

---

**Carlos Trinidad**

trinidad.carlos@hotmail.com

(+52) 999 1351 550

# Eagle Wear (sps-web-shop)

13 de marzo de 2025

## RESUMEN

Este documento detalla el proceso de análisis, diseño, desarrollo e implementación de la aplicación web. A partir de la evaluación de los requerimientos, se establecen las siguientes metas:

## METAS

1. Seleccionar y justificar las tecnologías adecuadas para el desarrollo del proyecto.
2. Desarrollar un prototipo funcional inicial que valide los requisitos establecidos.
3. Elaborar una propuesta de diseño de interfaz de usuario y experiencia de usuario que se adhiera a las mejores prácticas.
4. Generar la documentación técnica necesaria y comprobar las pruebas unitarias.
5. Implementar y publicar la aplicación web en el entorno de producción.

## ANÁLISIS

Con base en los requerimientos y especificaciones, se seleccionó **React JS** (versión más reciente y estable) y **TypeScript** para el desarrollo del frontend. Se optó por TypeScript para beneficiarse del tipado estático, componentes funcionales y hooks para el acceso a recursos.

La API especificada en el requerimiento (<https://fakestoreapi.com/>) permite la creación y autenticación de usuarios, pero presenta dos limitaciones: el inicio de sesión con usuarios recién creados no funciona y solo admite nombres de usuario en lugar de correos electrónicos. Por lo tanto, se desarrollará una solución de autenticación personalizada.

Dado que la aplicación se desplegará en la nube, se ha seleccionado **Google Cloud Platform** (GCP) como plataforma principal. Esta decisión se basa en la simplicidad de GCP y la experiencia previa con la plataforma. Además, se propone integrar Firebase para gestionar la autenticación de usuarios, la base de datos y la implementación de una arquitectura serverless, si fuera necesario.

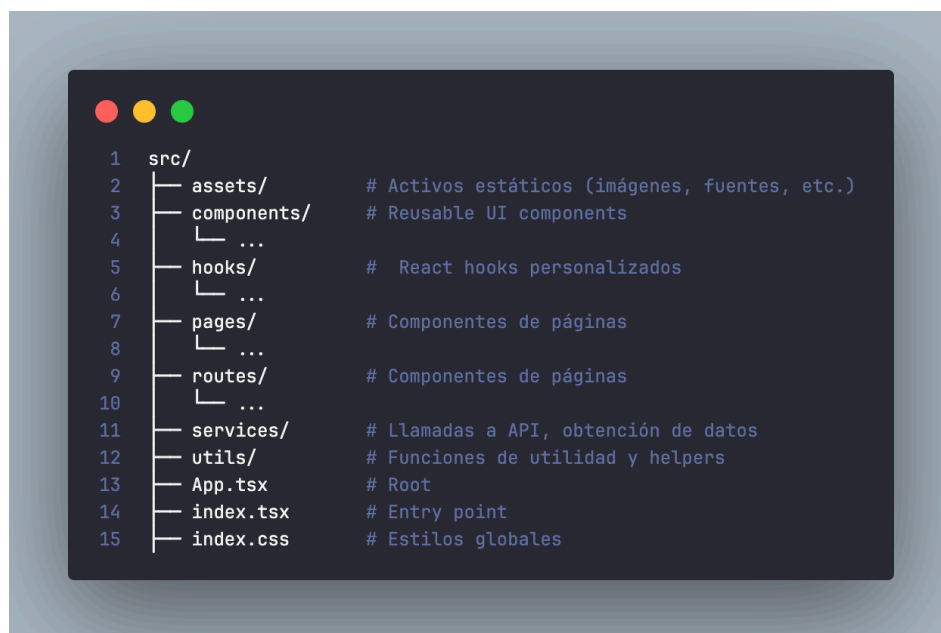
---

## DESARROLLO

### Generar prototipo funcional

Para la estructuración del proyecto, se ha optado por *Vite*, en combinación con React y TypeScript, como se estableció previamente. La elección de Vite se fundamenta en su capacidad para utilizar módulos ES nativos, lo que resulta en una mejora significativa de los tiempos de carga. Además, Vite ofrece una configuración inicial sencilla y altamente personalizable.

