



# **Instituto Politécnico Nacional**

Unidad Profesional Interdisciplinaria de Ingeniería  
campus Coahuila

## **Reporte Proyecto Final Curso de Python**

William Orta

Docente(s): Ing. Jessica Sarahí Méndez /  
Ing. Cuauhtemoc Bautista

San Buenaventura, Coahuila a August 3, 2023

# 1 Introduccion

Este código tiene como objetivo obtener los precios de tres artículos de comercio electrónico desde sus respectivas URLs, mostrar los precios en forma de gráfica de barras y etiquetarlos con los nombres de los artículos. A continuación, se explica paso a paso la funcionalidad del código: contenidos...

## 2 Desarrollo

1. Se importan las librerías necesarias: - 'matplotlib.pyplot': Para crear la gráfica de barras y mostrarla. - 'bs4.BeautifulSoup': Para extraer información del contenido HTML de las páginas web. - 'requests': Para realizar solicitudes HTTP y obtener el contenido de las páginas web.

2. Se define la función 'get\_price(url)': - Esta función recibe una URL como parámetro y realiza los siguientes pasos: - Realiza una solicitud HTTP a la URL especificada usando 'requests.get(url)'. - Parte el contenido de la página web usando 'BeautifulSoup' y lo almacena en 'soup'. - Busca el primer elemento 'span' que tenga el atributo 'class' con el valor "‘andes-money-amount\_\_fraction’", que generalmente contiene el precio del artículo. - Extrae el texto dentro del elemento 'span' y lo guarda en 'price\_str'. - Reemplaza las comas en 'price\_str' para eliminar la separación de miles en el precio. - Convierte el precio en formato de texto a un número entero y lo retorna.

4. Se crea una lista con las iteraciones de precios = [get\_price(url) for url in urls]

- 'precios': Esta es la nueva lista que queremos crear, que contendrá los precios de los artículos. - 'get\_price(url)': Esta es la expresión que se aplica a cada elemento ('url') de la lista 'urls'. En este caso, se llama a la función 'get\_price()' pasando cada URL como argumento para obtener el precio del artículo correspondiente. - 'for url in urls': Esta es la parte de la lista de iteración que especifica el bucle. Para cada elemento 'url' en la lista 'urls', la expresión 'get\_price(url)' se evaluará y el resultado se agregará a la nueva lista 'precios'.

Entonces, en resumen, la recorre la lista de URLs ('urls'), llama a la función 'get\_price()' con cada URL para obtener el precio correspondiente, y construye una nueva lista llamada 'precios' que contiene los precios de los tres artículos. Al finalizar la ejecución de este código de iteración, la lista 'precios' contendrá los precios obtenidos de las URLs. 5. Se define la función 'imprime(precios, articulos)': - Esta función recibe dos listas como parámetros: 'precios' (que contiene los precios de los artículos) y 'articulos' (que contiene los nombres de los artículos). - Crea una gráfica

de barras utilizando `plt.bar(articulos, precios)`, donde los nombres de los artículos se utilizan como etiquetas en el eje x y los precios se representan en el eje y. - Se agregan etiquetas al eje x e y mediante `plt.xlabel('Artículos')` y `plt.ylabel('Precio')`. - Se agrega un título al gráfico con `plt.title('Precios de tres artículos')`. - Finalmente, se muestra la gráfica con `plt.show()`.

6. Se llama a la función `imprime(precios, articulos)` para mostrar la gráfica con los precios de los tres artículos. Los datos para la gráfica son obtenidos previamente con `precios = [get_price(url) for url in urls]`.

En resumen, este código es un ejemplo de cómo obtener los precios de productos desde páginas web utilizando la biblioteca `BeautifulSoup`, mostrarlos en una gráfica de barras utilizando `matplotlib` y etiquetarlos con los nombres de los artículos.

## 3 Código

### 3.1 Se importan librerías

Se importan las librerías necesarias para realizar peticiones al sitio en internet (`requests`) y para crear la gráfica (`matplotlib.pyplot`).

```
1 # Se importa libreria para dibujar
2 import matplotlib.pyplot as plt
3 # Se importan librerías para hacer peticiones a las páginas web y
  obtener su código
4 # , así como también para tratar los datos web obtenidos
5 from bs4 import BeautifulSoup
6 import requests
```

### 3.2 Función `'mameluco(url)'`

```
1 def mameluco(url):
2     # Manda consultar la página "url" y guarda el contenido en "page"
3     page = requests.get(url)
4     # Secciona el contenido de la página usando BeautifulSoup
5     soup = BeautifulSoup(page.content, 'html.parser')
6     # Crea una lista de todos los elementos que contienen las
      etiquetas "span" y el atributo "class": "andes-money-
      amount__fraction" con un límite de lista de 2 (limit=2)
7     change_tag = soup.find_all('span', attrs={"class": "andes-money-
      amount__fraction"}, limit=2)
8     # Extrae el segundo elemento de la lista (posición 1) y lo
      convierte a string
```

```

9  chango_tag1 = str(chango_tag[1])
10 # Crea un nuevo objeto BeautifulSoup a partir del código HTML que
    est en chango_tag1
11 sopa1 = BeautifulSoup(chango_tag1, 'html.parser')
12 # Obtiene el contenido que est dentro de las etiquetas span
13 chango_tag2 = sopa1.find('span', attrs={"class": "andes-money-
    amount__fraction"})
14 # Quita las comas del texto obtenido
15 price_str = chango_tag2.text.replace(',', '', '')
16 # Convierte el precio a un entero y lo retorna
17 return int(price_str)

