

## **Paradigmas y Lenguajes de Programación III**

### **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:

Salazar, Facundo.

FECHA: 07-10-2025.



---

## INTRODUCCIÓN

El presente informe documenta el desarrollo del Trabajo Práctico N°2 de la materia Paradigmas y Lenguajes de Programación III, el cual representa una evolución significativa del proyecto Synk iniciado en el TP1. Mientras que la primera fase del proyecto se enfocó en establecer la estructura base mediante HTML5 y CSS3, esta segunda etapa transforma la plataforma estática en una aplicación web completamente funcional e interactiva mediante la incorporación de JavaScript.

La consigna del TP2 requería la creación de un sitio totalmente funcional y adaptativo, haciendo especial énfasis en el uso de JavaScript y diseño responsive. Para cumplir con estos objetivos, se implementaron múltiples funcionalidades que abarcan desde sistemas de autenticación de usuarios hasta formularios inteligentes con validación en tiempo real, pasando por carritos de compra dinámicos, carruseles de imágenes interactivos y menús de navegación responsive con accesibilidad completa.

El desarrollo de este trabajo práctico implicó no solo la programación de funcionalidades visibles para el usuario final, sino también la integración de servicios externos como Formspree y EmailJS para gestión de suscripciones, la implementación de buenas prácticas de accesibilidad mediante atributos ARIA, y la optimización del rendimiento mediante técnicas como debounce y gestión eficiente de eventos. Cada funcionalidad fue diseñada pensando en la experiencia de usuario, la escalabilidad del código y la preparación del sistema para futuras ampliaciones que podrían incluir integración con backend, bases de datos y sistemas de pago reales.



---

A lo largo de este informe se detallan las mejoras implementadas, organizadas por funcionalidad, explicando su propósito, características técnicas y beneficios para el usuario final. Asimismo, se documentan las decisiones técnicas tomadas durante el desarrollo y se identifican oportunidades de mejora para futuras iteraciones del proyecto.



---

## DESARROLLO

Synk es una plataforma web de comercio electrónico orientada a la venta de productos electrónicos y electrodomésticos, como smartphones, notebooks, consolas, televisores, auriculares y accesorios de última generación. El proyecto surge con el propósito de crear un espacio digital moderno que combine facilidad de uso, diseño atractivo y funcionalidad, ofreciendo a los usuarios una experiencia de navegación inspirada en la estética hype/tech, característica de las tendencias actuales en tecnología y diseño web.

En el Trabajo Práctico N°1 se desarrolló la estructura base del sitio web utilizando HTML5 semántico y CSS modular, sentando las bases para una plataforma de e-commerce funcional y escalable.

### Componentes desarrollados en el TP1

***Página principal (index.html):*** Sección hero con presentación de la marca, información corporativa sobre ventajas competitivas (envíos, seguridad, innovación) y footer con datos de contacto.

#### **Catálogo de productos en dos formatos:**

***Vista en grid (listado\_box.html):*** Presentación visual mediante tarjetas (cards) con imagen, nombre, precio y etiquetas de estado (Nuevo, Oferta, Envío Gratis, Agotado).

***Vista en tabla (listado\_tabla.html):*** Listado estructurado con columnas de imagen, producto, categoría, precio, stock y enlace a ficha técnica.

***Páginas de detalle de producto:*** Seis fichas individuales (celular, auriculares, notebook, monitor, teclado, mouse) con carrusel de imágenes estáticas (4 imágenes por producto), descripción, especificaciones técnicas y botones de acción.



---

**Formulario de compra (comprar.html):** Estructura básica para recolectar datos del cliente (nombre, email, teléfono, dirección), selección de método de pago y listado estático de productos disponibles para comprar.

**Navegación coherente:** Menú de navegación presente en todas las páginas, asegurando una experiencia de usuario fluida y consistente.

**Diseño responsive:** Implementación de media queries y técnicas de CSS para adaptar el contenido a dispositivos móviles, tablets y desktop.

### **Objetivos generales del TP1**

Desarrollar una plataforma web de comercio electrónico que permita a los usuarios explorar un catálogo de productos tecnológicos y simular el proceso de compra de manera sencilla, segura y atractiva, priorizando la experiencia de usuario y la navegación intuitiva mediante el uso de tecnologías web estándar (HTML5 y CSS3).

### **Resultados Obtenidos en el TP1**

El desarrollo del sitio Synk en su primera fase permitió cumplir satisfactoriamente con los objetivos planteados:

**Navegación clara y consistente:** El menú de navegación facilita el acceso a todas las secciones del sitio desde cualquier página.

**Catálogo organizado:** Las vistas en formato grid y tabla permiten al usuario elegir la presentación que mejor se adapte a su preferencia de navegación.

**Estructura semántica:** Se respetaron las buenas prácticas de codificación, utilizando HTML5 semántico y CSS modular para favorecer el mantenimiento y futuras ampliaciones.

**Diseño responsive funcional:** El sitio se adapta correctamente a dispositivos móviles, tablets y desktop, garantizando una experiencia de usuario consistente.

---



---

**Base sólida para el TP2:** La estructura desarrollada permitió la incorporación posterior de funcionalidades dinámicas mediante JavaScript, tal como se detalla en la siguiente sección del presente informe.

### **Evolución hacia el TP2**

En este trabajo comenzamos realizando los diagramas fundamentales para describir el funcionamiento y la estructura de Synk, una plataforma web de comercio electrónico orientada a productos tecnológicos. Los artefactos elaborados —Diagrama de Clases, Diagrama Entidad–Relación (ER) y Diagramas de Secuencia— permiten comprender, respectivamente, la arquitectura lógica del dominio, el modelo relacional que garantiza la integridad de datos y los flujos dinámicos de interacción durante los procesos clave (compra y autenticación).

