# Documentación del Script Savant\_Reto\_Tecnico

# **Introducción**

# Este script automatiza la actualización de información sobre productos en oferta de MercadoLibre Colombia y obtiene datos de precios y disponibilidad en Amazon Colombia. La información actualizada se guarda en un archivo CSV para su posterior análisis.

# **Requisitos Previos**

# Python 3.x instalado en el sistema.

# **Bibliotecas necesarias: requests, beautifulsoup4, csv.**

# **Pasos para Ejecutar el Script**

# **1. Instalación de Dependencias**

# Instala las dependencias necesarias ejecutando el siguiente comando en tu terminal:

# **pip install requests beautifulsoup4**

# **2. Ejecución del Script**

# Ejecuta el script Savant\_Reto\_Tecnico.py utilizando el siguiente comando:

# **python Savant\_Reto\_Tecnico.py**

# **3. Observación de la Ejecución**

# El script comenzará a buscar productos en oferta en MercadoLibre Colombia y buscará información adicional en Amazon Colombia. Los resultados actualizados se guardarán en un archivo CSV llamado resultados.csv.

# **4. Configuración de la Actualización Automática (Opcional)**

# Si deseas actualizar automáticamente la información en intervalos regulares, puedes configurar el script para que se ejecute periódicamente utilizando una tarea programada (como cron en sistemas Unix) o un servicio de automatización en la nube.

# **Detalles de Implementación**

# **Funciones Principales:**

# **obtener\_productos\_mercadolibre():** Esta función realiza una solicitud HTTP para obtener los productos en oferta de MercadoLibre Colombia utilizando web scraping con BeautifulSoup.

# **buscar\_en\_amazon(productos):** Esta función toma los productos obtenidos de MercadoLibre y realiza búsquedas en Amazon Colombia para obtener información adicional sobre precios y disponibilidad.

# **guardar\_resultados(productos):** Esta función guarda los resultados en un archivo CSV llamado resultados.csv.

# **actualizar\_info():** Esta función ejecuta un bucle infinito que actualiza periódicamente la información utilizando las funciones anteriores.

# **Consideraciones de Eficiencia:**

# **Caché de Datos:** Implementa una capa de caché para almacenar datos temporales y reducir la carga en los servicios principales. Esto puede mejorar significativamente el rendimiento al evitar consultas repetitivas a fuentes externas.

# **Programación Asincrónica:** Utilizar un enfoque de programación asincrónica para manejar las solicitudes de manera eficiente, permitiendo que los microservicios respondan a eventos y actualicen los datos de forma concurrente.

# **Escalabilidad Automática:** Configurar tu infraestructura para que pueda escalar automáticamente según la demanda. Esto podría incluir el uso de contenedores y orquestadores como Docker y Kubernetes, así como servicios de nube que ofrecen escalabilidad automática como AWS ECS o Google Kubernetes Engine.

# **Consideraciones de Seguridad**

# Es importante asegurarse de que el scraping de datos cumpla con los términos de servicio de los sitios web consultados (MercadoLibre y Amazon).

# Se recomienda utilizar técnicas de manejo seguro de datos, como el almacenamiento seguro de tokens de autenticación si es necesario.