

Examen final 2020-09-07

95.14/75.40 - Algoritmos y Programación I - Curso Essaya

Objetivo

Implementar la clase `Tuiter`, que permite modelar el funcionamiento de una plataforma sospechosamente similar a Twitter.

Se dispone de los siguientes archivos:

- `tuiter.py`: implementación (inicialmente vacía) de la clase `Tuiter`
- `pruebas.py`: Pruebas automáticas

El archivo `pruebas.py` efectúa una serie de pruebas para verificar el correcto funcionamiento de la clase `Tuiter`.

La idea es agregar en `tuiter.py` todo el código necesario para que las pruebas automáticas pasen.

Hay 5 ejercicios. Es condición necesaria (pero no suficiente) para aprobar el examen que haya 3 ejercicios OK.

Salida del programa

Al ejecutar el programa (`python3 pruebas.py`), se ejecutan todas las pruebas, y se imprime el resultado de cada ejercicio (OK o FAIL), junto con la cantidad de ejercicios OK. Ejemplo:

```
$ python3 pruebas.py
ejercicio_1: OK
ejercicio_2: OK
ejercicio_3: OK
```

```
Traceback (most recent call last):
  File "pruebas.py", line 237, in main
    ejercicio()
  File "pruebas.py", line 148, in ejercicio_4
    assert ok
AssertionError
```

```
ejercicio_4: FAIL
ejercicio_5: OK
Cantidad de ejercicios OK: 4
```

Recordar: es condición necesaria (pero no suficiente) para aprobar el examen que haya al menos 3 ejercicios OK.

La clase `Tuiter`

La clase `Tuiter` modela una plataforma de publicación de contenido (**tuits**) por parte de usuarios (**autores**).

Un **autor** tiene, como mínimo:

- Un número identificador único (el ID del autor)
- Un nombre
- Un **muro** (explicado abajo)

Un **tuit** tiene, como mínimo:

- Un número identificador único (el ID del tuit)
- El ID del autor del tuit
- Un mensaje

IDs

Desde "afuera" de la implementación, tanto los tuits como los usuarios se identifican mediante sus IDs. Es decir, todas las operaciones reciben y devuelven IDs. La elección de estos IDs al crear usuarios/tuits queda a cargo de la clase Twitter.

Los IDs de los usuarios deben ser únicos, y lo mismo para los tuits, pero no hay problema en que un tuit y un autor tengan el mismo ID.

Recomendación: usar números crecientes:

- Creamos el primer autor -> ID = 0
- Creamos el primer tuit -> ID = 0
- Creamos el segundo autor -> ID = 1
- Creamos el segundo tuit -> ID = 1
- Creamos el tercer tuit -> ID = 2

El muro

Cada autor tiene un **muro** que es una secuencia de tuits (que pueden haber sido publicados por el mismo autor o no):

- Al **publicar** un tuit, el tuit se agrega automáticamente al muro del autor.
- Un autor puede **compartir** un tuit previamente publicado por otro autor. El tuit se agrega al muro del autor que lo comparte.
 - Un autor no puede compartir un tuit creado por sí mismo.
 - El tuit puede ser compartido muchas veces por el mismo autor (siempre y cuando no sea el autor del tuit).

Descripción de las pruebas

ejercicio_1: Publicar y compartir Prueba el funcionamiento básico de Twitter. Sin esta prueba funcionando probablemente no se pueda pasar ninguna de las otras pruebas.

Métodos a implementar:

- `__init__`: Crea una instancia de Twitter con 0 autores y 0 tuits.
- `crear_autor`: Recibe un nombre, y crea un nuevo autor con un ID único (es decir, no puede ser igual al ID de otro autor ya existente) y el nombre especificado. Devuelve el ID del autor.
- `publicar`: Recibe el ID de un autor y un mensaje, y crea un nuevo tuit con un ID único (es decir, no puede ser igual al ID de otro tuit ya existente), el autor y el mensaje especificados. Agrega el tuit al muro del autor. Devuelve el ID del tuit.
- `compartir`: Recibe el ID de un tuit y el ID de un autor. Si el tuit puede ser compartido por el autor lo agrega al muro de ese autor y devuelve True. Devuelve False en caso contrario.
- `tuit_id_autor`: Recibe el ID de un tuit, y devuelve el ID de su autor.
- `tuit_mensaje`: Recibe el ID de un tuit, y devuelve el mensaje del tuit.

- `muro_cantidad`: Recibe el ID de un autor, y devuelve la cantidad de tuits que contiene el muro de ese autor.
- `muro_id_tuit`: Recibe el ID de un autor y una posición `p` (entero no negativo). Devuelve el ID del tuit que está en la posición `p` en el muro del autor (según el orden en que fueron agregados al muro). Es decir, si `p = 0` devuelve el ID del primer tuit agregado al muro, si `p = 1` el segundo tuit, etc.

ejercicio_2: CSV Prueba que los tuits de un muro puedan ser guardados en un archivo CSV.

Métodos a implementar:

- `muro_escribir_csv`: Recibe el ID de un autor y el nombre de un archivo. Escribe en el archivo el contenido de todos los tuits del muro del autor, con formato `autor|mensaje` (incluyendo la cabecera).

ejercicio_3: Tuits más compartidos Prueba que podamos obtener los IDs de los tuits ordenados por cantidad de veces que fueron compartidos.

Métodos a implementar:

- `tuits_mas_compartidos`: Devuelve una lista de tuplas, en la que cada tupla tiene la forma `(id_tuit, cantidad)`, donde `cantidad` es la cantidad de veces que el tuit fue compartido. La lista está ordenada por cantidad en forma descendente, es decir el tuit más compartido primero.

ejercicio_4: Likes Prueba que los tuits puedan ser "likeados".

- Un tuit puede ser likeado por cualquier autor menos el autor del tuit.
- Un tuit no puede ser likeado más de una vez por el mismo autor.

Métodos a implementar:

- `tuit_dar_like`: Recibe el ID de un tuit y el ID de un autor. Si el tuit puede ser likeado por el autor, le da like y devuelve `True`. En caso contrario devuelve `False`.
- `tuit_fue_likeado_por`: Recibe el ID de un tuit y el ID de un autor. Devuelve `True` o `False` según si el tuit ya fue likeado por el autor o no.

ejercicio_5: Hilos Prueba el mecanismo de respuestas de tuits.

- Un autor puede publicar un tuit **en respuesta de** otro tuit ya existente, generando así un **hilo**.
- Los hilos no son lineales sino que pueden tener forma de árbol: hay un tuit que es la raíz o el tronco del árbol y luego hay respuestas que son las ramificaciones; y cada respuesta a su vez puede tener más respuestas.

Métodos a implementar:

- `responder`: Recibe el ID de un tuit, el ID de un autor y un mensaje, y publica un nuevo tuit en respuesta del tuit indicado. El nuevo tuit tendrá el autor y mensaje indicados y se agregará al muro del autor, igual que publicar.
- `tuit_respuestas`: Recibe el ID de un tuit y devuelve la lista de IDs de los tuits que son respuestas inmediatas (es decir, no incluye los tuits que son respuestas de respuestas).
- `tuit_en_respuesta_de`: Recibe el ID de un tuit. Si el tuit es la raíz de un hilo, devuelve `None` ya que no es en respuesta de ningún otro tuit. En caso contrario, devuelve el ID del tuit que responde.
- `tuit_cantidad_hilo`: Recibe el ID de un tuit, y devuelve la **cantidad** del hilo que tiene como raíz el tuit indicado. Es decir, la cantidad de tuits en total contando el tuit y todas sus respuestas, en forma recursiva.