#### **Diagrama de clases**

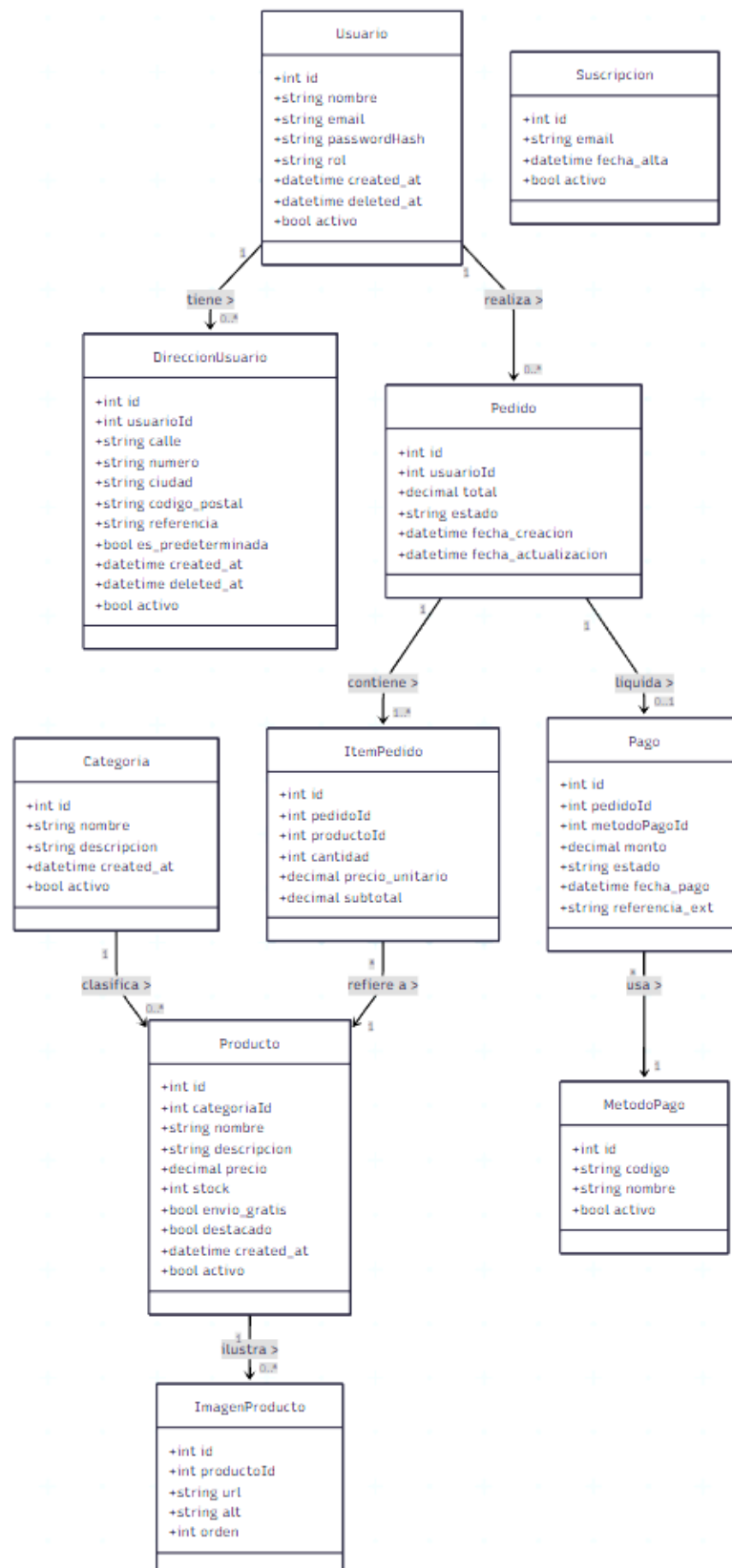
Modela los conceptos del dominio y sus relaciones principales.

- Usuario (1) — DirecciónUsuario (N): un usuario puede registrar múltiples direcciones y marcar una como predeterminada.
- Categoría (1) — Producto (N): cada producto se clasifica en una sola categoría.
- Producto (1) — ImagenProducto (N): múltiples imágenes ordenadas por producto para el carrusel.



- 
- Usuario (1) — Pedido (N) — ItemPedido (N): el pedido agrupa ítems; cada ítem referencia producto, cantidad y precio unitario (conserva histórico de precio).
  - Pedido (1) — Pago (0..1) — MétodoPago (1): se registra la liquidación y su estado; el método de pago queda parametrizado.
  - Suscripción: almacena emails del newsletter, desacoplada del registro de usuarios.

El diseño favorece cohesión, bajo acoplamiento y trazabilidad (created\_at, deleted\_at/activo).







