



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 — Programación Avanza (II/2017)
Tarea 7

1. Entrega

- Tarea
 - **Fecha/hora:** 27 de noviembre del 2017, 23:59 horas.
 - **Lugar:** GitHub – Carpeta: Tareas/T07/
- README.md
 - **Fecha/hora:** 28 de noviembre del 2017, 23:59 horas.
 - **Lugar:** GitHub – Carpeta: Tareas/T07/

2. Objetivos

- Interacción con APIs para la creación de un programa.
- Estudiar y utilizar una documentación de diferentes API.
- Generar *logs* utilizando el módulo `logging`.
- Sintetizar conocimientos del curso.

3. Introducción

Después de entregar tu tarea 6 de Avanzación Programada, te das cuenta que todo esto fue un plan malvado de Ainavi y sus secuaces Liben y Sacul para hacerte sufrir durante todo el semestre. A raíz de esto, planeas tu venganza e irás a buscarlos y hacerlos programar (en Visual Basic y PHP) para que sufran. Oportunamente, escuchaste a un ayudante decir que tus ahora tres enemigos mortales, son usuarios frecuentes de Twitter y les encanta publicar estados, comentando lo entretenido que es para ellos torturar a sus estudiantes. Gracias a un *hackeo* anónimo, sus últimas ubicaciones desde donde han *twitteado* han sido registradas en una hoja de Google Sheets. Por lo tanto, se te ocurre la brillante idea de ir a dichas localizaciones y consultar a la persona que encuentres para que te entregue información sobre tus enemigos.

4. Problema

En esta tarea debes crear una aplicación que consuma las API de **Google Sheets**, **Google Maps** y **Twitter**, para lograr buscar las localizaciones de ciertos lugares y mostrar estadísticas de los *tweets* de la zona. Todo esto tiene que ser con la interacción de una interfaz gráfica (ver sección 7).

5. Especificaciones

5.1. API

Como se mencionó anteriormente, para realizar esta tarea tendrás que usar la API de **Google Sheets**, **Google Maps** y **Twitter**. Una API (del inglés, *application programming interface*) consiste en un conjunto de recursos que permiten acceder a funcionalidades de alguna plataforma en particular, de forma amigable para ser utilizada por otros programadores. Como se pueden imaginar, es de vital importancia que los clientes que consumen alguna API —es decir, los programas desarrollados por ustedes— sepan exactamente cuáles parámetros deben entregar al realizar una solicitud, así como también se espera que lean correctamente las respuestas de las solicitudes de la API. Por ello, es importante que estudien y comprendan la documentación de las diferentes API que se usarán para la tarea. En particular, recomendamos fuertemente los siguientes enlaces para que se familiaricen con la estructura de la documentación:

- [Google Sheets](#)
- [Google Maps](#)
- [Twitter](#)

De necesitar credenciales, estas deben ser leídas a través de un archivo `config.py`. Para que el usuario fácilmente pueda poner sus credenciales personales para el correcto funcionamiento del programa.

Es importante mencionar que, con el fin de que no se colapsen los servidores de una empresa, todas las API tienen un número limitado de *requests* que un usuario puede hacer por intervalo de tiempo. Luego de ese lapso de tiempo, se reinicia el número de peticiones que se le puede hacer. Debes tener cuidado de no caer en un ciclo infinito y hacer todas las *requests* permitidos al mismo tiempo, pues no podrás seguir haciéndolos hasta que transcurra el intervalo de tiempo y se reinicien.

