

April 29, 2023

1 Fundamentos de Programación

1.1 PAC 5 - Enunciado

En este Notebook se encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podréis sacar la máxima nota de la PAC sin necesidad de hacer este ejercicio. El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Veréis que todas las actividades de la PEC tienen una etiqueta, que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- **NM Sólo materiales:** las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- **EG Consulta externa guiada:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- **EI Consulta externa independiente:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, podéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

1.2 Ejercicios para la PEC

A continuación encontraréis los ejercicios que se deben completar en esta PEC y que forman parte de la evaluación de esta unidad.

1.2.1 Ejercicio 1

La respuesta recibida después de realizar una petición a una web API http es un objeto que contiene, entre otros, los siguientes atributos: **status.code**, **content** y **headers**. Describe qué información contiene cada uno de los atributos anteriormente enumerados y pon un ejemplo de cada uno.

Recordad que hay que citar las referencias consultadas para responder la pregunta, y que la respuesta que proporcionéis debe ser original (redactada por vosotros mismos, después de haber leído y entendido las referencias que consideréis oportunas).

(1 punto) NM

[21]: # Respuesta

1.2.2 Ejercicio 2

Implemente una o varias funciones para obtener la **secuencia de ADN** del organismo *Homo sapiens* del cromosoma mitocondrial (**chrM**) desde la posición 10000 hasta 10050 por la referencia **hg38**.

Para ello utilice la API de [UCSC](#).

Nota: El genoma de una célula es un repositorio de secuencias de ADN (ácido desoxiribonucleico) empaquetado en forma de cromosoma. El ADN es un ácido nucleico que contiene la información genética que dirige el desarrollo y funcionamiento de todos los seres vivos. El ADN puede entenderse como una secuencia de nucleótidos (A,C,T y G) de cierta longitud. Este material hereditario codifica los genes que, una vez descifrados, son indispensables por la síntesis de las proteínas.

Un genoma de referencia es la representación de la secuencia de ADN del genoma de una especie. En el caso del organismo *Homo sapiens*, existen diferentes versiones del genoma de referencia. La última versión, hg38, se publicó en 2014 y es la más detallada y precisa.

UCSC es un navegador de la Universidad de Santa Cruz de California que ofrece acceso a datos de secuencias de genomas y su correspondiente anotación (genes, mRNAs, CpG,...) de una gran variedad de organismos, vertebrados e invertebrados.

Referencia: [Genómica Computacional](#). Enrique Blanco. Barcelona, Universidad Abierta de Cataluña, 2011.

Importante: No es necesario entender toda la información que podemos obtener a través de la API del UCSC. Fíjese con lo que le pide el enunciado (prestando especial atención a las palabras clave en negrita), y revise los ejemplos de acceso a datos que hay en la web de la [UCSC](#).

(1,5 puntos) EG

[]: # Respuesta

1.2.3 Ejercicio 3

Programa una función que devuelva el estado meteorológico actual a partir de una localización dada, especificada por su código postal **zip code** y código de país (e.g: us, uk, es, fr, etc).

La función debe devolver lista de tuplas de dos elementos, correspondientes al resumen del estado actual del tiempo **weather.main** ya la descripción extendida **weather.description**.

Utiliza la API de [openweathermap](#) para obtener las predicciones.

Para poder utilizar la API necesitarás registrarte y obtener una API key. Puedes registrarte [aquí](#) y obtener tu API key [aquí](#) una vez registrados.

Tened en cuenta que la API key puede tardar un tiempo en funcionar una vez registrados, y que durante este tiempo la llamada puede dar un error 401 conforme la clave no es válida:

```
{"cod":401, "message": "Invalid API key. Please see http://openweathermap.org/faq#error401 for more info."}
```

Pistas:

- Veréis que por lo general la API está documentada sin incluir la API Key, pero esta es necesaria. Deberéis incluir la API key en la llamada como uno de los parámetros de la URL (&appid=your_api_key):

```
http://example_url.com?param1=value1&param2=value2&appid=your_api_key
```

- Os animamos a que sigáis el proceso de registro para aprender de qué se trata y cómo se generan API Keys. Os proporcionamos una API Key de ejemplo por si tenéis problemas con el proceso.

```
own_api_key = 'd54f26dbcf6d4136bc0ef8ba5f07825b'
```

