

PROYECTO : API REST

ALEJANDRO RUIZ CRESPO

PROGRAMACIÓN

29/04/2024

Sumario

1.- ¿Que es una Api?.....	3
2.- Objetivo del proyecto.....	3
3.- Estructura básica.....	3
4.- Funciones.....	4
4.1.- Check_internet :.....	4
4.2.- Chatch_error :.....	5
4.3.- Search_character_by_code :.....	5
4.4.- Search_species_of_character :.....	6
4.5.- List_episode_characters :.....	7
4.6.- Residences_of_character :.....	8
4.7.- Character_residences :.....	9
4.8.- Character_info :.....	10
4.9.- Menu Principal:.....	11
5.- Web grafía de la API.....	12

1.- ¿Que es una Api?

Api(Application Programming Interface) es un conjunto de herramientas que permite a los desarrolladores acceder y utilizar las funciones y datos de un software o servicio de manera programática sin necesidad de conocer los detalles internos de como funciona el software o servicio.

2.- Objetivo del proyecto

Conocer mediante un ejercicio practico las funcionalidades que puede tener una API, y como usar estas para recolectar información de las paginas web que lo permitan.

3.- Estructura básica

La estructura básica que hemos usado para las peticiones a la API es la siguiente. Como primer paso, hacemos una solicitud GET a la URL de la API de Rick y Morty, que es la API que he usado.

```
response = requests.get(url)
```

Usamos la biblioteca request de python, que permiten hacer este tipo de solicitudes y guardar las respuestas. Con el método GET se le pasa como parámetro la URL que es asociado en cada función, y response es la variable que guarda la respuesta de la solicitud. Con esta variable se puede acceder al código de estado , al contenido de la página o a los encabezados entre otros.

```
if response.status_code == 200:
```

Aquí comprobamos que la solicitud se haya hecho correctamente y no ha dado ningún tipo de error, si no fuera así, se llama a una función que te dice el error.

```
response_json = response.json()
```

Aquí convierte la respuesta de la petición a formato JSON en un objeto de python, lo cual es mejor ya que es mucho mas legible para nosotros y podemos trabajar mas cómodamente .

```
print("\n Código de personaje: ", response_json["id"], "\nNombre Personaje: ", response_json["name"], )
```

Una vez ya con el JSON, mediante la estructura de este que esta compuesto de diccionarios, podemos buscar las respuestas deseadas teniendo la clave.

4.- Funciones

4.1.- Check_internet :

Esta función controla que tengamos internet antes de hacer una solicitud. Controlando a modo de futuro el posible error de conexión que nos daría en caso de no existir. Esto lo controlamos haciendo una petición básica a la URL de Google .La salida que ofrece es la siguiente:

Caso de conexión:

```
C:\Users\alexp\OneDrive\Escritorio\IDAW\pro  
Chequeando conexion a internet:  
Conexion a internet realizada con exito
```

Caso de no conexión:

```
Chequeando conexion a internet:  
No se ha podido conectar a internet  
  
Process finished with exit code 1
```

En caso de no tener conexión, se muestra el error y se termina el programa de forma anormal.

4.2.- Chatch_error :

Esta función es llamada cuando el código de estado es diferente a 200, que sería el código que devuelve cuando está todo correcto. Le pasa como parámetro el código de estado, y haciendo un match para los diferentes códigos de error que he capturado. Siendo el 400, 401, 403 y 404. Y te devuelve el mensaje del error correspondiente. Un ejemplo es el siguiente.

```
ValueError: El recurso solicitado no pudo encontrarse en el servidor
```

En el que le hemos pasado un código de personaje que no pertenece a ninguno.

4.3.- Search_character_by_code :

En esta función se pide un código de un personaje al usuario y te devuelve el nombre. Usando la estructura básica mostrada anteriormente y teniendo en cuenta, que hay varias páginas y la navegación entre estas en caso de que el código no sea encontrado. Salida que ofrece:

```
Dime el código del personaje: 567

Código de personaje: 567
Nombre Personaje: Michael
```

Comprobamos en la página web de la API que corresponde a ese nombre.

```
[
  {
    "id": 567,
    "name": "Michael",
    "status": "Alive",
    "species": "Mythological Creature",
    "type": "Dragon",
    "gender": "Male",
    "origin": {
      "name": "Dragon".
    }
  }
]
```

Coincide el nombre y demuestra la navegación entre páginas.