Con esta base, se procede a la personalización de la estructura del proyecto.



A continuación, se generan los componentes de alto nivel necesarios para configurar las rutas privadas y públicas de la aplicación.

Para la gestión del sistema de rutas, se implementará *react-router*, debido a su simple configuración y a las funcionalidades que permiten un enrutamiento eficaz.

Con el sistema de rutas configurado, se integrará la autenticación de usuarios mediante Firebase. Se creará un proyecto en Firebase y se instalará la librería necesaria para acceder a sus servicios.

Para configurar Firebase, primero se crea un proyecto en la consola y se habilitan los proveedores de autenticación deseados (correo electrónico, Google, Facebook). La integración de Facebook requiere configuración adicional en su plataforma de desarrolladores. Luego, se

---

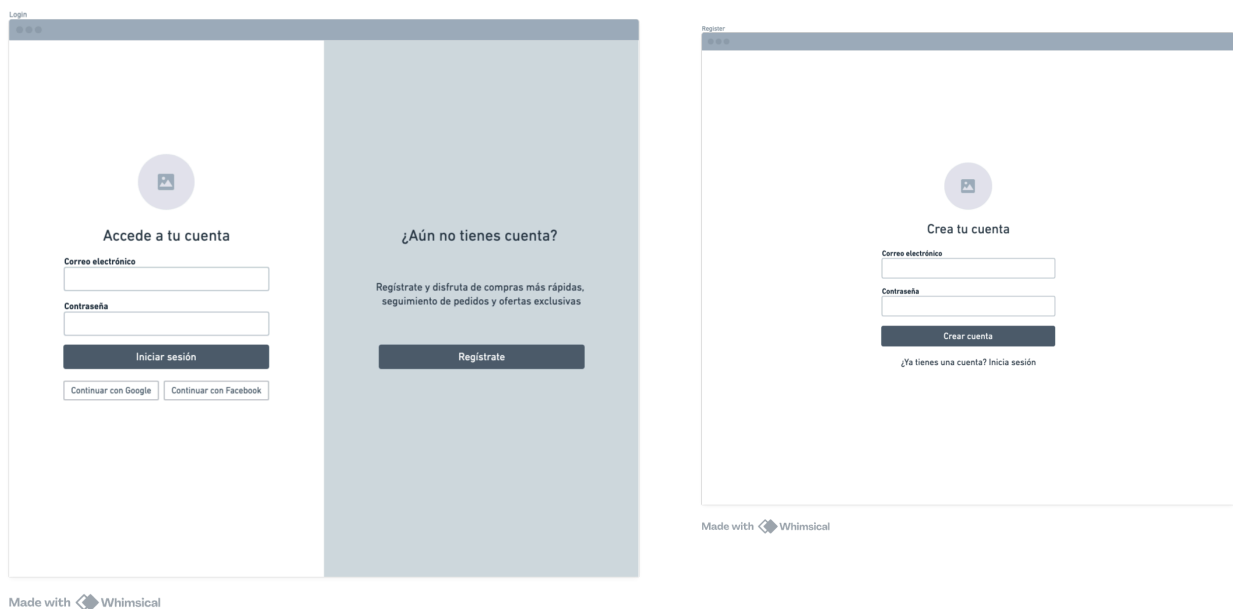
Se procedió a instalar el SDK de Firebase y a configurar firebase.ts en la carpeta services. Aunque inicialmente se contempló excluir este archivo del repositorio por motivos de seguridad, se decidió incluirlo para agilizar el proceso de pruebas locales. Se generaron también los archivos .gitignore y firebase.example.ts

