

Documentación de APIs Utilizadas

Introducción

En el desarrollo del sistema de comercio electrónico para una marca de ropa, se integraron diversos servicios externos mediante APIs para agregar funcionalidades esenciales como autenticación de usuarios, gestión de imágenes en la nube y simulación de pagos. A continuación, se detallan las APIs utilizadas, sus propósitos dentro del sistema y cómo fueron integradas a nivel técnico.

1. Firebase Authentication (Google - Firebase)

- Propósito:

Proveer un sistema de autenticación seguro, escalable y moderno, permitiendo a los usuarios registrarse e iniciar sesión con correo electrónico y contraseña, o mediante autenticación federada (por ejemplo, Google).

- Implementación Técnica:

- Utilizada desde el frontend (React + TypeScript) para el login/registro de usuarios.
- Firebase genera un firebaseUid único para cada usuario autenticado.
- Este UID se guarda en la base de datos relacional (Usuario.firebaseUid) y se utiliza como identificador seguro.
- Se valida el token de Firebase en el backend antes de permitir acciones protegidas.

Documentación oficial: <https://firebase.google.com/docs/auth>

- Ventajas:

- Seguridad integrada con JWT.
- Soporte para múltiples proveedores de autenticación.
- Manejo de sesiones persistentes.

2. Mercado Pago – Checkout Pro

- Propósito:

Simular el proceso de pago online, permitiendo a los usuarios completar compras con medios ficticios en un entorno de pruebas.

- Implementación Técnica:

- Se crea una preferencia de pago desde el backend mediante la API.
- Uso de cuentas sandbox y credenciales de prueba.

- El frontend recibe una URL y redirige al usuario al Checkout Pro.
- Al finalizar, se actualiza el estado del pedido según el resultado del pago.

Documentación oficial: <https://www.mercadopago.com.ar/developers/es/guides/payments/web-payment-checkout/introduction>

- Ventajas:
 - Entorno seguro de simulación.
 - Integración sencilla.
 - Flujo adaptable a futuras integraciones reales.

3. Cloudinary (Servicio de Gestión de Imágenes)

- Propósito:

Permitir la carga, almacenamiento y visualización eficiente de imágenes de productos sin consumir espacio del servidor local.

- Implementación Técnica:
 - Las imágenes se suben desde el frontend a Cloudinary utilizando su API.
 - Se obtiene una URL pública que se guarda en la base de datos.
 - La URL se utiliza para renderizar las imágenes de los productos.

Documentación oficial: https://cloudinary.com/documentation/image_upload_api_reference

- Ventajas:
 - Alojamiento rápido con CDN global.
 - Optimización automática.
 - Escalabilidad sin infraestructura adicional.

Conclusión

La elección de estas APIs responde a la necesidad de ofrecer un sistema moderno, seguro y funcional con bajo costo de mantenimiento. Estas herramientas permiten agilidad en el desarrollo, escalabilidad gracias al uso de servicios en la nube, y seguridad al delegar procesos críticos a plataformas confiables. Estas integraciones fueron fundamentales para cumplir con los requerimientos funcionales del sistema, ofreciendo a los usuarios una experiencia fluida y profesional.