(1.5 puntos) EG

[25]: # Respuesta

b) Encuentra el tiempo para estos códigos postales de España.

```
cps = ["11380", "38111", "48003", "08800"]
```

(0.5 puntos) NM

[]: # Respuesta

1.2.4 Ejercicio 4

Para cada una de las siguientes direcciones:

1. <https://zombo.com>
2. <https://www.pixelsfighting.com>

3. <http://hasthelargehadroncolliderdestroyedtheworldyet.com>
4. <https://api.fbi.gov/wanted/v1/list>

Llama a cada dirección con `query.get()`, y muestra:

- El código de `status.code`.
- Si aplica, el contenido de la dirección web, limitando la salida hasta 500 caracteres

Accede a cada una de las direcciones con tu ordenador y observa el contenido en el navegador con el texto de la web. ¿Cuál es la diferencia en el contenido de la cuarta dirección respecto a las tres primeras?

(1.5 puntos) NM

[]: `# Respuesta`

1.2.5 Ejercicio 5

En este ejercicio lucharemos un poco contra el aburrimiento. Hay una API para justamente esto en la dirección <https://www.boredapi.com>, podéis ver la documentación de la API en <https://www.boredapi.com/documentation>.

Por ejemplo, dada una llamada a esta dirección, obtenemos una actividad de forma aleatoria:

Call | Description |

|: _____: |: _____ | |
<http://www.boredapi.com/api/activity/> | Random activity |

- a) Usa requests para llamar a la dirección anterior 100 veces. ¿Cuántas actividades aparecen para dos o más participantes?

(1.5 puntos) NM

[]: `# Respuesta`

- b) La documentación de la API indica que es posible hacer búsquedas incluyendo parámetros, como por ejemplo vía la ruta:

`/api/activity?minprice=:minprice&maxprice=:maxprice`

Escribe una función llamada `fun` que:

1. Tenga dos parámetros de entrada (`minprice`, `maxprice`).
2. Devuelva un diccionario solo con los elementos `price`, `activity` y `type`.

Llama a esta función otras 100 veces y comprueba que todos los precios se ajustan a los valores de la llamada.

(1 punto) NM

[7]: `# Respuesta`

1.2.6 Ejercicio 6

En este ejercicio haremos un poco de *Fun with numbers*. Hay una API para *number-facts* (curiosidades sobre números) en la base de datos <http://numbersapi.com>. Esta API tiene unos puntos de acceso en los que podemos interrogar la API sobre diferentes curiosidades sobre números enteros y fechas:

Dado un número N , o una fecha con día D y más M , la llamada a las siguientes direcciones proporcionan la siguiente información:

Call	Description
Http: //numbersapi.com/N	Trivia about N
Http: //numbersapi.com/N/math	Mathematical facts about N
Http: //numbersapi.com/M/D/date	Facts about a particular date
Http: //numbersapi.com/N/year	Facts about a particular date

Así, para obtener el *fact* número 1, debemos construir una solicitud a la *url*:

- `http: //numbersapi.com/1`

El objeto que se nos devolverá, contendrá la información indicada en la tabla en formato de texto directo.

- a) Construye una función que tenga de argumento dos año AI y AF . La función debe recoger los *facts* desde el año AI hasta el año AF ambos incluidos, y devolver un diccionario donde las claves sean el año y los valores sean el *fact* sobre ese año.

(1 punto) NM

```
[ ]: # Respuesta
```

- b) Usa esta función para mostrar por pantalla los *facts* entre 1915 y 1925, ambos incluidos.

(0.5 puntos) NM

```
[ ]: # Respuesta
```

1.2.7 Ejercicio Opcional

El New Mexico Tech Seismological Observatory almacena un registro de eventos sísmicos recientes. Este registro está disponible, por ejemplo en esta web.

- <https://geoinfo.nmt.edu/nmtso/events/home.cfml>

Usa **scrapy** para mostrar la información del listado por pantalla de las “Date+Time (UTC)” de cada evento de la página.

Para ello:

- Utiliza el tutorial de Scrapy para encontrar un **xpath** que contenga la información requerida.
- Muestra la información requerida en forma de diccionario.

EI

Nota: si la ejecución del *crawler* os devuelve un error `ReactorNotRestartable`, reiniciad el núcleo del Notebook (en el menú: **Kernel - Restart**). **Nota:** si scrapy no está instalado, lo podéis instalar vía

```
!pip install scrapy
```

```
[ ]: # Respuesta
```