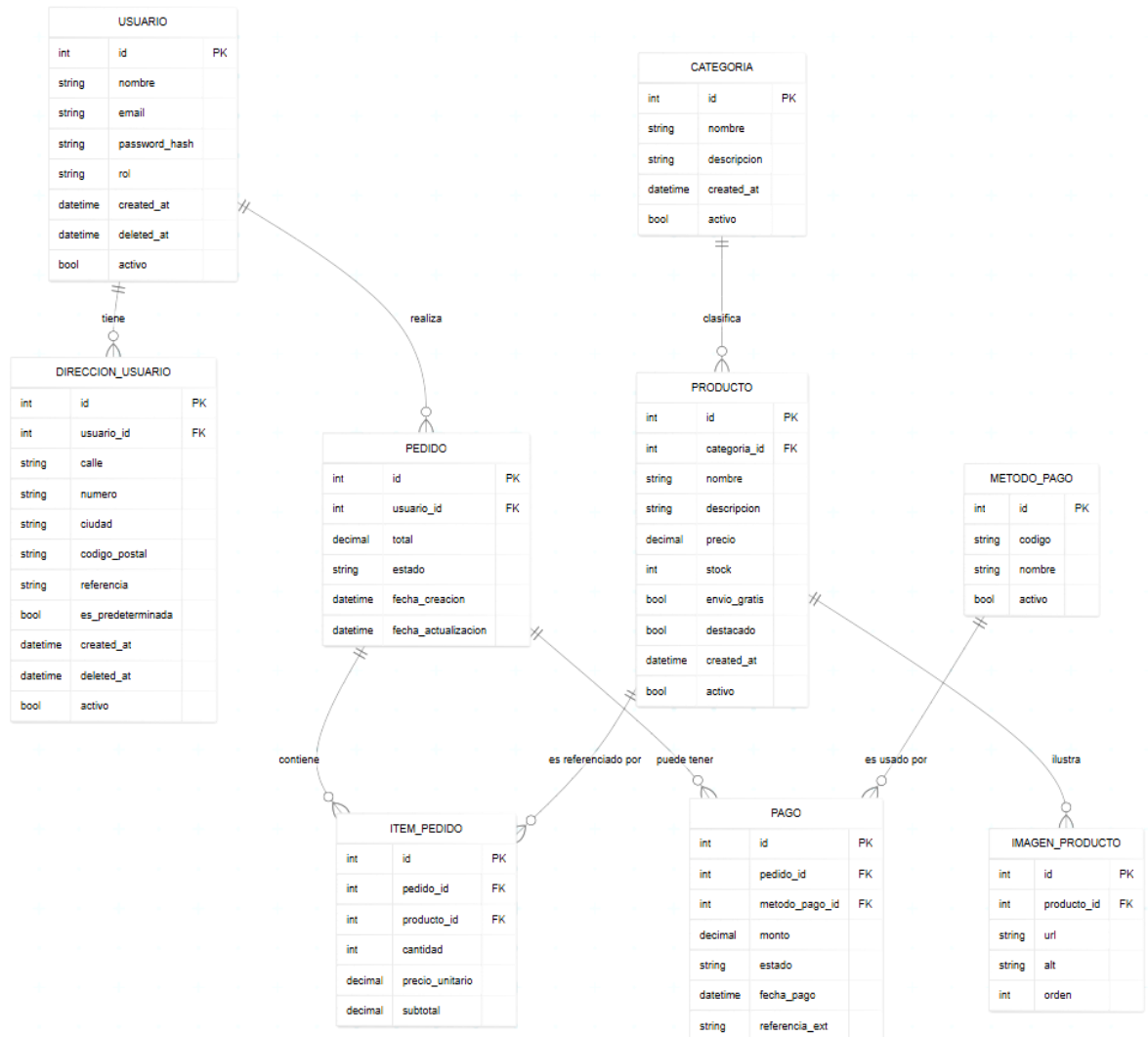
---

## Diagrama ER

Traslada el dominio a tablas con claves primarias/foráneas e integridad referencial.

- USUARIO, DIRECCION\_USUARIO(usuario\_id); CATEGORIA, PRODUCTO(categoria\_id); IMAGEN\_PRODUCTO(producto\_id).
- PEDIDO(usuario\_id) y ITEM\_PEDIDO(pedido\_id, producto\_id) implementan la descomposición encabezado–detalle; precio\_unitario evita inconsistencias ante cambios de lista.
- PAGO(pedido\_id, metodo\_pago\_id) registra la transacción y referencia al METODO\_PAGO.
- SUSCRIPCION permite alta de correos sin requerir cuenta.

El modelo está normalizado para reducir redundancias y asegurar consistencia transaccional.