4.4.- Search_species_of_character :

Los personajes de la serie, pertenecen a una especie, esta función te saca todas las especies reconocidas de la serie. Al ser una característica de cada personaje, recorre todos estos y guarda la especie en una lista, la cual convierte después en un conjunto, para que no se repitan las especies. Tiene la siguiente salida:

```
Recogiendo datos...
1.-Mythological Creature
2.-Human
3.-Cronenberg
4.-Animal
5.-Robot
6.-Disease
7.-Alien
8.-Humanoid
9.-unknown
10.-Poopybutthole
```

4.5.- List_episode_characters :

Esta función muestra el nombre de los personajes que salen en el episodio que el usuarios introduzca. Principalmente pide la temporada y el episodio y le da el formato necesario para buscarlo luego, esta no busca en los personajes, si no en la url de episodios, los cuales tienen como atributo una lista con la url de todos los personajes que salen en el episodio, de esta url cogemos el código y usando la primera función del programa, nos devuelve una lista de nombres que son los pertenecientes al episodio. Lista de personajes de un episodio:

```
{
  "id": 9,
  "name": "Something Ricked This Way Comes",
  "air_date": "March 24, 2014",
  "episode": "S01E09",
  "characters": [
    "https://rickandmortyapi.com/api/character/1",
    "https://rickandmortyapi.com/api/character/2",
    "https://rickandmortyapi.com/api/character/3",
    "https://rickandmortyapi.com/api/character/4",
    "https://rickandmortyapi.com/api/character/5",
    "https://rickandmortyapi.com/api/character/88",
    "https://rickandmortyapi.com/api/character/192",
    "https://rickandmortyapi.com/api/character/240",
    "https://rickandmortyapi.com/api/character/243",
    "https://rickandmortyapi.com/api/character/251",
    "https://rickandmortyapi.com/api/character/272",
    "https://rickandmortyapi.com/api/character/307",
    "https://rickandmortyapi.com/api/character/419",
    "https://rickandmortyapi.com/api/character/420",
    "https://rickandmortyapi.com/api/character/421",
    "https://rickandmortyapi.com/api/character/422",
    "https://rickandmortyapi.com/api/character/826"
  ],
  "url": "https://rickandmortyapi.com/api/episode/9",
  "created": "2017-11-10T12:56:34.645Z"
},
```

Buscando ese episodio el cual nos debería dar una lista de 17 personajes, nos da esta salida:

```
Dime la temporada del episodio: 1
Dime el episodio de la temporada: 9
1.-Rick Sanchez
2.-Morty Smith
3.-Summer Smith
4.-Beth Smith
5.-Jerry Smith
6.-Cynthia
7.-King Flippy Nips
8.-Mr. Goldenfold
9.-Mr. Needful
10.-Nancy
11.-Principal Vagina
12.-Scroopy Noopers
13.-Plutonian Hostess
14.-Plutonian Host
15.-Rich Plutonian
16.-Rich Plutonian
17.-Butter Robot
```

4.6.- Residences_of_character :

Al igual que las especies, la serie tiene distintos lugares de residencia donde estan atribuidos a personajes. Esta funcion te devuelve la lista completa de las ubicaciones, teniendo en cuenta que son varias páginas, te va mostrando de 20 en 20, preguntandote si quieres ir a la siguiente página. Este es la salida de la función:

Id	Nombre Residencia	Tipo Residencia
1	Earth (C-137)	Planet
2	Abadango	Cluster
3	Citadel of Ricks	Space station
4	Worldender's lair	Planet
5	Anatomy Park	Microverse
6	Interdimensional Cable	TV
7	Immortality Field Resort	Resort
8	Post-Apocalyptic Earth	Planet
9	Purge Planet	Planet
10	Venzulon 7	Planet
11	Bepis 9	Planet
12	Cronenberg Earth	Planet
13	Nuptia 4	Planet
14	Giant's Town	Fantasy town
15	Bird World	Planet
16	St. Gloopy Noops Hospital	Space station
17	Earth (5-126)	Planet
18	Mr. Goldenfold's dream	Dream
19	Gromflom Prime	Planet
20	Earth (Replacement Dimension)	Planet
Continuar listando [S/N]		

De cada ubicación, te vuelve el ID , el nombre de residencia y el tipo de residencia.

