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

Actividad 04

1^o **semestre 2018**
12 de abril

Manejo de excepciones y *testing*

Introducción

Últimamente, la renombrada página de videos Tube-You ha tenido problemas con sus usuarios, por lo que su *staff* se ha mantenido muy ocupado tratando de resolverlos. En este esfuerzo, intentaron construir un algoritmo para generar estadísticas de sus videos *trending* sin éxito, debido a que sus programadores no sabían manejar excepciones. Es por esto que se contactaron contigo para que los ayudes a terminar el algoritmo.

Instrucciones

En esta actividad se medirá la capacidad de poder identificar errores de una base de datos que podrás ver en `data_errores.txt` que el equipo de Tube-You ya lee por ti. Deberás levantar una excepción correspondiente al error de formato en los datos. Luego, deberás manejar estas excepciones y, por último, crear *tests* unitarios.

Parte I. Identificación de errores

El equipo de Tube-You creó un módulo `library.py` para analizar los datos de sus videos. Sin embargo, dicho código no identifica adecuadamente los errores de formato (o de consistencia) que pueden aparecer. Tu deber será tomar la librería fallida, identificar los errores pertinentes, y levantar las excepciones según sea necesario.

Para esta parte trabajarás en el archivo `library.py`, donde podrás encontrar las funciones a revisar. **ESTÁ PROHIBIDO BORRAR CUALQUIER LÍNEA DE CÓDIGO DE ESTE ARCHIVO**; es decir, sólo podrás agregar las líneas necesarias. Además, tampoco podrás utilizar `try` o `except` en el archivo `library.py`, pues el manejo de excepciones lo haremos después en otro módulo.

Los errores a identificar por función son los siguientes:

- `tiempo_trending(publish_date: str, trending_date: str) -> int`:

  Esta función retorna un número entero que indica la cantidad de días desde que el video fue publicado, hasta que se volvió *trending*.

  Para esta función, debemos revisar el argumento `publish_date`. La fecha `publish_date` debe venir con el formato `%y. %d. %m`, en el que el año, día y mes aparecen con dos dígitos y separados con puntos. Por

ejemplo, la fecha de hoy escrita como "18.12.04" es válida, mientras que "18/12/04", "12/04/2018" o "12-abril" no respetan el formato.

Por lo tanto, se debe levantar una excepción personalizada **creada por ti** cuando la fecha no esté en el formato correcto.

- `like_dislike_ratio(likes: str, dislikes: str) -> float`:

  Esta función retorna el cociente entre los *likes* y *dislikes* del video.

  En esta función debemos revisar el formato de `likes`. Si el argumento *likes* no está compuesto por solo números, entonces se debe levantar una excepción del tipo adecuado. Por ejemplo, un *input* válido es "1546", mientras que "1$5-4r6" no lo es.

  Como se espera obtener el resultado de una división, además debes asegurarte de que una excepción de tipo `ZeroDivisionError` se levante al dividir por cero[1].

- `info_video(title: str, views: str, likes: str, dislikes: str, tags: str)`:

  Esta función imprime un resumen de las estadísticas del video.

  En este caso, **debes crear dos *custom exceptions***. La primera excepción se levantará, cuando algún video posea mas *likes* que *views*, es decir, cuando *likes* > *views*. La segunda, se levantará en caso de que *tags* sea un *string* de largo 0 o `None`.

Todas las *custom exceptions*, deberán tener un mensaje personalizado, que haga reconocible la excepción que se esta levantando. Estas deben guardarse en un modulo llamado `custom_exceptions.py`.

Parte II. Archivo de *logs* para excepciones

Una vez que hayas levantado las excepciones indicadas en la parte anterior, deberás encargarte de manejarlas de tal manera que el código en `main.py` no se detenga por ellos. Por lo tanto, **si un video genera una excepción se deja de procesar inmediatamente para seguir con el siguiente**.

Además, debes crear un archivo de *logs* llamado `excepciones.txt`, donde se registrará cada una de las excepciones que sean levantadas. Entonces, si algún video de la base de datos levanta una excepción, dicho video deja de ser procesado, y luego será añadido al *log* de la siguiente manera:

El video <title> levantó la siguiente excepción: <nombre_de_excepción>.

Parte III. *Testing*

Una vez que ya hayas solucionado los problemas, deberás demostrar que la librería funciona correctamente. Para esto deberás crear un archivo llamado `testing.py` donde exista una clase que testee las siguientes situaciones:

1. Test de funciones
   - Verificar que la función `tiempo_trending` retorna la cantidad de días correctos según las fechas entregadas.
   - Verificar que la función `like_dislike_ratio` retorna el valor esperado según los parámetros entregados.
2. Test de excepciones

[1] Recuerda que esta excepción se levanta por defecto al dividir por cero en Python.

- Verificar que la función `tiempo_trending` levanta una excepción al ingresar `publish_date` con un formato incorrecto.
- Verificar que la función `like_dislike_ratio` levante una excepción cuando los *likes* tienen caracteres no numéricos.

Notas

- No pueden modificar la clase video ni ningún funcionamiento de las funciones. Solo deben levantar excepciones y manejarlas. Pueden crear funciones auxiliares.
- Deben utilizar correctamente cada tipo de excepción. **El uso de `except Exception` será penalizado.**

Requerimientos

- (2.0 pts) Levantamiento de excepciones:
  - (0.4 pts) Se levanta una excepción correspondiente al distinto formato de fechas.
  - (0.4 pts) Se levanta una excepción cuando los *likes* tienen caracteres no numéricos.
  - (0.6 pts) Se levanta una excepción cuando los *likes* son mayores que los *views*.
  - (0.6 pts) Se levanta una excepción cuando no existen *tags* o tienen largo 0.
- (2.40 pts) Archivo de *logs*:
  - (0.4 pts) Se maneja correctamente la excepción correspondiente al distinto formato de fechas.
  - (0.4 pts) Se maneja correctamente la excepción cuando los *likes* tienen caracteres no numéricos.
  - (0.4 pts) Se maneja correctamente la excepción cuando los *dislikes* son 0.
  - (0.4 pts) Se maneja correctamente la excepción cuando los *likes* son mayores que los *views*.
  - (0.4 pts) Se maneja correctamente la excepción cuando no existen *tags* o tienen largo 0.
  - (0.4 pts) Se registra cada excepción levantada a un archivo de *logs*.
- (1.60 pts) *Testing*:
  - (0.4 pts) Uso de `setUp`.
  - (0.4 pts) Verificar funcionalidad de `tiempo_trending`.
  - (0.4 pts) Verificar funcionalidad de `like_dislike_ratio`.
  - (0.2 pts) Verificar que ocurra la excepción con formato de fecha erróneo.
  - (0.2 pts) Verificar que ocurra la excepción cuando los *likes* no se componen de sólo dígitos.

Entrega

- **Lugar:** En su repositorio de GitHub en la **carpeta** `Actividades/AC04/`
- **Hora:** 16:30

Créditos

La DB fue extraída desde `https://www.kaggle.com/datasnaek/youtube-new`