## Diagrama de secuencia - flujo de compra

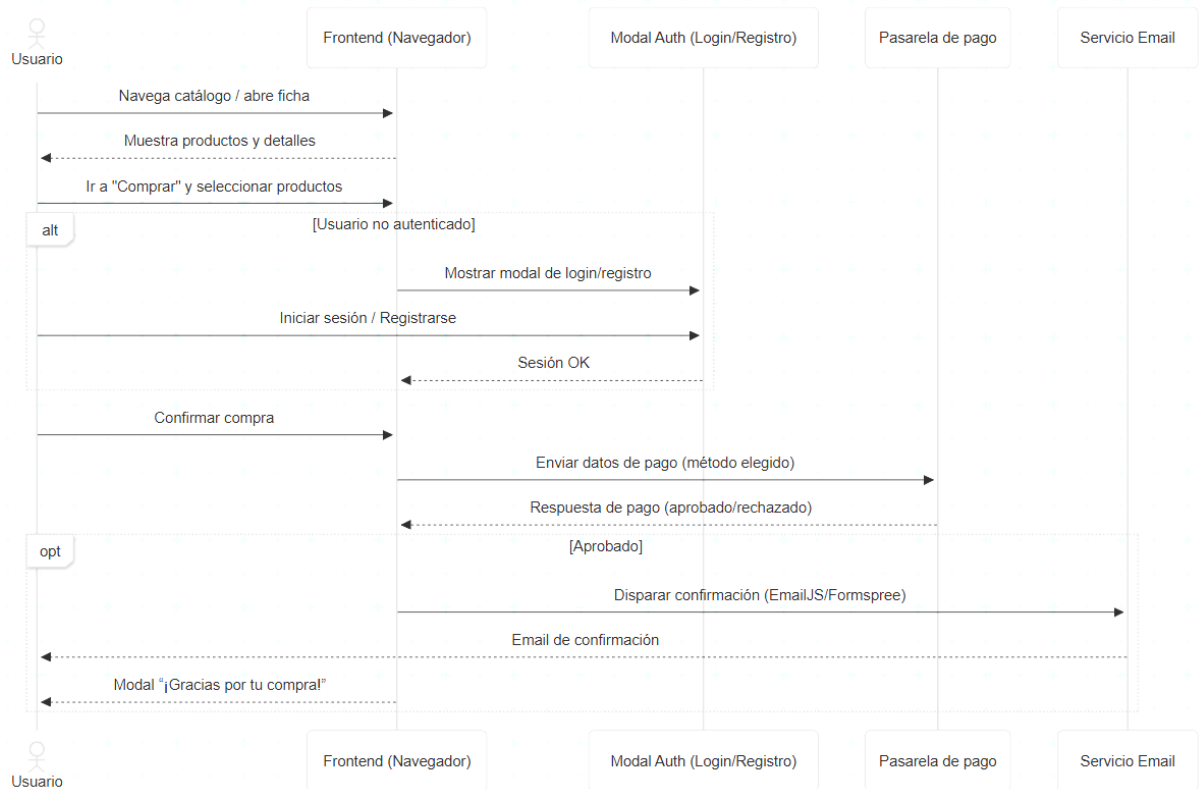
### a) Compra

- Usuario navega y selecciona productos. 2) Si no hay sesión, se abre el modal de autenticación.
- En “Confirmar compra”, se envían datos a la pasarela y se recibe el resultado.



- Si se aprueba, se envía confirmación por email (EmailJS/Formspre) y se muestra agradecimiento.

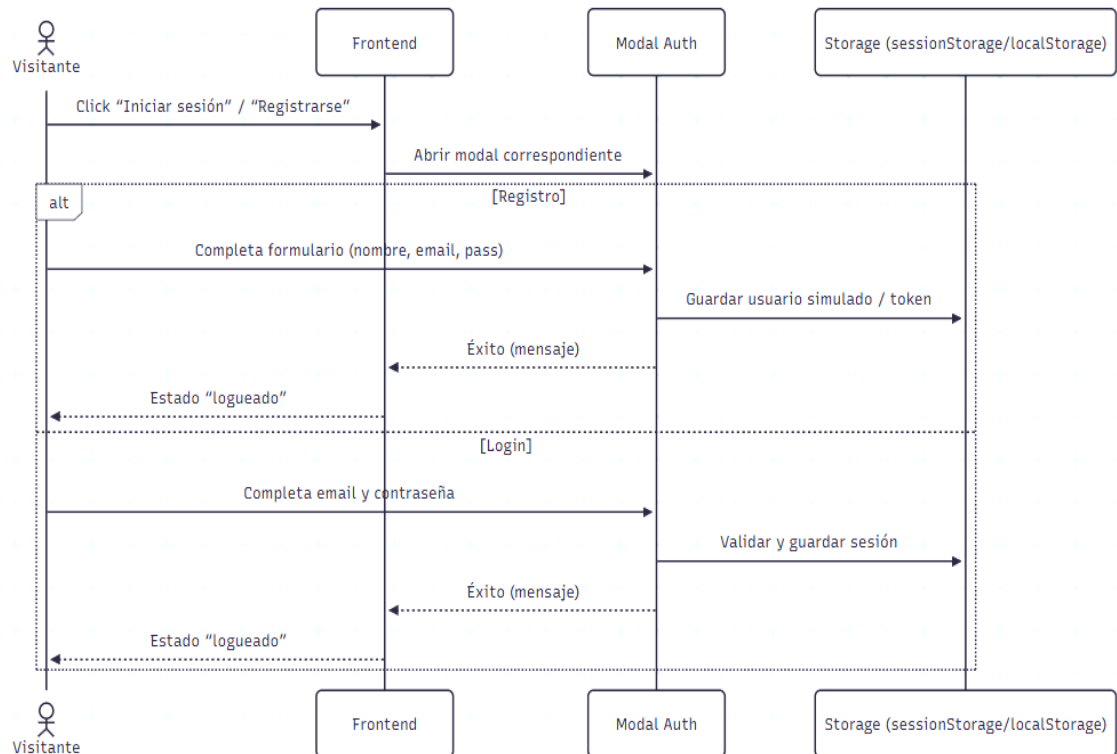
Refleja la coordinación entre UI, autenticación, pasarela y notificaciones.



#### b) Autenticación (Login/Registro)

- Apertura del modal desde la UI. 2) Registro o login con validación.
- Persistencia de estado de sesión (storage del cliente). 4) Actualización de la UI para continuar la compra.

Describe la transición de estados del usuario y su efecto inmediato en el flujo de compra.



El Diagrama de Clases define “qué existe”, el ER asegura datos consistentes y consultables, y las Secuencias muestran “cómo ocurre” la compra y la autenticación en tiempo de ejecución.

### **Evolución a Aplicación Web: Mejoras Implementadas con JavaScript**

El Trabajo Práctico N°2 transformó la plataforma Synk de un sitio estático a una aplicación web completamente funcional e interactiva. A continuación se detallan las mejoras implementadas mediante JavaScript, organizadas por funcionalidad.



---

## Sistema de Autenticación y Bienvenida

Se implementó un sistema completo de registro e inicio de sesión que permite a los usuarios crear cuentas y autenticarse en la plataforma, mejorando significativamente la experiencia de usuario y preparando el terreno para futuras funcionalidades personalizadas.

