

Control de Estructura de datos

Tópicos:

- consumo de api
- arreglos
- diccionarios

Para ello, vamos a trabajar con la página <https://randomuser.me/api> y <https://jsonplaceholder.typicode.com/users>

Pasos:

- 1.- Crear entorno virtual en python para instalar dependencias
- 2.- Instalar dependencias
- 3.- Construir la consulta a la api
- 4.- Crear el algoritmo que solucione los siguientes problemas

Desarrollo de los pasos

1.- Crear entorno virtual en python para instalar dependencias

- 1.- instalar el controlador de paquetes en python pip, con el siguiente comando en consola
`sudo apt install python3-pip`
- 2.- instalar el controlador de ambientes virtuales con el siguiente comando
`apt instal python3.8-venv`
- 3.- Crear una carpeta para el proyecto, en consola lo siguiente
`mkdir consulta-api`
- 4.- ingresar a la carpeta
`cd consulta-api`
- 5.- crear el ambiente vitrual
`python3 -m venv venv`

Con eso tenemos el ambiente virtual creado.

- 6.- Activar el ambiente virtual, para que el sistema reconozca que vamos a trabajar en ese lugar.
`source venv/bin/activate`

En la consola, debe aparecer como prefijo (venv), con esto estamos listos

```
(venv) gallo@gallo-pc:~/consulta_api_estructura$
```

2.- Crear dependencias en python

las dependencias son librerías de terceros que vamos a usar en nuestro proyecto.

1.- Crear un archivo llamado “requirements.txt”

```
touch requirements.txt
```

2.- Llenar el archivo requirements.txt por consola con

```
nano requirements.txt
```

Eso ingresa a una pantalla negra, donde debemos escribir lo siguiente

```
requests
```

Ahora, apretar control+o para guardar, luego enter para aceptar el guardado, y luego salir con control+x

3.- instalar dependencias con el siguiente comando en consola

```
python3 -m pip install -r requirements.txt
```

3.- Construir la consulta a la API

1.- abrir visual studio coda con el siguiente comando

```
code .
```

Revisar el siguiente tutorial

<https://j2logo.com/python/python-requests-peticiones-http/>

2.- escribir el siguiente código en un archivo .py

```
import requests
import json

# request.get hace una petición a alguna página web, le damos la página web por parámetro
response = requests.get('https://randomuser.me/api')

# para hacerlo más fácil de cambiar, podemos hacerlo así
# url = 'https://randomuser.me/api'
# response = requests.get(url)

# así, si queremos consultar en otro sitio web, solo cambiamos el valor de la variable url

# con esto, transformamos la respuesta web, que está en formato string, a un formato
# conocido por python
# transforma a diccionario y a arreglo cuando corresponda.
consulta = json.loads(response.text)

# De esta forma, "navegamos" por cada elemento de la respuesta web, en forma de
# diccionario y arreglo
print(consulta['results'][0]['name']['first'])
```

3.- analizar el código anterior

4.- Crear el algoritmo que solucione los siguientes problemas

1.- para la siguiente url = <https://randomuser.me/api>

Cree la función “show” que muestre por pantalla lo siguiente:

Si el usuario es mujer, que muestre por consola su nombre (first + last), su dirección (nombre de calle, dirección + ciudad)

Si el usuario es hombre, que muestre por consola su nombre (first + last), su username y su contraseña.

```
def show() => None
```

2.- para la siguiente url = <https://jsonplaceholder.typicode.com/users>

Cree la función “georeference” que, reciba como parámetro un número, y que devuelva un arreglo donde el primer elemento sea el nombre, el segundo elemento sea la latitud y el tercer elemento sea su longitud.

```
def georeference(n) => list
```

3.- para la siguiente url = <https://jsonplaceholder.typicode.com/comments>

Cree la función “coment” que recibe como parámetro un número, y devuelve como diccionario el comentario que tenga como atributo “id” el número que se agregó como parámetro:

```
def coment(n) => dict
```

Los algoritmos realizados, deben ser subidos a su perfil de github en el proyecto llamado “control 1 estructura”, y pegar el link de ello en el foro llamado “**Control 1**” que está en aula virtual.