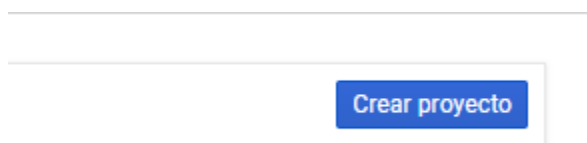
6. Funcionalidades

6.1. Recolectar datos de países y ciudades (30 %)

Dada una planilla de Google Sheets con el siguiente formato:

1	País	Ciudad 1	Ciudad 2	...	Ciudad N
2	Francia	Lyon	Paris
3	Chile	Rancagua	Concepción
4	Argentina	Buenos Aires	Mendoza
5

Tu programa deberá ser capaz de recolectar datos de países desde una planilla de Google Sheets creada por ti mismo/a. En particular, la planilla guardará una lista de países y una lista de ciudades para cada uno de estos países. El formato de la plantilla se muestra en la siguiente imagen:



Estos datos deberán ser obtenidos a través de la **API de Google Sheets**, que necesita un *access token* para poder utilizarla. Para obtenerlo, deben acceder al [sitio de desarrolladores](#) y crear un nuevo proyecto

(botón en la esquina superior derecha).

Luego de crear el proyecto, debes habilitar la API de Google Sheets para tu proyecto:


Panel

 [HABILITAR LAS API Y LOS SERVICIOS](#)

No hay API o servicios habilitados

Explora la [Biblioteca](#) para encontrar y utilizar cientos de API y servicios que están disponibles

Después de habilitar la API, dirígete al menú izquierdo y entra en “Credenciales”. En el menú de arriba, selecciona “Pantalla de consentimiento OAuth”, dale un nombre al proyecto y selecciona “Guardar”. Luego selecciona “Crear credenciales” y crea un “ID de cliente OAuth”:

 **Crear ID de cliente**

Tipo de aplicación

- ☐ Aplicación web
- ☐ Android [Más información](#)
- ☐ App de Chrome [Más información](#)
- ☐ iOS [Más información](#)
- ☐ PlayStation 4
- ☒ Otro

Nombre

T07|

Crear

Cancelar

Cierra la pantalla de diálogo presionando “Aceptar”. Luego, dirígete al ID de cliente recién creado y haz clic sobre él.

Credenciales Pantalla de consentimiento de OAuth Verificación de dominio

Crear credenciales ▼ Borrar

Crea credenciales para acceder a tus API habilitadas. [Consulta la documentación de la API](#) para obtener más detalles.

ID de clientes OAuth 2.0

<input type="checkbox"/> Nombre	Fecha de creación ▼	Tipo
<input type="checkbox"/> T07	18 nov. 2017	Otro

Selecciona la opción que dice “Descargar JSON”:

← ID de cliente de Otro [DESCARGAR JSON](#) [RESTABLECER SECRETO](#) [BORRAR](#)

ID de cliente	596727604976-j65l7ml90rick7tnav8coc9k8tspkrt7.apps.googleusercontent.com
Secreto de cliente	WsZFZgsz_q15KUQzPga50XWs
Fecha de creación	18 nov. 2017 21:04:54

Nombre

T07

Guardar Cancelar

Cámbiale el nombre al archivo descargado y nómbralo `client_secret.json`. Ahora ya estás listo para utilizar el ejemplo proporcionado en [este enlace](#).

6.2. Buscar coordenadas de localizaciones (10 %)

Una vez obtenidos los países y ciudades de la planilla, deberán obtener las coordenadas de cada una de estas ciudades. Para hacer esto, se debe utilizar la **API de Google Maps**. Se necesita una *API key* para poder hacer esta solicitud, por lo que se debe crear una cuenta y solicitarla, en el sitio de desarrolladores.

En este caso, estamos en *web-services*, por lo que se pedirá una clave en esa sección. Para esto, es necesario crear un proyecto (puede ser el mismo que usarán para Sheets) en <http://console.developers.google.com> y después de eso solicitar la clave (*API key*) para el servicio adecuado. Con eso será suficiente para realizar las solicitudes necesarias para buscar las coordenadas de localizaciones.

Google Maps API Página principal Documentación Tarifas y planes TODOS

Servicios web

Implementa API altamente escalables en el servidor para cualquier app o sitio web.