### Modal de Bienvenida Automático

Al ingresar al sitio por primera vez, el usuario es recibido por un modal de bienvenida que se activa automáticamente 1 segundo después de cargar la página. Este modal ofrece tres opciones claras.

***Iniciar sesión:*** Abre el formulario de login para usuarios registrados.

***Registrarse:*** Abre el formulario de creación de cuenta nueva.

***Seguir como invitado:*** Permite navegar sin autenticarse.

### Características técnicas.

- Incluye un checkbox "No mostrar de nuevo" que guarda la preferencia del usuario en `localStorage (clave: synk_hide_welcome)`.
- Solo se muestra si el usuario no está autenticado y no marcó previamente la opción de ocultar.
- Puede cerrarse mediante el botón "Cerrar" o haciendo clic en cualquiera de las opciones.

## Sistema de Registro (Sign Up)

El formulario de registro permite crear una cuenta nueva con validación completa en el frontend.

---



---

### **Campos requeridos.**

- Nombre completo
- Email (con validación de formato)
- Contraseña (mínimo 6 caracteres)
- Confirmación de contraseña

### **Sistema de Login**

El formulario de inicio de sesión permite a los usuarios registrados acceder a su cuenta.

### **Campos requeridos.**

- Email
- Contraseña

### **Interfaz Dinámica de Usuario**

Una vez autenticado, la interfaz se actualiza dinámicamente.

***Usuario NO autenticado (por defecto):***

[Iniciar Sesión] [Registrarse]

***Usuario autenticado:***

Nombre del Usuario [Cerrar Sesión]

### **Funcionalidad de Logout:**

- Al hacer clic en "Cerrar Sesión", se elimina la sesión de **localStorage**
- Se actualiza la UI para mostrar nuevamente los botones de login/register



- 
- Muestra mensaje de confirmación: "Has cerrado sesión correctamente"

### **Formulario de Suscripción Inteligente**

Se implementó un sistema de newsletter con doble notificación que permite a los usuarios suscribirse para recibir promociones y novedades, utilizando dos servicios externos complementarios.

#### **Funcionamiento General**

Cuando un usuario completa el formulario de suscripción en el footer y hace clic en enviar, se ejecutan dos procesos en paralelo:

***Formspree:*** Envía los datos del suscriptor al administrador del sitio

***EmailJS:*** Envía un email de confirmación automático al usuario que se suscribió

#### **Servicios utilizados:**

***Formspree*** (<https://formspree.io/f/xqadqpzq>): Endpoint que recibe el formulario y lo reenvía al email configurado

***EmailJS*** (*Service ID: service\_c4jrp86* | *Template ID: template\_ukx7hf9*): Servicio de envío de emails desde el cliente.

#### **Cómo llegan las diferentes notificaciones:**

***Notificación recibida por el administrador (Formspree).***

Al administrador le llega el mail, notificando que un usuario se ha suscrito esperando las diferentes novedades y promociones:



### New form submission on SuscripcionSynk

Someone just submitted a form on [synktechs.netlify.app/](#). Here's what they had to say:

email  
[mariavictoriasalazar560@gmail.com](mailto:mariavictoriasalazar560@gmail.com)

Submitted 04:36 AM - 07 October 2025

Mark as spam



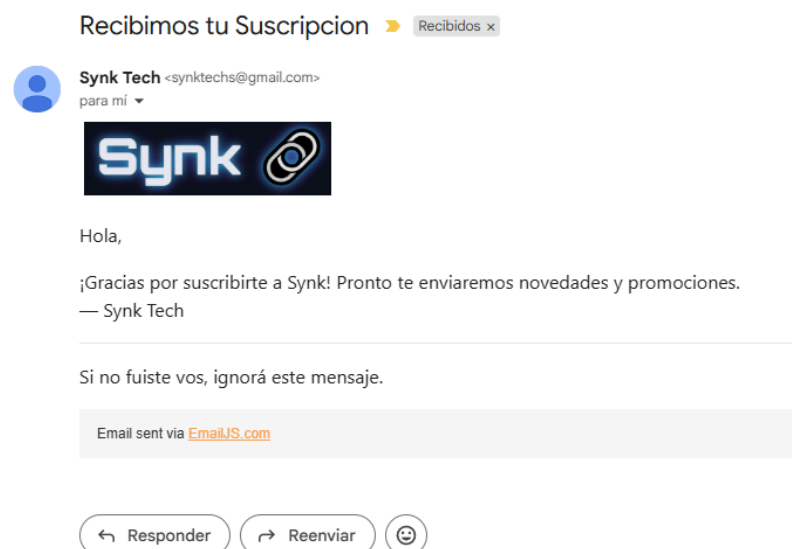
You are receiving this because you confirmed this email address on Formsfree.

Don't want these emails anymore? No problem!  
Simply remove the form from your website or [unsubscribe from this form's notifications](#).

*Salazar, F. (2025). Notificación de envío de formulario en Formsfree [Captura de pantalla]. Archivo personal.*

### Confirmación automática recibida por el suscriptor (EmailJS).

Al usuario, se le notifica que su suscripción se realizó con éxito y que pronto se le enviará novedades y promociones:







---

Salazar, F. (2025). *Confirmación automática recibida por el suscriptor (EmailJS)*

[Captura de pantalla]. Archivo personal.

### **Carrito de Compras Dinámico**

Se implementó un sistema de resumen de compra en tiempo real que permite al usuario seleccionar productos y visualizar instantáneamente el total de su pedido.

#### **Funcionalidad principal:**

##### ***Selección de productos:***

- En la página de compra (comprar.html), cada producto tiene un checkbox con atributos data-precio y data-nombre.
- El usuario puede marcar/desmarcar los productos que desea adquirir.

