PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2233 - PROGRAMACIÓN AVANZADA

Actividad 14

1^o semestre 2018
21 de junio

RegEx & web services

Introducción

¡Peligro! El nuevo ciber-hotel DCCommodity de la reina Barros, debido a la competencia ofrecida por XABBO Hotel, ha registrado pérdidas monumentales. Para peor, sus ~~ayudantes~~ consejeros financieros han tomado todos los fondos y los han invertido en memes obsoletos, con cero valor en el meme-mercado. Pero cuentan los rumores que en las próximas semanas la moneda virtual *Memecoin* subirá fuertemente de valor, momento ideal para vender los memes y salvar el hotel. Lamentablemente sus consejeros financieros, creyéndose sumamente ingeniosos han enmarañado los memes, y ahora se encuentran desaparecidos estudiando los exámenes. Ahora todo cae en manos de los estudiantes de *Programación avanzada*, quienes deberán rearmar y generar los memes para salvar el día.

Instrucciones

Parte 1: RegEx

Para salvar el club virtual, deberás primero arreglar el archivo `memes.txt`. El formato que se debe obtener es `'id imagen, texto superior, texto inferior'`, el problema es que las comas desaparecieron! Los consejeros financieros las reemplazaron por unas extrañas expresiones matemáticas que cumplen las siguientes reglas:

- Las expresiones matemáticas se encuentran entre dos símbolos $.
- Las expresiones puede contener cualquier número y símbolo de operaciones básicas (+, -, *, /, =).
- La cantidad y orden de los caracteres dentro de la expresión no están definidos (por eso son extrañas).

Un ejemplo de una de estas expresiones matemáticas es $5626+*0+864307757$. A la hora de implementar esta parte debes hacerlo creando la función `arreglar_memes` que debe escribir el resultado en un archivo llamado `arreglados.txt`.

Además, deberás crear la función `validar_correo` que reciba un correo y retorne `True` si es válido o no. Primero debe chequear a través de **RegEx** y luego a través de un *request* HTTP a la dirección del correo (recuerda agregar el prefijo correspondiente al protocolo), la cual debe retornar un código de estado válido (menor a 400, puedes revisar qué significa cada uno en los contenidos), retornando `True` solo si estas dos condiciones se cumplen.

Para esta parte sólo está permitido usar las funcionalidades que la librería `re` les proporciona, **no** deben usar manejo de *strings*.

Parte 2: *Web services*

Para esta actividad deberán usar la API de Imgflip (`https://api.imgflip.com/`) que les permitirá hacer solicitudes `GET` y `POST` para obtener los memes más populares y crear memes personalizados, respectivamente. Deberán:

- Crear la función `obtener_mejores` capaz de obtener los memes más populares y elegir aleatoriamente 10 de estos memes para ser mostrados.
- Crear la función `generar_meme` que dados el id de un meme, texto superior e inferior cree y retorne el meme creado.

El programa no debe mostrar directamente los memes, solo basta con entregar el URL para poder verlos en el navegador.

API

La API de imgflip tiene dos posibles solicitudes

- `GET`: llamando a ***https://api.imgflip.com/get_ memes*** sin necesidad de variables, se retornarán 100 memes populares aleatorios. La respuesta será de la forma:

```
{"success": true,
"data": {
 "memes": [
        {"id": "61579",
            "name": "One Does Not Simply",
            "url": "http://i.imgflip.com/1bij.jpg",
            "width": 568,
            "height": 335},
        { "id": "101470",
            "name": "Ancient Aliens",
            "url": "http://i.imgflip.com/26am.jpg",
            "width": 500,
            "height": 437},
            ...
    ]
}
}
```

- `POST`: llamando a ***https://api.imgflip.com/caption_ image*** este retornará el enlace al meme generado según las siguientes llaves esenciales:

Llave	**Significado**
template_id	ID de la imagen
username	Nombre de usuario en imgflip
password	Contraseña de imgflip
text0	Texto de arriba del meme
text1	Texto de abajo del meme

Una respuesta exitosa será de la forma:

```
{
    "success": true,
    "data": {
        "url": "http://i.imgflip.com/123abc.jpg",
        "page_url": "https://imgflip.com/i/123abc"
    }
}
```

- **Importante: Les recomendamos no usar *loops* para hacer *requests* a la API. Esta los podría bloquear y no podrán seguir usándola**.

Notas

- **Debes crear un archivo `config.py` en el que guardarás tus credenciales para importarlas al programa, este archivo no debe ser subido a GitHub. ¡Protege tus datos!**.
- La función `sample` de la librería `random` puede ser de utilidad.
- Puede que regex101 les sirva.
- Los memes de `memes.txt` pueden no tener texto superior o inferior.

Requerimientos

- (2.50 pts) RegEx.
  - (1.50 pts) Arreglar `memes.txt`.
  - (1.00 pts) Validar correo mediante RegEx.
- (3.50 pts) Web-Services.
  - (0.50 pts) Validar correo mediante `request`.
  - (1.50 pts) Solicitud de memes más populares utilizando GET.
  - (1.50 pts) Solicitud de meme con texto utilizando POST.

Entrega

- **Lugar:** En su repositorio de GitHub en la **carpeta** `Actividades/AC14/`
- **Hora:** 16:00