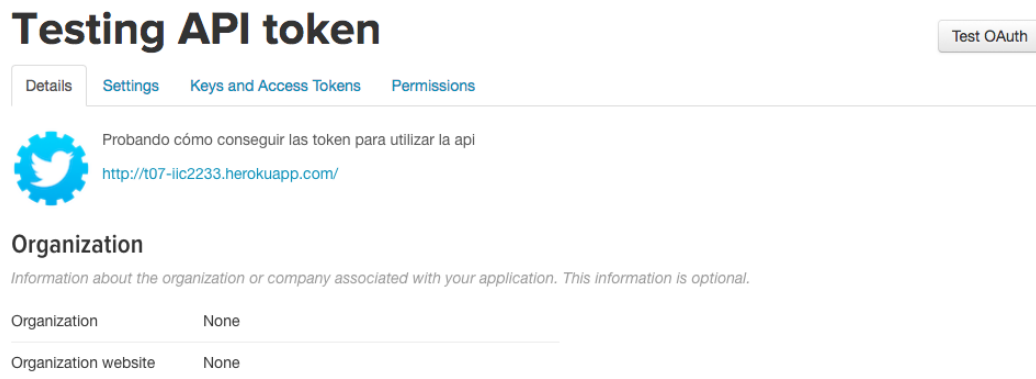
[OBTÉN UNA CLAVE](#)

INFORMACIÓN GENERAL GUÍAS SOPORTE

Mediante esta API se debe obtener la latitud y longitud de un nombre de ciudad o país.


6.3. Estadísticas de Twitter (30 %)

A continuación, debes utilizar la **API de Twitter**. Debes crearte una cuenta de Twitter si es que no tienes una. Luego, debes ingresar al siguiente enlace: <https://apps.twitter.com/> y seleccionar **Create New App**. Tendrás que llenar un formulario simple, con un nombre que elijas para la aplicación y una descripción; en el campo **Website** puedes ingresar el enlace <http://t07-iic2233.herokuapp.com/>. Una vez que hayas registrado tu aplicación, debes seleccionar la pestaña **Keys and Access Tokens**, como se muestra en la siguiente figura:



Testing API token Test OAuth

Details Settings **Keys and Access Tokens** Permissions

 Probando cómo conseguir las token para utilizar la api
<http://t07-iic2233.herokuapp.com/>

Organization
Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Una vez que ingreses ahí, debes bajar hasta el botón que indica **Create my access token**. Al presionarlo, te asignarán las siguientes claves:

- API_KEY
- API_KEY_SECRET
- ACCESS_TOKEN
- ACCESS_TOKEN_SECRET

Obteniendo estas credenciales puedes comenzar a obtener información de Twitter, autenticándote correctamente cada vez que hagas una solicitud a la API. Esto se puede hacer mediante el uso del método *OAuth1* de la librería *requests_oauthlib*, donde al entregarle como parámetro las cuatro contraseñas, podrás usarlo como parámetro en los *requests*.

6.3.1. Trending topics de una ubicación

Con las coordenadas obtenidas anteriormente, debes obtener los *trending topics* (o *#hashtags*, si prefieres) que se dieron en esa ubicación específica y debes obtener aquellos **cinco** que tuvieron mayor número de *tweets*.

6.3.2. Tweets en una ubicación

Dadas las coordenadas anteriores, tienes que obtener, usando la API, 20 *tweets* (en el caso de que hayan menos de veinte *tweets* en esa ubicación, debes obtenerlos todos) que se escribieron en esa ubicación en los últimos 2 días y obtener el número de seguidores (o *followers*) que tienen los usuarios que los escribieron. Una vez obtenida esta información, debes ordenar los *tweets* obtenidos según el número de seguidores de los usuarios.

6.4. Generar *log* de búsqueda (15 %)

Como programadores nos interesa mantener un seguimiento constante de que es lo que ocurre con nuestro programa, por lo que hay que mantener un sistema de notificaciones y registro de información actualizado. Para estructurar nuestros registros, habrá cuatro categorías de ocurrencias que debemos identificar en pantalla.