##### ***Actualización automática del resumen:***

- Cada vez que el usuario selecciona o deselecciona un producto, el resumen se actualiza instantáneamente sin necesidad de recargar la página.
- Si no hay productos seleccionados, muestra: "No hay productos seleccionados."
- Si hay productos seleccionados, muestra una lista con cada producto y su precio, más el total general.

##### ***Cálculo dinámico:***

- JavaScript recorre todos los checkboxes marcados.
  - Suma los precios de los productos seleccionados.
-



- 
- Formatea los números con separador de miles (formato argentino: \$1.299.000).
  - Actualiza el HTML del contenedor .resumen-compra con la información.

***Productos disponibles:***

- iPhone 15 Pro - \$1.499.999
- Auriculares Logitech G733 - \$189.999
- Notebook Acer Aspire Vero - \$899.999
- Monitor ViewSonic 27" - \$399.999
- Teclado HyperX Alloy FPS - \$129.999
- Mouse Glorious Model O RGB - \$69.999

Esta funcionalidad mejora significativamente la experiencia de compra al permitir que el usuario vea en tiempo real cuánto gastará antes de confirmar el pedido.

**Carrusel de Imágenes con Navegación por Puntos**

Se implementó un sistema de visualización de múltiples imágenes en las páginas de detalle de cada producto, permitiendo al usuario explorar diferentes vistas del artículo de forma intuitiva.

**Características implementadas:**

***Sistema de navegación visual:***

- Cada producto cuenta con un carrusel de 4 imágenes que muestran diferentes ángulos: frontal, lateral, en uso y accesorios.



- 
- Debajo de las imágenes se encuentran puntos de navegación (indicadores circulares) que permiten cambiar entre las vistas.
  - El punto correspondiente a la imagen activa se resalta visualmente, indicando al usuario en qué posición del carrusel se encuentra.

#### ***Funcionamiento técnico:***

El sistema utiliza radio buttons invisibles combinados con labels estilizados como puntos. Al hacer clic en cualquier punto, la imagen correspondiente se muestra instantáneamente mediante transiciones CSS. No requiere JavaScript para la funcionalidad básica, lo que garantiza un rendimiento óptimo y compatibilidad universal. El cambio entre imágenes es suave gracias a las transiciones CSS implementadas.

### **Menú Móvil Hamburguesa**

Se implementó un menú de navegación responsive que se adapta automáticamente a dispositivos móviles y tablets, mejorando significativamente la experiencia de usuario en pantallas pequeñas.

#### **Características implementadas:**

##### ***Botón hamburguesa dinámico:***

En dispositivos con ancho menor a 769 px, el menú tradicional se reemplaza por un icono de hamburguesa compuesto por tres líneas horizontales. El botón se crea dinámicamente mediante JavaScript si no existe en el HTML original, y se posiciona estratégicamente junto al logo de la marca en el header. Incluye una animación visual



---

atractiva donde las tres líneas se transforman en una "X" cuando el menú está abierto, proporcionando feedback visual claro del estado del menú.

### ***Apertura y cierre del menú:***

Al hacer clic en el botón hamburguesa, el menú se desliza desde el lateral con una animación suave. El usuario tiene múltiples formas de cerrarlo: haciendo clic nuevamente en el botón hamburguesa, seleccionando cualquier enlace del menú para navegar automáticamente, haciendo clic en cualquier parte de la página fuera del menú, o presionando la tecla ESC del teclado. Esta flexibilidad de opciones garantiza que cada usuario pueda interactuar con el menú de la forma que le resulte más natural.

### ***Bloqueo de scroll:***

Cuando el menú está abierto, se agrega automáticamente la clase **menu-abierto** al elemento body, lo que previene el desplazamiento de la página de fondo y evita confusiones en la navegación, especialmente en páginas largas. Al cerrar el menú, el scroll se restaura automáticamente sin pérdida de posición.

### ***Responsive automático***

En pantallas mayores a 769 px, el menú vuelve automáticamente a su estado horizontal tradicional. Si el usuario redimensiona la ventana mientras el menú móvil está abierto, este se cierra automáticamente al superar el breakpoint. Este comportamiento se logra mediante la API matchMedia de JavaScript, que detecta cambios en las media queries en tiempo real sin necesidad de recargar la página.

---



---

### ***Preservación del último foco***

El sistema recuerda qué elemento tenía el foco del usuario antes de abrir el menú (generalmente un enlace o botón). Al cerrarlo, devuelve el foco automáticamente a ese elemento, proporcionando una experiencia de navegación fluida y sin interrupciones, especialmente valiosa para usuarios que navegan mediante teclado o tecnologías asistivas.

### **Validación de Formularios**

Se implementó un sistema de validación en tiempo real para el formulario de compra, garantizando que los datos ingresados por el usuario cumplan con los formatos requeridos antes de procesar el pedido. Esta validación mejora significativamente la calidad de los datos recibidos y proporciona feedback inmediato al usuario sobre errores en sus datos.

#### **Campos validados**

**Email:** Se verifica que el correo electrónico tenga un formato válido mediante una expresión regular que comprueba la presencia de caracteres antes y después del símbolo "@", así como un dominio válido con extensión. La expresión regular utilizada es `/^\S+@\S+\.\S+$/,` que asegura que no haya espacios y que contenga los elementos básicos de un email válido. Si el formato es incorrecto, se muestra un alert con el mensaje "Ingresa un email válido" y el campo recibe el foco automáticamente para su corrección.