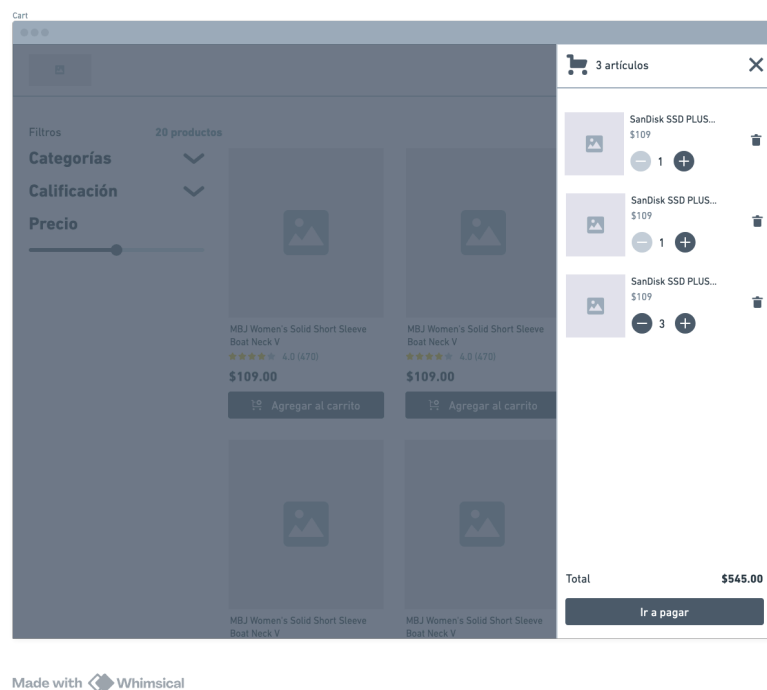
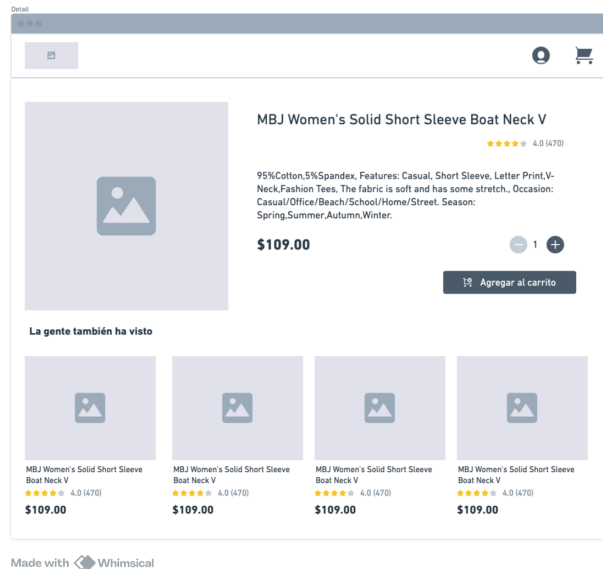
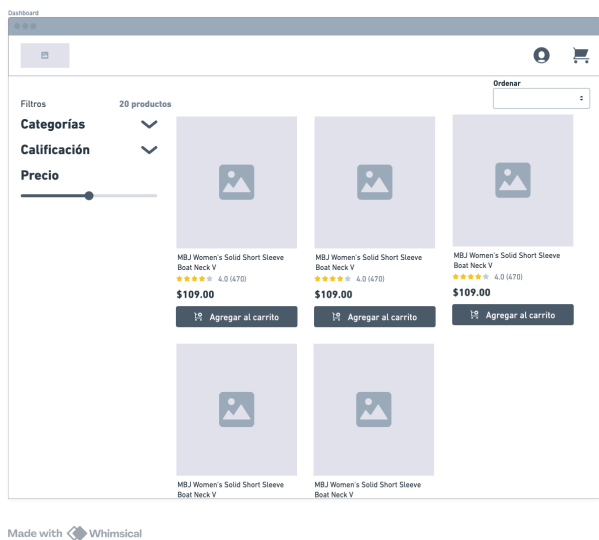
Implementaré pruebas unitarias para los componentes a medida que avance el desarrollo. Para ello, utilizaré Vitest. La configuración es rápida y simple, requiriendo únicamente la instalación de las dependencias y la especificación de la ruta de configuración.

Se crearán formularios de registro e inicio de sesión con botones de envío y enlaces de navegación. También se añadirá un botón de cierre de sesión en la pantalla principal. La integración con Firebase permitirá probar el flujo básico de autenticación.