4.7.- Character_residences :

Similar a la de los episodios, esta devuelve los personajes que residen en la misma ubicación, siguiendo el mismo procedimiento para devolver los nombres de los personajes, un ejemplo, sería buscar "Anatomy Park":

JSON de la página web:

```
{
  "id": 5,
  "name": "Anatomy Park",
  "type": "Microverse",
  "dimension": "Dimension C-137",
  "residents": [
    "https://rickandmortyapi.com/api/character/12",
    "https://rickandmortyapi.com/api/character/17",
    "https://rickandmortyapi.com/api/character/96",
    "https://rickandmortyapi.com/api/character/97",
    "https://rickandmortyapi.com/api/character/98",
    "https://rickandmortyapi.com/api/character/99",
    "https://rickandmortyapi.com/api/character/100",
    "https://rickandmortyapi.com/api/character/101",
    "https://rickandmortyapi.com/api/character/108",
    "https://rickandmortyapi.com/api/character/268",
    "https://rickandmortyapi.com/api/character/300"
  ],
  "url": "https://rickandmortyapi.com/api/location/5",
  "created": "2017-11-10T13:08:46.060Z"
},
```

La salida de la función debería dar el nombre de los 11 personajes que se ubican en Anatomy Park:

Dime la residencia que quieras saber: *Anatomy Park*

- 1.- Alexander
- 2.- Annie
- 3.- Tuberculosis
- 4.- Gonorrhea
- 5.- Hepatitis A
- 6.- Hepatitis C
- 7.- Bubonic Plague
- 8.- E. Coli
- 9.- Dr. Xenon Bloom
- 10.- Poncho
- 11.- Roger

4.8.- Character_info :

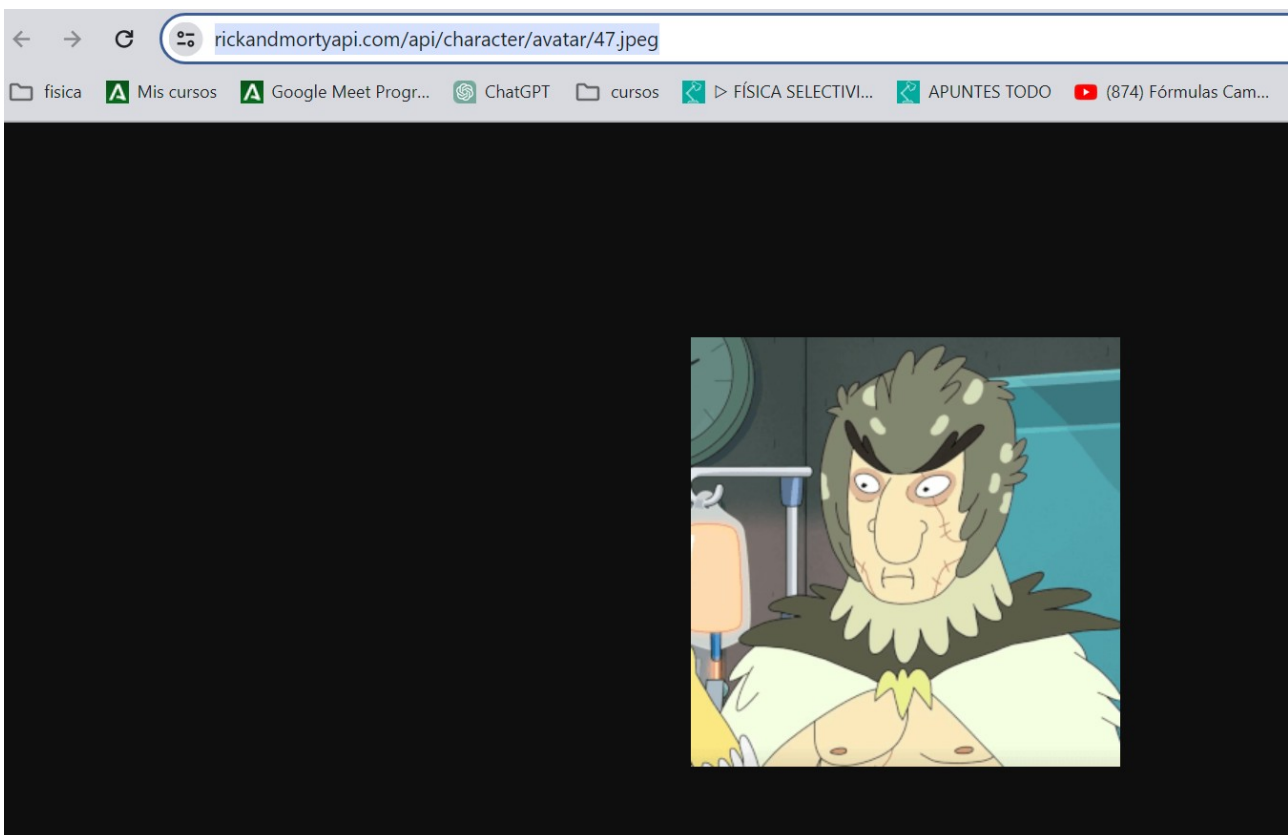
Esta última función, te muestra las características principales de un personaje que usuario pide, una vez obtenidos, da la opción de ver una imagen del personaje en el navegador.