**Teléfono:** La validación acepta números, espacios, guiones y el símbolo "+", con una longitud mínima de 6 caracteres. Esto permite formatos como "+54 9 376 428-1622", "3764281622" o "376 428 1622". La expresión regular `/^[0-9+\s-]{6,}$/` garantiza



---

flexibilidad en el formato mientras mantiene la integridad del dato. Si el teléfono no cumple con estos requisitos, se muestra el mensaje "Ingresa un teléfono válido" y el campo recibe el foco.

**Método de pago:** Se verifica que el usuario haya seleccionado una opción del menú desplegable (tarjeta de crédito/débito, transferencia bancaria o efectivo). Si el campo queda vacío, se muestra el mensaje "Elegí un método de pago" y el select recibe el foco para que el usuario complete su selección.

### Comportamiento de la validación

La validación se ejecuta en el momento en que el usuario intenta enviar el formulario mediante el evento `submit`. JavaScript intercepta este evento y realiza las verificaciones necesarias antes de permitir el envío. Si algún campo no cumple con los requisitos, se previene el envío del formulario mediante `event.preventDefault()`, se muestra un mensaje de error específico mediante `alert()`, y el campo problemático recibe automáticamente el foco del teclado para facilitar su corrección inmediata.

### Orden de validación

El sistema valida los campos en secuencia lógica para una mejor experiencia de usuario:

1. Primero valida el email
2. Luego el teléfono (solo si el email es válido)
3. Finalmente el método de pago (solo si los anteriores son válidos)



---

Este enfoque secuencial evita mostrar múltiples alertas simultáneas y guía al usuario a corregir los errores uno por uno, reduciendo la frustración y mejorando la tasa de completitud del formulario.

### **Estructura del CSS (style)**

El sitio adopta una estética dark con acentos cian/violeta que refuerzan la identidad “hype/tech”. La tipografía principal es Orbitron y el layout prioriza legibilidad y contraste. El header es sticky con gradiente y borde luminoso, y la navegación incluye estados de hover con brillo. La portada (hero) usa una imagen full-width con overlay oscuro y títulos con glow para jerarquía visual. Las secciones informativas emplean una grilla responsiva de cards con elevación en hover para transmitir modernidad. El catálogo tiene dos presentaciones: una grilla de tarjetas (.producto-card) pensada para exploración visual, y una tabla estilizada dentro de .table-wrapper para comparación rápida; ambas mantienen consistencia de color y espaciados. En detalle de producto, el layout combina una galería/carrusel por CSS con un panel de información (.producto-info) de fondo translúcido y sombras suaves. El formulario de compra usa fieldsets, una grilla adaptativa (.grid-2) y anillos de enfoque (focus ring) en cian para accesibilidad. El footer integra el formulario de suscripción con botón compacto. Para móviles, el menú hamburguesa despliega un panel lateral con bloqueo de scroll y animaciones; además, hay estilos para modales, estados de error en formularios, lazy-load visual de imágenes y componentes utilitarios como “Back to top”. Las media queries afinan columnas, tamaños y gaps desde ~1024px hacia abajo, garantizando que todo sea usable en pantallas pequeñas sin perder identidad visual.

### **Estructura del HTML**

---



---

### ***index.html (portada)***

Es la puerta de entrada al sitio y establece la marca. Incluye el header global con logo y navegación, el hero con slogan, CTA hacia el catálogo y una sección informativa “¿Por qué elegir Synk?” con cuatro cards (rendimiento, diseño, seguridad, innovación). El footer cierra con datos de contacto y el formulario de suscripción. Los modales de Autenticación (login/registro) y de Bienvenida están presentes al final del documento para ser gestionados por JavaScript: el de Bienvenida se muestra al cargar (si el usuario no lo ocultó antes), ofreciendo iniciar sesión, registrarse o seguir como invitado; el de Autenticación permite registrar usuarios y acceder con credenciales.

### ***listado\_box.html (catálogo en tarjetas)***

Presenta el catálogo en formato visual, con una grilla responsiva de tarjetas. Cada producto muestra imagen principal, etiqueta de estado (Nuevo, Oferta, Envío Gratis), nombre, precio y un botón “Ver más” hacia su ficha. Este enfoque favorece el browsing rápido y atractivo, especialmente en móvil, y mantiene coherencia estética con sombras y brillos del tema general.

### ***listado\_tabla.html (catálogo en tabla)***

Ofrece la misma información pero en formato tabular para lectura densa y comparación. Cada fila incluye miniatura, nombre, categoría, precio, estado de stock y acceso a la ficha. Es útil para usuarios que desean revisar muchos ítems de un vistazo y tomar decisiones comparando columnas. La tabla está dentro de un wrapper con scroll horizontal para que se mantenga usable en pantallas chicas.





---

### ***comprar.html (formulario de compra)***

Contiene el formulario para datos del cliente (nombre, email, teléfono, dirección), la elección del método de pago y una lista de productos con checkboxes. Al marcar productos (cuando incluyen data-attributes de precio y nombre), el JavaScript genera un "Resumen de compra" dinámico con subtotal y total. La página incluye un modal de confirmación (anclado por hash #mensaje) que comunica el éxito del envío. La validación básica (email/teléfono/pago) se realiza en el front, y el diseño mantiene el mismo lenguaje visual del resto del sitio.

### ***producto\_\*.html (fichas de producto)***

Cada producto tiene su página de detalle con un carrusel de imágenes basado en inputs de tipo radio (sin dependencia de librerías), y una sección de información con título, precio, descripción y lista de características. Los botones de acción invitan a volver al catálogo o avanzar a la compra. El CSS le da un tratamiento "premium" al panel de detalles con fondo translúcido, sombras suaves y tipografías consistentes. El JS agrega mejoras opcionales como autoplay, navegación con teclado y swipe en pantallas táctiles (cuando está disponible).