Tras la autenticación exitosa, se procederá a la construcción del catálogo de productos. Paralelamente, se analizará la respuesta del endpoint para fundamentar la propuesta de diseño subsiguiente.

A partir de los datos disponibles, se creará un mockup del diseño de la aplicación. Esta propuesta, sujeta a cambios durante la implementación, servirá como base para el desarrollo del diseño.





A partir de los colores del logotipo, se definirá un diseño con predominio de tonos claros y contrastes oscuros. El objetivo es crear una interfaz visualmente limpia que enfatice los productos, maximizando así las ventas.

Se procederá a construir los componentes, desde los más básicos hasta los más complejos, priorizando la consistencia y reutilización. Se emplea Material-UI para acelerar el desarrollo del diseño.

---

Además, decidí utilizar i18next para la internacionalización debido a que es una de las librerías de (i18n) más populares y flexibles en el ecosistema de JavaScript.

En primer lugar, configuramos el tema de Material-UI con los colores primarios, secundarios y tamaño de fuente. Después procedemos a crear los componentes del formulario que estaremos utilizando en otros componentes.

Al concluir las pantallas anteriores, se abordará el home, iniciando con el layout principal y la barra de navegación, un componente transversal a toda la aplicación. Se añadirá un botón de usuario para la gestión del idioma y la sesión. La gestión del carrito se realizará mediante Zustand, una biblioteca que ofrece ventajas en rendimiento y escalabilidad, en comparación con el Contexto de React.

A continuación, se abordará el diseño del layout de la pantalla principal. Se propone implementar un drawer para la gestión de filtros, el cual podrá ser ocultado en la vista responsiva, permitiendo así maximizar el espacio disponible para la visualización de los productos.

Tras generar las tarjetas de producto, se implementará la cuadrícula de productos, incluyendo filtros, categorías y ordenamiento. Debido a la limitación del endpoint, estas funcionalidades se implementarán en el frontend. He implementado el filtrado, ordenamiento y paginado mediante parámetros en la URL para mejorar la experiencia del usuario, permitiendo búsquedas persistentes y fácilmente compartibles.

Tras la presentación de los productos, implementaré el carrito de compras. Utilizaré el mismo contexto del drawer para gestionar el carrito, permitiendo agregar productos desde cualquier sección de la aplicación.

Finalmente, abordaremos la vista detallada del producto. Obtendré el ID desde la URL, recuperaré la información del producto y la mostraré. Comenzaré con el diseño, que difiere de las vistas anteriores.

Hecho esto solo hago una revisión rápida para ver detalles visuales y ajustes en la parte responsiva.

## **PUBLICACIÓN DE LA APLICACIÓN**

Dado que la publicación es sencilla, la realizaré de inmediato. No anticipo cambios significativos en la próxima sesión de programación. Procedo a configurar Firebase Hosting, estableciendo 'dist' como la carpeta pública, y genero los archivos de producción. La aplicación está ahora alojada en: <https://sps-web-shop-ctrinidad.web.app>

---

## DOCUMENTACIÓN

Una vez completado el desarrollo, me dediqué a la creación del README. Para asegurar la claridad y utilidad del documento, incorporé capturas de pantalla que destacan las características más relevantes de la aplicación. Asimismo, incluí las instrucciones necesarias para la instalación, ejecución y despliegue.

## CONCLUSIONES

Disfruté mucho este ejercicio práctico. Desde la lectura de las instrucciones, pude visualizar los componentes y funcionalidades que incorporaría. A pesar de que el producto no está completo, estoy satisfecho con el avance logrado en el plazo establecido.

Me permito hacer algunas sugerencias con respecto al inicio de sesión. Las instrucciones mencionan el uso de la API, y entiendo que la intención era utilizarla para la autenticación. Sin embargo, observé que el sistema actualmente soporta el inicio de sesión con nombre de usuario, no con correo electrónico y contraseña como se indica en el documento. Para evitar confusiones, sugiero actualizar las instrucciones y aclarar que el inicio de sesión debe realizarse a través de la API, especificando el uso de nombre de usuario.

Agradezco la oportunidad y su tiempo, y espero futuras colaboraciones.