

DEPARTAMENTO: Ciencias de la Ingeniería **PROFESOR:** PhD. Nelson Herrera

CARRERA: Ingeniería Informática

ESTUDIANTES: Alejandro Egas, Adrián Lema, Ariel Muñoz y Jhonathan Pizarra

CURSO: 8vo

PARALELO: "A "

DESCRIPCIÓN: Avance Proyecto Final

ASIGNATURA: Tendencias Innovadoras de la Profesión

TEMA: Avance Análisis con Web Scraping

DESARROLLO:

Una de las aplicaciones más evidentes del web scraping es en el comercio electrónico, donde las empresas pueden utilizarlo para monitorear los precios de la competencia, recopilar información sobre productos y tendencias del mercado, y mantenerse al día con las opiniones de los clientes. Al extraer datos de múltiples fuentes, las empresas pueden obtener una visión más completa y granular del mercado en el que operan, lo que les permite tomar decisiones más informadas y estratégicas.

El presente proyecto plantea una comparativa entre las marcas más populares de comida rápida, como lo son KFC y Campero, dos industrias que tiene popularidad en sus platillos. El objetivo planteado es recopilar información sobre sus productos, girando en torno a sus precios para conocer la variedad de ofertas, así como las promociones y el restaurante con los precios más costosos y más económicos.

Análisis del proyecto

Función scrape_prices: Esta función realiza una solicitud HTTP a una URL dada, recupera el contenido de la página y utiliza expresiones regulares para encontrar y extraer los precios del contenido HTML

```
web scrap.py  prueba.py  ejemplo.py
prueba.py
37 def compare_prices(prices1, prices2):
2 import requests
3
4 def scrape_prices(url, pattern):
5     response = requests.get(url)
6     content = response.text
7     prices = re.findall(pattern, content)
8     return prices
9
```

Función convert_to_float: Esta función toma una cadena de precio como entrada y trata de convertirla a un número flotante. Si la conversión falla (por ejemplo, si la cadena no representa un número válido), devuelve None.

```
9
10 def convert_to_float(price):
11     try:
12         return float(price)
13     except ValueError:
14         return None
15
```



Recopilación de Precios de KFC y Campero: Se utilizan las funciones `scrape_prices` y `convert_to_float` para recopilar y convertir los precios de las páginas web de KFC y Campero.

```

16 # Recopilar precios de KFC
17 kfc_url = "https://www.kfc.com.ec/menu/combos-801AC979-D94D-E611-80CF-0050568B0200.html"
18 kfc_pattern = r"$\d+(?:\.\d+)?"
19 kfc_prices = scrape_prices(kfc_url, kfc_pattern)
20 kfc_prices = [convert_to_float(price[1:]) for price in kfc_prices if price]
21
22 print("Precios recopilados de KFC:")
23 for price in kfc_prices:
24     print(price)
25
26 # Recopilar precios de Campero
27 campero_url = "https://campero.ec/menu/promociones-a8aebdee-f014-4542-84c0-B050f47f9e29.html"
28 campero_pattern = r"$\d+(?:\.\d+)?"
29 campero_prices = scrape_prices(campero_url, campero_pattern)
30 campero_prices = [convert_to_float(price[1:]) for price in campero_prices if price and convert_to_float(price[1:]) > 0]
31
32 print("\nPrecios recopilados de Campero:")
33 for price in campero_prices:
34     print(price)
35

```

Filtrado de Precios para Campero: Antes de calcular el precio mínimo, se filtran los precios que son cero o negativos para asegurarse de que el precio mínimo sea mayor que cero.

```

36 # Comparar precios y variedad
37 def compare_prices(prices1, prices2):
38     unique_prices1 = set(prices1)
39     unique_prices2 = set(prices2)
40
41     print("\nComparación de precios:")
42     print("-" * 50)
43     print("KFC:")
44     print(f"Cantidad de precios únicos: {len(unique_prices1)}")
45     print(f"Precios únicos: {unique_prices1}")
46     print(f"Precio más económico: ${min(prices1)}")
47     print(f"Precio más caro: ${max(prices1)}")
48     print("-" * 50)
49     print("Campero:")
50     print(f"Cantidad de precios únicos: {len(unique_prices2)}")
51     print(f"Precios únicos: {unique_prices2}")
52     print(f"Precio más económico: ${min(prices2)}")
53     print(f"Precio más caro: ${max(prices2)}")
54     print("-" * 50)
55

```

Comparación de Precios y Variedad: Se calculan los precios únicos para cada negocio y se determina el precio más económico y el más caro utilizando las funciones `min()` y `max()`. Se comparan la cantidad de precios únicos y los precios promedio para determinar dónde hay más variedad de precios y qué negocio es más económico en promedio.

```

56 if len(unique_prices1) > len(unique_prices2):
57     print("KFC tiene más variedad de precios.")
58 elif len(unique_prices2) > len(unique_prices1):
59     print("Campero tiene más variedad de precios.")
60 else:
61     print("Ambos tienen la misma variedad de precios.")
62
63 avg_price1 = sum(prices1) / len(prices1) if prices1 else 0
64 avg_price2 = sum(prices2) / len(prices2) if prices2 else 0
65
66 if avg_price1 < avg_price2:
67     print("KFC es más económico en promedio.")
68 elif avg_price2 < avg_price1:
69     print("Campero es más económico en promedio.")
70 else:
71     print("Ambos tienen el mismo precio promedio.")
72
73 compare_prices(kfc_prices, campero_prices)

```

Resultado

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
17.99
15.99
19.99
16.5
19.5
18.99
21.5
27.5
26.0
8.0

Comparación de precios:
-----
KFC:
Cantidad de precios únicos: 16
Precios únicos: {2.99, 3.75, 4.99, 5.99, 6.25, 7.25, 6.99, 7.5, 5.75, 4.25, 12.99, 14.5, 14.99, 17.99, 25.99, 28.5}
Precio más económico: $2.99
Precio más caro: $28.5
-----
Campero:
Cantidad de precios únicos: 22
Precios únicos: {5.75, 6.5, 5.5, 8.0, 11.3, 14.99, 15.99, 16.5, 17.99, 18.99, 18.25, 19.99, 21.5, 21.99, 23.5, 23.99, 22.99, 25.99, 26.75, 19.5, 27.5, 26.0}
Precio más económico: $5.5
Precio más caro: $27.5
-----
Campero tiene más variedad de precios.
KFC es más económico en promedio.
```

BIBLIOGRAFÍA:

Lawson, R. (2015). Web Scraping with Python. Packt Publishing Ltd.

Mitchell, R. (2018). Web Scraping with Python: Collecting More Data from the Modern Web. O'Reilly Media.

McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.