La salida es la siguiente:

```
Dime el nombre del personaje: Birdperson

Código de personaje: 47
Nombre Personaje: Birdperson
Status: Alive
Nombre Genero: Male
Especie: Alien
Quieres ver una imagen del personaje (S/N) : s
```

Dando respuesta 'S', nos lleva a la url de la imagen.



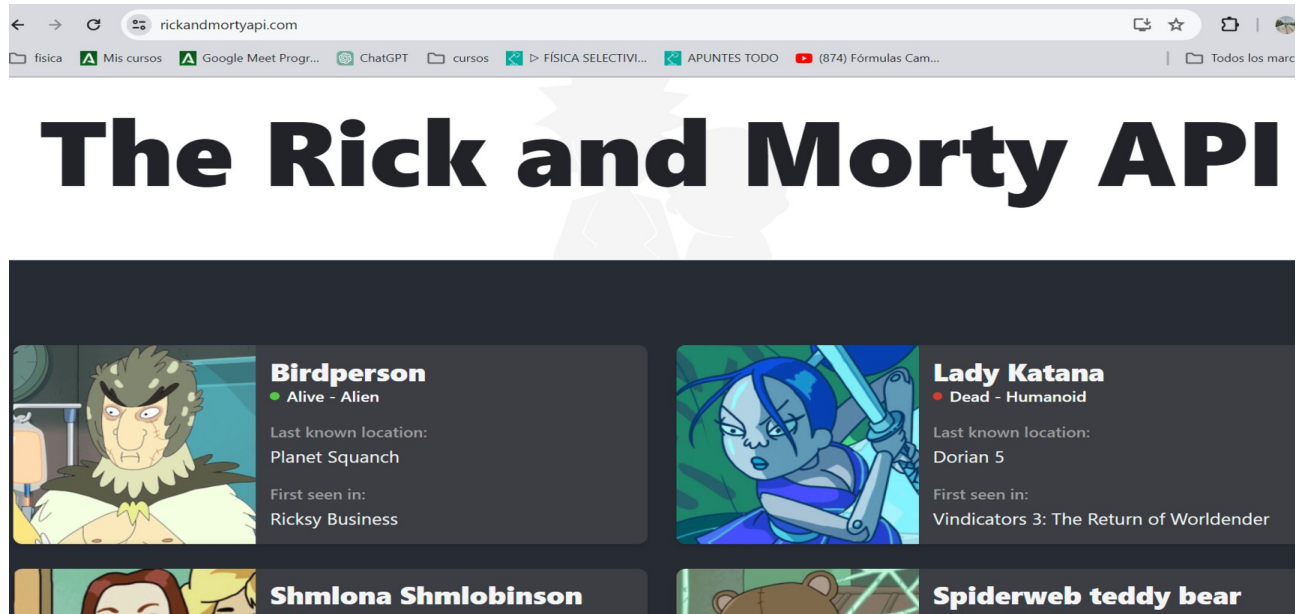
4.9.- Menu Principal:

```
RICK Y MORTY
-----
1. Busca nombre de un personaje por su código.
2. Lista todas las especies.
3. Listar los personajes que salen en un episodio.
4. Listas todos los sitios de residencia.
5. Lista todos los personajes que residen en el mismo sitio.
6. Información de un personaje por su nombre.
0. Salir.
```

Este menu esta en la función main, el cual llama primero a la función para chequear le internet, y una vez este nos de el aprobado, muestra el menu del que puedes escoger las diferentes seis opciones, en el caso de querer salir, seleccionar la opción 0.

5.- Web grafía de la API

Pagina Principal : <https://rickandmortyapi.com/>



Deriva en 3 url:

La url de los personajes : <https://rickandmortyapi.com/api/character>

La url de las localizaciones : <https://rickandmortyapi.com/api/location>

La url de los episodios : <https://rickandmortyapi.com/api/episode>