

Documentación Técnica

HACKTECH

DOCUMENTACIÓN TÉCNICA DEL PROYECTO HACKTECH

1. Introducción

En el marco de este proyecto, se nos encomendó la tarea de interactuar con la API de Inditex, teniendo la opción de elegir entre tres servicios disponibles. Optamos por integrar dos de ellos: Search Productos y Visual Search. Para facilitar esta interacción, se diseñó e implementó una aplicación web estructurada en dos componentes principales: Frontend y Backend.

2. Arquitectura del Sistema

2.1. Descripción General

La API de Inditex permite la búsqueda de productos pertenecientes a marcas como Zara, Zara Home, Oysho, Massimo Dutti, Stradivarius, Massimo Dutti y Lefties. Implementa un sistema de autenticación basado en OAuth 2.0 y admite consultas parametrizadas para filtrar resultados por nombre de producto y marca. La API devuelve información detallada de los productos, incluyendo:

- **ID del producto**
- **Nombre**
- **Precio**
- **Enlace al producto**
- **Marca**

Adicionalmente, la API maneja distintos tipos de respuestas a conexiones http.

- **200 Ok: Solicitud exitosa**
- **400 Bad Request: Parámetros de consulta incorrectos.**
- **401 Unauthorized: Problemas de autenticación.**
- **403 Forbidden: Acceso no autorizado.**
- **404 Not Found: Producto no encontrado.**
- **429 Too Many Requests: Se ha excedido el límite de peticiones.**
- **500 Internal Server Error: Error en el servidor.**

La API también incorpora soporte para paginación, utilizando los parámetros page y perPage. Los servidores disponibles para la API son:

- Producción: <https://api.inditex.com/searchmpma>
- Sandbox (pruebas): <https://api-sandbox.inditex.com/searchmpma-sandbox>

3. Desarrollo del Frontend

El Frontend se encarga de la gestión de la interfaz de usuario y de la comunicación con el Backend. Dentro de sus funcionalidades principales se incluyen:

- Renderización de la página web y sus componentes visuales.
- Interacción del usuario con la API de Inditex.
- Manejo de eventos en la interfaz, tales como búsquedas de productos, resaltar marcas y cargar imágenes.
- Obtención y actualización de las imágenes de productos desde fuentes externas.
- Manejo de errores y validaciones de los datos obtenidos de la API.

4. Desarrollo del Backend

El Backend es el encargado de implementar todas las funcionalidades necesarias para procesar las peticiones a la API de Inditex. Para su desarrollo se utilizó Python junto con el framework FastAPI, el cual facilita la creación y gestión de API REST.

4.1. Instalación del Entorno

Para desplegar el backend en un entorno local, se deben ejecutar los siguientes comandos en la terminal:

```
sudo apt install python3-venv
python3 -m venv envHUDC
source envHUDC/bin/activate
pip install fastapi[standard]
pip install requests
fastapi dev main.py
```

4.2. Funcionalidades del Backend

En cuanto al funcionamiento de la aplicación tenemos varias opciones, en caso de no especificar la tienda (Bershka, Massimo Dutti, Zara ...) aparecerán los datos de Zara por defecto. Para que ponga los datos de otras tiendas hay que especificarlo, a excepción de Lefties que no está correctamente implementada en la API.

En la aplicación se implementan las siguientes operaciones clave:

1. Filtrar productos por marca
2. Obtener un token de autenticación para interactuar con la API de Inditex.
3. Obtener resultados de Google mediante la API de Google Custom Search.
4. Realizar búsquedas de productos por texto en la API de Inditex.

5. Realizar búsquedas de productos por imagen, permitiendo identificar productos a partir de una imagen de referencia.

5. Problemas Encontrados y Solución Implementada

Durante la implementación del proyecto, nos enfrentamos a ciertas restricciones impuestas por Inditex respecto a la obtención de imágenes de productos. Inicialmente, solo podíamos obtener la URL de la página donde se encontraba la imagen, pero no la imagen en sí.

Para solventar este inconveniente, decidimos emplear la API de Google Custom Search, filtrando los resultados para que solo mostraran imágenes relacionadas con las tiendas de Inditex. Esto nos permitió mejorar la experiencia del usuario al proporcionar imágenes directamente en los resultados de búsqueda.

6. Conclusiones y Mejoras Futuras

El desarrollo de esta aplicación web nos permitió explorar la integración de APIs externas, el uso de autenticación OAuth 2.0 y el manejo eficiente de peticiones HTTP. A pesar de las limitaciones encontradas, logramos implementar una solución funcional y escalable.

Entre las mejoras futuras que podrían implementarse destacan:

1. Optimizar la gestión de imágenes para una carga más eficiente.
2. Incluir un sistema de caché para reducir la cantidad de peticiones a la API.
3. Ampliar la integración a otras marcas del grupo Inditex que actualmente no estén completamente soportadas.
4. Mejorar la autenticación mediante JWT para una gestión segura de los usuarios.

7. Dependencias y Licencias

1. FastAPI:

- Licencia: MIT License

- Descripción: FastAPI es un framework web moderno y de alto rendimiento para construir APIs con Python 3.6+ basado en estándares como OpenAPI y JSON Schema.

2. Requests:

- Licencia: Apache License 2.0

- Descripción: Requests es una librería HTTP simple y elegante para Python, construida para hacer solicitudes HTTP de manera fácil y rápida.

8. Equipo de desarrollo

Nombre y apellidos	Github
Jaime Camiña Caneda	JaimeCHUB
Alejandro Fernández Expósito	AlejandroFEX

Juan Vazquez Folgueira Juan_vazquez_folgueira