- Búsqueda exitosa: ocurre cuando todo opera de la forma que se ha indicado en las secciones anteriores del enunciado.

```
[2017-11-21 00:29:18,008] [Info] Búsqueda exitosa: La consulta funcionó correctamente.
```

- Alerta de *tweets*: ocurre cuando se han realizado menos de 20 *tweets* en una ubicación.

```
[2017-11-21 00:29:18,008] [Warning] Alerta de tweet: Hay menos de 20 tweets en la zona.
```

- Error de localización: ocurre cuando la API de Google Maps no es capaz de localizar un lugar.

```
[2017-11-21 00:29:18,008] [Error] Error de localización: La localización no existe.
```

- Error de tweets: ocurre cuando no se encuentra ningún *tweet*, por lo que no es posible llevar a cabo las funcionalidades.

```
[2017-11-21 00:29:18,008] [Error] Error de tweet: No se encontró ningún tweet.
```

Para realizar los registros, **deberás** usar el módulo `logging`, anotándolos con las herramientas que te entrega. **Para esto solo podrás usar una sola instancia de la clase `Logger`**. Con sus distintos métodos deberás notificar, a través de la interfaz, los sucesos anteriormente descritos, y adicionalmente deberás anotar los errores en un archivo de texto.

Todas las anotaciones deberán llevar la siguiente información.

- **Fecha y hora:** con el formato dado por el módulo `logging`.
- **Tipo de notificación:** indica si ese log es para informar, alertar o informar de un error ocurrido.
- **Información:** El detalle de porqué se generó esa notificación.

Finalmente se debe generar una hoja nueva dentro de la planilla de la sección 6.1, donde se guarden todas las acciones hechas. Cada entrada, representada por una fila, debe tener una columna con la fecha y hora, otra con el tipo de suceso, y otra con la información. Si el suceso es un error, toda la fila debe estar con letra negrita.

7. GUI (15 %)

Toda la interacción de tu programa con los usuarios, debe ser a través de una interfaz gráfica. Esta debe tener algunos requisitos mínimos; sin embargo, el orden o manera de cumplirlos quedará a tu criterio. En primer lugar, antes de iniciar la obtención de información, se debe dar a elegir la planilla ingresando el enlace de un Google Sheets que se usará para obtener los países y ciudades. Luego, se debe poder seleccionar un país y poder desplegar la lista de ciudades que se encuentran en él. Después de mostrar las ciudades de dicho país, se debe poder elegir alguna y mediante el uso de las APIs obtener las estadísticas de Twitter de la elección. Toda la información de las estadísticas debe estar ordenada y visualizada en pantalla. El usuario debe poder ver sólo las estadísticas que él desee; es decir, se debe poder seleccionar o filtrar que datos obtener.

Finalmente, también se debe poder acceder al enlace que contiene todos los *sheets* de ciudades y *log*.

8. Restricciones y alcances

- Tu programa debe ser desarrollado en Python v3.6.
- Una tarea que no funcione a través de una interfaz gráfica será evaluada con la nota mínima.
- Esta tarea es estrictamente individual, y está regida por el Código de Honor de la Escuela: Clickear para Leer.
- Tu código debe seguir la guía de estilos descrita en el **PEP8**.
- Si no se encuentra especificado en el enunciado, asume que el uso de cualquier librería Python está prohibida. Pregunta en el foro si es que es posible utilizar alguna librería en particular.
- El ayudante puede castigar el puntaje¹ de tu tarea, si le parece adecuado. Se recomienda ordenar el código y ser lo más claro y eficiente posible en la creación algoritmos.
- Debes adjuntar un archivo **README.md** donde comentes sus alcances y el funcionamiento del sistema (*i.e.* manual de usuario) de forma *concisa y clara*. **Tendrás hasta 24 horas después de la fecha de entrega** de la tarea para subir el **README.md** a tu repositorio.
- Crea un módulo para cada conjunto de clases. Divídelas por las relaciones y los tipos que poseen en común. **Se descontará hasta un punto si se entrega la tarea en un solo módulo**².
- Cualquier aspecto no especificado queda a tu criterio, siempre que no pase por sobre otro.

Tareas que no cumplan con las restricciones señaladas en este enunciado tendrán la calificación mínima (1.0).

¹ Hasta -5 décimas.

² No agarres tu código de un solo módulo para dividirlo en dos; separa su código de forma lógica