

ACCESO INTELIGENTE A LA INFORMACIÓN

EJERCICIO BEAUTIFULSOUP I

Vamos a realizar webscraping, haciendo uso de BeautifulSoup, sobre una web de ventas de vinos, concretamente, nos interesa la información de los vinos tintos:

<https://www.vinissimus.com/es/vinos/tinto/?cursor=0>

ALMACENAR LOS VINOS DE AL MENOS TRES PÁGINAS. CADA PÁGINA AUMENTA EL CURSOR DE LA URL EN 36

Buscamos construir un programa en Tkinter con un MENÚ con dos opciones:

a) “Datos”, con tres opciones:

- a. “Cargar”, que sea capaz de extraer y almacenar en una base de datos sqlite los distintos vinos. Para cada vino hay que almacenar: nombre del vino, precio (si tiene descuento, el precio es con el descuento), denominación de origen (Rioja, Ribera del Duero,...), bodega (Bodegas Emilio Moro,...) y tipos de uvas (Tempranillo, Garnacha,...). Mostrar una ventana de información indicando el número de vinos que se han almacenado en la base de datos.
- b. “Listar”, que muestre en una ventana (con una listbox con scrollbar) todos los vinos (nombre, precio, bodega y denominación de origen).
- c. “Salir”, que cierre la ventana principal.

b) “Buscar”, con tres opciones:

- a. “Denominación”, que muestre una spinbox que permita al usuario seleccionar una denominación, y muestre en otra ventana (en una listbox con scrollbar) todos los vinos de dicha denominación de origen (nombre, precio, bodega y denominación de origen).
- b. “Precio”, que muestre una ventana con un entry que permita al usuario introducir un precio, y muestre en otra ventana (en una listbox con scrollbar) todos los vinos con un precio inferior al indicado (nombre, precio, bodega y denominación de origen) ordenados por orden de precio ascendente.
- c. “Uvas”, que muestre una spinbox que permita al usuario seleccionar un tipo de uva, y muestre en otra ventana (en una listbox con scrollbar) todos los vinos (nombre, tipo y tipos de uvas) que contengan dicho tipo de uva en su composición.

IMPORTANTE: Para evitar errores SSL es conveniente incluir las siguientes líneas en el código

```
import os, ssl
if (not os.environ.get('PYTHONHTTPSVERIFY', '')) and
getattr(ssl, '_create_unverified_context', None)):
    ssl._create_default_https_context = ssl._create_unverified_context
```