesibles ARIA y resaltado de términos. *js/pages/contacto.js* implementó slider de fondo, validación con contador de caracteres, integración FormSubmit y efecto confetti. *js/pages/nosotros.js* agregó contadores animados y parallax.



Autenticación: *js/pages/login.js* implementó alternancia entre formularios, validación en tiempo real, confirmación de contraseña instantánea, alternador de visibilidad, sesiones con *sessionStorage*, edición de perfil con validación, historial completo de pedidos consultable y cierre de sesión con limpieza.

JavaScript transformó el sitio en aplicación completa con gestión de estado mediante *localStorage/sessionStorage*, componentes dinámicos cargados con *fetch*, persistencia entre sesiones y experiencia comparable a e-commerce profesionales con funcionalidad completa de compra, autenticación y procesamiento de pedidos.

Arquitectura CSS

La base tipográfica, sistema de color, espaciados, sombras, normalización y utilidades de layout viven en *css/global.css*, estableciendo tokens reutilizables para tarjetas, botones, formularios, modales y avisos, con reglas de accesibilidad *:focus-visible*, contrastes y breakpoints para responsividad. Las hojas específicas en *css/pages/** agregan composición propia sin romper coherencia del sistema.

css/pages/index.css alinea hero full-screen, carrusel, tarjetas de destacados y bloques de beneficios con transiciones suaves. *css/pages/monitores.css* y *notebooks.css* estructuran filtros, conmutadores de vista, grillas, badges, tarjetas con especificaciones y modal de vista rápida para dispositivos táctiles. *css/pages/ficha_producto.css* articula grilla de dos columnas para galería y detalle, escala tipográfica, controles de cantidad y pestañas con estados activos. *css/pages/comprar.css*, *checkout.css* y *proceso_pago.css* ordenan pasos del embudo con barras de progreso, paneles, formularios, resúmenes y validaciones. *css/pages/ofertas.css* enfatiza hero de alto contraste, distintivos de campaña, contador y tarjetas con doble precio.



css/pages/politica.css y *terminos.css* trabajan layout editorial con índice pegajoso, anclajes y bloques de derechos manteniendo ritmo de lectura. *css/pages/preguntas.css* prioriza buscador, acordeones y estados vacíos, mientras *login.css* y *profile.css* garantizan formularios ordenados con manejo visual de errores y éxito. La arquitectura conserva nomenclatura estable, patrones repetibles y responsividad coherente permitiendo integración sin fricción.

Tecnologías utilizadas actualmente

Digital Point está construido con HTML5 semántico usando `<header>`, `<nav>`, `<main>` y `<section>`, CSS3 estructurado en *css/global.css* y hojas por página, y JavaScript vanilla modularizado en *js/global.js*, *js/components.js* y *scripts* de vista. El layout combina CSS Grid para estructuras responsivas y Flexbox para alineación, con sistema de diseño basado en custom properties definiendo paleta y espaciados, transiciones suaves y pseudo-elementos para micro-interacciones. El enfoque es mobile-first con media queries progresivas, tipografías del sistema para rendimiento y patrones de accesibilidad integrados. La interactividad se resuelve sin frameworks: sliders, vista rápida, enrutamiento simple por querystring, validaciones de formularios y manejo de estado con localStorage para carrito y preferencias, más componentes compartidos de header y footer inyectados por JavaScript.

Sección 3: 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.

El proyecto está construido con HTML5 semántico, CSS3 responsive utilizando Grid y Flexbox para adaptabilidad multidispositivo, y JavaScript vanilla modular que gestiona



navegación, validaciones de formularios, cambio dinámico de vistas, cálculo automático de totales con impuestos y envío, manejo integral del carrito y persistencia mediante localStorage.

La arquitectura implementa un enfoque frontend-first donde el comportamiento es operativo extremo a extremo: el flujo completo de compra, gestión de usuarios y resúmenes de pedidos funcionan íntegramente en el navegador sin dependencias de servidor. Esta estructura permite evaluar la experiencia de usuario real y establecer contratos de interfaz para futura integración con APIs backend cuando se requiera escalabilidad.

Repositorio de código fuente: https://github.com/RominaaVeraa/VeraRomina_TP2.git

Sección 4: Despliegue y Hosting del Proyecto

Digital Point está desplegado y operativo en dos plataformas, ofreciendo acceso completo a todas las funcionalidades desarrolladas.

Sitio desplegado en Netlify: <https://digital-point.netlify.app/>

Netlify funciona como plataforma de despliegue principal para demostración del proyecto, donde el sitio opera con plena funcionalidad en todos los dispositivos y resoluciones. El desarrollo ha sido validado exitosamente en múltiples entornos incluyendo servidor local y Netlify, garantizando correcta visualización y operación de todas las secciones: portada, catálogo de productos, fichas detalladas, carrito de compras, proceso de checkout y confirmación de pedidos, además de secciones institucionales, ofertas especiales, FAQs, política de privacidad, términos y condiciones, y sistema de autenticación de usuarios.



Dominio propio registrado: <http://digital-point.shop>

El proyecto cuenta con dominio propio registrado y activo con hosting contratado, actualmente en proceso de configuración. La arquitectura implementada permite incorporar certificados SSL para HTTPS y servicios de CDN cuando sea necesario para optimizar tiempos de carga.

La experiencia de usuario mantiene identidad visual consistente mediante paleta de colores definida (tema oscuro con acento rosa #e8c5d8), tipografía coherente y micro-interacciones, priorizando accesibilidad, rendimiento óptimo con tiempos de carga inferiores a 4 segundos, y fluidez en móviles y escritorio.



Conclusión

Digital Point consolida una base estable, coherente y escalable para un e-commerce moderno. El sitio ofrece experiencia completa de exploración y compra con 20 productos cargados desde base de datos local PRODUCTOS_DB, mantiene normas de accesibilidad y rendimiento razonables con tiempos de carga inferiores a 4 segundos, y está publicado en dominio propio con hosting operativo habilitando demostraciones y pruebas en condiciones reales.

La arquitectura frontend resuelve presentación, interacción y feedback mediante componentes reutilizables cargados dinámicamente con fetch API, gestión unificada de carrito con CartAPI, eventos personalizados cart:updated sincronizando estado entre páginas, validaciones en tiempo real con feedback visual inmediato, y persistencia entre sesiones. Los puntos de extensión están claramente definidos para incorporar backend productivo sin reescrituras de interfaz.

La transición a producto final requiere conectar servicios de autenticación OAuth, exponer APIs REST de catálogo con stock y precios en tiempo real, integrar pasarela de pago con manejo de estados e idempotencia, endurecer seguridad con cabeceras CSP y protección CSRF, instrumentar métricas con Google Analytics y trazabilidad con Sentry, formalizar pruebas automatizadas con Jest y Cypress, y establecer pipeline CI/CD con GitHub Actions para despliegue continuo bajo HTTPS con certificado SSL y monitoreo de performance.

Con esos incrementos, el prototipo avanzado se convierte en aplicación lista para producción, preservando la experiencia validada y capitalizando la inversión en diseño, componentes modulares y arquitectura CSS/JavaScript que hoy opera en digital-point.shop.



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/>

MSI. (s. f.). *Cyborg 15 A2RX*. MSI. <https://www.msi.com/Laptop/Cyborg-15-A2RX>

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

MSI. (s. f.). *Monitores: Serie MAG — Productos*. MSI. <https://www.msi.com/Monitors/Products#?tag=MAG-Series>

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

Lenguaje JS. (s. f.). *Callbacks en JavaScript*. Lenguaje JS. <https://lenguajejs.com/javascript/funciones/callbacks/>