```

La función `mameluco(url)` realiza los siguientes pasos:

Hace una solicitud a la página web proporcionada por `url` utilizando `requests.get(url)` y guarda el contenido en `page`. Utiliza `BeautifulSoup` para extraer el contenido de la página y lo guarda en `sopa1`. Busca todos los elementos que contengan la etiqueta `span` y el atributo `class` con valor `andes-money-amount__fraction`, limitando la búsqueda a 2 resultados (esto es para evitar obtener precios no deseados). Estos elementos se guardan en `chango_tag`. Extrae el segundo elemento de `chango_tag` (posición 1) y lo convierte a un string, almacenándolo en `chango_tag1`. Crea un nuevo objeto `BeautifulSoup` llamado `sopa1` utilizando el código HTML que está en `chango_tag1`. Utiliza `find()` para obtener el contenido que está dentro de las etiquetas `span` en `sopa1` y lo guarda en `chango_tag2`. Quita las comas del texto obtenido en `chango_tag2`. Convierte el precio a un entero y lo retorna.

### 3.3 Lista de URLs y obtención de precios

En esta lista se guardan las direcciones de donde obtendremos los precios (que en este caso serán tres).

```

1 # Lista de URLs de los artículos a analizar
2 urls = [
3 'https://www.mercadolibre.com.mx/realme-11-pro-plus-12gb-ram-512gb-
    rom-200mp-ois-dual-sim-verde-5000-mah-100w-pantalla-curva-120hz-
    fhd/p/MLM23438802?pdp_filters=item_id:MLM1922556715#
    is_advertising=true&searchVariation=MLM23438802&position=1&
    search_layout=stack&type=pad&tracking_id=97805d59-1549-48d8-a60a-
    8b8beea8d935&is_advertising=true&ad_domain=VQCATCORE_LST&
    ad_position=1&ad_click_id=
    ZjZjY2ZlZWEtY2YwNi00YWl0LWJkZWItMDE3OWQwNWlyOTY3',
4 'https://articulo.mercadolibre.com.mx/MLM-2286029578-celular-8849-
    tank-2-de-12gb-256gb-155000mah-con-proyector-_JM#polycard_client=
    bookmarks',

```

```

5 'https://www.mercadolibre.com.mx/realme-gt-2-pro-dual-sim-256-gb-
   steel-black-12-gb-ram/p/MLM19130693?pdp_filters=item_id:
   MLM1507221033#is_advertising=true&searchVariation=MLM19130693&
   position=9&search_layout=stack&type=pad&tracking_id=d7676704-bae0
   -4c80-8d25-29fa31f1b171&is_advertising=true&ad_domain=V
6

```

### 3.4 Funcion imprime(precios, articulos):

Con esta funcion graficaremos los precios en una grafica de tipo barras. La funcion recibe dos listas(la de precios, y la de articulos).

```

1 def imprime(precios, articulos):
2     # Crear la grafica de barras
3     plt.bar(articulos, precios)
4
5     # Etiquetas del eje x
6     plt.xlabel('Art culos')
7
8     # Etiquetas del eje y
9     plt.ylabel('Precio')
10
11    # T tulo del gr fico
12    plt.title('Precios de tres art culos')
13
14    # Mostrar el gr fico
15    plt.show()
16

```

### 3.5 Llamada de funcio y creacion de listas de precios y articulos.

Llamamos las funcion mameluco, y por medio de un "for" le pasamos de una a una las paginas contenidas en la lista "urls", y nos regresara la lista de precioss. Ademas de cargar las lista del los nombres de los articulos.

```

1 precios = [mameluco(url) for url in urls]
2 #print(precios)
3 # Se crea la lsita de aritculos
4 articulos = ['realme-11-pro-plus-', 'Celular 8849 Tank 2 ', 'GT 2
   Pro']

```

### 3.6 Llamamos la funcion "imprime(precios, articulos)" y le pasamos las dos listas.

Se llama a la funcion "imprime", que a su vez carga las dos listas hechas previamente(una por medio de la funcione "mameluco" y la otra de forma manual)

```
1 # Se manda a llamar la funcion "imprime" mandandole dos parametros:  
2 # precios y articulos  
3 imprime(precios, articulos)
```

### 3.7 Grafica de barras.

Al final obtenemos la siguiente grafica de barras.

