

# Análisis de Entradas Disponibles - Prueba Técnica PHP

## Descripción del Proyecto

Este proyecto es una implementación de la prueba técnica propuesta para demostrar mis habilidades en PHP, enfocándome en la calidad y claridad del código, así como en la capacidad para manejar información proveniente de diferentes fuentes. El objetivo principal fue desarrollar una aplicación en PHP que reciba como parámetro una URL de un evento y muestre las entradas disponibles por sector, fila y precio para eventos de dos plataformas diferentes: VividSeats y SeatGeek.

## Funcionalidades Implementadas

### 1. Análisis de Entradas Disponibles en VividSeats

- **URL de Ejemplo:** [VividSeats - Real Madrid](#)
- **Descripción:** Implementé una función en PHP que analiza la página de un evento en VividSeats y muestra las entradas disponibles por sector, fila y precio.
- **Limitaciones:**
  - La API creada para VividSeats solo funciona si el navegador está conectado.
  - Después de varios intentos, si los tickets salen [], es necesario recargar el navegador y pasar el CAPTCHA, lo que obliga a mantener el navegador abierto y realizar clics manuales en el CAPTCHA.
  - Para automatizar completamente este proceso, se necesitaría crear un servicio adicional utilizando Flask y Selenium para manejar los CAPTCHAs, lo cual no se implementó debido a las restricciones mencionadas.

## Limitaciones y Consideraciones

### 2. Análisis de Entradas Disponibles en SeatGeek

- **URL de Ejemplo:** [SeatGeek - Taylor Swift](#)
- **Descripción:** No implementé la funcionalidad para SeatGeek debido a las siguientes razones:
  - La página de SeatGeek está protegida por CAPTCHA, lo que dificulta el web scraping directo.
  - La única manera efectiva de acceder a la información sería utilizando Selenium para automatizar el navegador y superar los CAPTCHAs.
  - Esto requeriría la creación de un servicio en Docker que ejecute Selenium y otro servicio con Flask para manejar los CAPTCHAs, lo que aumentaría significativamente la complejidad del proyecto.

## Problemas Éticos y Legales

- **Web Scraping y CAPTCHAs:** Intentar realizar web scraping en sitios protegidos por CAPTCHA puede violar los términos de servicio de las plataformas y plantear problemas legales.
- **Uso de APIs Oficiales:** Es preferible utilizar APIs oficiales proporcionadas por las plataformas (como la API de SeatGeek) para acceder a los datos de manera legítima y eficiente.

## Recomendaciones

1. **Uso de la API Oficial de SeatGeek:**
  - **URL:** [SeatGeek API](#)
  - Utilizar la API oficial simplificaría enormemente la implementación, permitiendo acceder a los datos de eventos de manera más segura y confiable sin necesidad de realizar web scraping.
2. **Evitar Web Scraping en Sitios Protegidos:**
  - Dado que las APIs oficiales están disponibles, es altamente recomendable utilizarlas en lugar de intentar bypassar las protecciones como los CAPTCHAs.
  - Esto asegura el cumplimiento de los términos de servicio y evita posibles problemas legales y técnicos.

## Video de Demostración

En el directorio de este proyecto se encuentra un video de demostración.

## Instrucciones de Despliegue

Este proyecto utiliza Docker para facilitar el despliegue y la ejecución del servicio.

### Requisitos Previos

- [Docker](#) instalado en tu máquina.

### Pasos para Desplegar

1. **Clonar el Repositorio:**

```
git clone https://github.com/tu-usuario/analisis-entradas.git
cd analisis-entradas
```

2. **Desplegar el Servicio:**

```
docker-compose up -d
```