

Programación multiproceso

Programación asíncrona en Python con asyncio

Desarrollo de Aplicaciones Multiplataforma – PSP

Resultado de Aprendizaje y Criterios de Evaluación aplicados a esta práctica

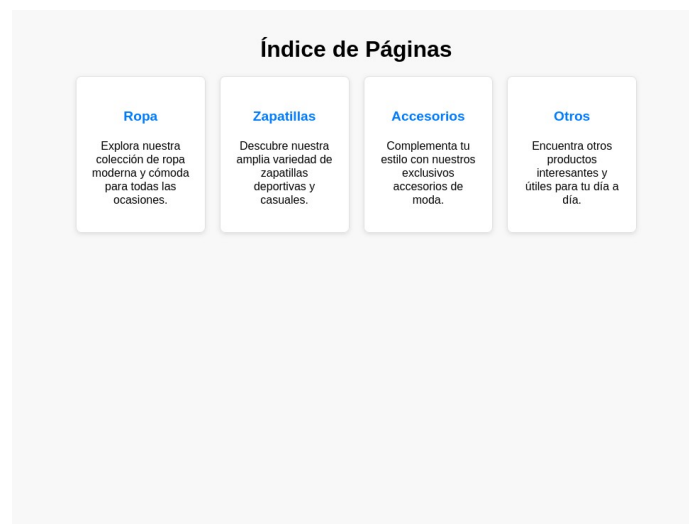
- RA1. Desarrolla aplicaciones compuestas por varios procesos reconociendo y aplicando principios de programación paralela.
 - a) Se han reconocido las características de la programación concurrente y sus ámbitos de aplicación.
 - i) Se han depurado y documentado las aplicaciones desarrolladas.

Descripción

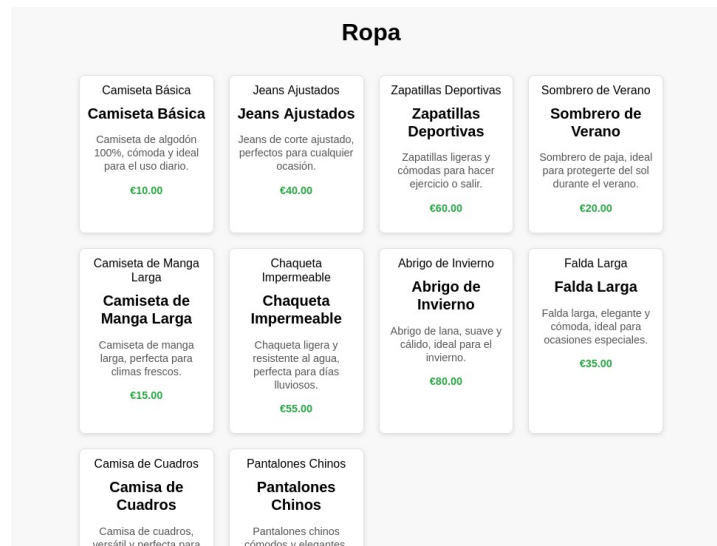
En esta práctica vas a escribir un crawler asíncrono en Python, es decir, un programa que recorra un sitio web y todas sus páginas web de forma automática para recabar cierta información, y todo esto de forma asíncrona.

La web que tiene que recorrer tu crawler te la doy lista para que la puedas dockerizar y poner en marcha en tu entorno de desarrollo.

Se trata de una tienda online de ropa y productos relacionados. Esta es la página web inicial:



Y cada una de esas categorías te lleva a otra página web del sitio web. Por ejemplo, aquí puedes ver la página de ropa:



Este sitio web podría crecer de dos maneras:

- Se podrían añadir nuevas secciones o categorías.
- En cada categoría se pueden añadir nuevos productos.

El programa que vas a escribir va a obtener toda la información del sitio web para mostrar por pantalla qué producto es el más caro, el más barato y qué precio medio tienen los productos, todo ello por categorías.

La salida por pantalla seguirá este formato:

Category Ropa

Max. Price: 80.0€ (Abrigo de invierno)

Min. Price: 10.0€ (Camiseta básica)

Avg. Price: 38.5€

Category Zapatillas

Max. Price: 175.0€ (New Balance 990)

Min. Price: 55.0€ (Converse All Star)

Avg. Price: 108.5€

Category Accesorios

Max. Price: 75.0€ (Reloj de Mano)

Min. Price: 10.0€ (Mascarilla de Tela)

Avg. Price: 35.2€

Category Miscelánea

Max. Price: 40.0€ (Lámpara LED)

Min. Price: 8.0€ (Taza de Café)

Avg. Price: 22.5€

Cómo ejecutar el programa

Tu programa debe constar de un script llamado `main.py` que será el que se ejecute usando el intérprete de Python 3.

Además, este script o módulo de Python esperará por la entrada en línea de comandos la URL del sitio web que tiene que consumir. Ejemplo:

```
python main.py http://localhost:8080
```

Si el programa no se invoca de esta manera, pasando la URL, tendrá que terminar con un error, indicando cómo se tiene que ejecutar.

Al margen del módulo `main.py` puedes crear otros que consideres oportuno pero el script inicial tiene que llamarse `main.py`.

Librerías o módulos de Python que puedes usar

Estas son las únicas librerías de Python que puedes usar para la realización de este programa:

- `sys`
- `asyncio`
- `aiohttp`
- `re`
- `lxml`

Todas estas librerías están en Python menos `aiohttp` y `lxml` que tendrás que instalar con PIP en tu entorno virtual de Python.

Entorno Virtual de Python

Como se ha comentado en clase, es habitual configurar un entorno virtual para tus programas de Python de manera que puedas ajustar las versiones de las librerías necesarias y, en definitiva, trabajar en un entorno controlado al mergen de tu sistema.

Entrega

Entrega tu proyecto de Python que, tan solo, tiene que incluir:

- Los ficheros `.py` que hayas escrito
- El fichero `requirements.txt` con las librerías instaladas.

Evidentemente, estos ficheros los puedes organizar en directorios pero, recuerda, que el fichero `main.py` y `requirements.txt` TIENEN que estar en la raíz del proyecto.