## **Mejoras Futuras Planificadas**

Si bien el proyecto alcanzó un nivel funcional completo, existen funcionalidades identificadas durante el desarrollo que no fueron implementadas en esta versión pero que están preparadas técnicamente para su incorporación futura.



---

## Toolbar de Catálogo Interactivo

Se encuentra desarrollada en el código JavaScript (función `initCatalogoToolbar()`) una funcionalidad completa de filtrado y ordenamiento del catálogo que permitiría a los usuarios:

**Búsqueda en tiempo real:** Un campo de texto que filtraría los productos del catálogo mientras el usuario escribe, utilizando técnica de debounce (retardo de 200 ms) para optimizar el rendimiento y evitar búsquedas excesivas. La búsqueda compararía el texto ingresado con los nombres de los productos de forma insensible a mayúsculas/minúsculas.

**Filtro por categoría:** Un menú desplegable que permitiría mostrar sólo productos de categorías específicas (Celulares, Audio, Notebooks, Monitores, Teclados, Periféricos) o ver todas las categorías simultáneamente. Cada producto tendría un atributo data-categoría que permitiría su clasificación automática.

**Ordenamiento por precio:** Un selector con tres opciones (sin ordenar, ascendente, descendente) que reorganizará visualmente los productos según su valor. El sistema leería el atributo data-precio de cada producto y los reordenará dinámicamente en el DOM sin recargar la página.

Esta funcionalidad está completamente programada y solo requiere la incorporación de los elementos HTML correspondientes (`#buscar-producto`, `#filtro-categoría`, `#orden-precio`) y los atributos `data-nombre`, `data-categoría` y `data-precio` en cada tarjeta de producto para activarse automáticamente. Su implementación se pospuso para priorizar funcionalidades core del TP2, pero está lista para integrarse en futuras iteraciones del proyecto.

## Errores Críticos identificados y plan de mejora

---



---

Durante el desarrollo del Trabajo Práctico N°2, se identificaron ciertos errores críticos que, si bien no impiden el funcionamiento del sitio en su contexto académico, deben documentarse para su corrección en futuras iteraciones.

### ***Uso de localStorage para Almacenamiento de Contraseñas***

El sistema almacena contraseñas en texto plano dentro de localStorage, representando un riesgo de seguridad. Cualquier script malicioso podría leer estas contraseñas directamente.

El objetivo del TP2 era implementar la lógica de autenticación usando JavaScript del lado del cliente sin backend. El uso de localStorage permitió simular persistencia de datos y demostrar el flujo completo de registro, login y logout.

En el TP3 se implementará un backend (Node.js con Express) que gestionará autenticación segura con contraseñas encriptadas mediante bcrypt en una base de datos. El frontend solo recibirá tokens de sesión (JWT) sin exponer información sensible.

### ***Validación Únicamente del Lado del Cliente***

Todas las validaciones se realizan exclusivamente en el frontend mediante JavaScript. Un usuario con conocimientos técnicos podría deshabilitar JavaScript o modificar el código para enviar datos inválidos.

El TP2 se enfocó en demostrar validación en tiempo real para mejorar la experiencia de usuario. Esta validación client-side es fundamental para interfaces reactivas, pero no se contempló backend en esta fase académica.



---

El backend del TP3 incluirá validación server-side para todos los datos recibidos, implementando el principio de "nunca confiar en el cliente". Las validaciones client-side se mantendrán para UX, complementadas con validaciones del servidor y sanitización de inputs.



---

## CONCLUSIÓN

El desarrollo del Trabajo Práctico N°2 permitió transformar exitosamente la plataforma Synk de un sitio web estático a una aplicación web completamente funcional e interactiva, cumpliendo con todos los objetivos planteados en la consigna. La incorporación de JavaScript como lenguaje de programación principal no solo agregó dinamismo al sitio, sino que mejoró significativamente la experiencia de usuario mediante validaciones en tiempo real, feedback inmediato y funcionalidades que responden instantáneamente a las acciones del usuario.

Las funcionalidades implementadas demuestran un dominio integral de los conceptos estudiados en la materia. El sistema de autenticación con localStorage simula un entorno real de gestión de usuarios, la integración de servicios externos como Formspree y EmailJS evidencia la capacidad de trabajar con APIs de terceros, y el menú móvil hamburguesa con accesibilidad completa refleja un compromiso con las mejores prácticas de desarrollo web. La modularidad y organización del código JavaScript facilita el mantenimiento y la expansión futura del sistema.

El aspecto responsive del sitio alcanzó su máximo potencial mediante JavaScript que detecta y responde a cambios en el tamaño de la ventana en tiempo real, garantizando una experiencia óptima tanto en dispositivos móviles como en desktop. Aunque el proyecto alcanzó un nivel funcional completo para un entorno académico, existen oportunidades de mejora identificadas durante el desarrollo, como la funcionalidad de búsqueda y filtrado de productos que permanece lista para implementación futura, y la posible migración hacia un backend seguro con encriptación de contraseñas y gestión de inventario real.



---

En términos de aprendizaje, este trabajo práctico consolidó conocimientos fundamentales sobre manipulación del DOM, gestión de eventos, almacenamiento en el navegador, validación con expresiones regulares, integración de APIs externas y técnicas de optimización de rendimiento. El proyecto Synk representa ahora una aplicación web completa que equilibra funcionalidad, diseño y experiencia de usuario, preparada para continuar evolucionando hacia una plataforma de comercio electrónico profesional.

Link de Netlify:

<https://synktechs.netlify.app>

Link del Repositorio:

<https://github.com/Facundosalaa/SYNK-.git>