

Notificador de polen

Jesús Rodríguez Heras
Juan Pedro Rodríguez Gracia

24 de julio de 2018

Resumen

Notificador de polen para las personas alérgicas.

Índice

1. Introducción	4
2. Almacenamiento y salida de datos	4
3. Tecnologías utilizadas	4
4. Problemas encontrados	4
5. Demostración	5
6. Referencias	8

1. Introducción

Nuestra aplicación consta de dos programas: `tareas.py` y `PolenCiudad.py`.

- **tareas.py**: El comportamiento de este programa se basa en recolectar la información acerca del polen que hay en cada ciudad de `https://www.eltiempo.es/polen`. La obtención de datos se realiza periódicamente cada 2 horas y se insertan en un archivo “.csv” que se guarda en dropbox. También genera una gráfica que se mostrará por pantalla para que se pueda ver, de modo general el nivel de polen en el resto de todas las ciudades.
- **PolenCiudad.py**: Este programa se encarga de interactuar directamente con el usuario, el cual introducirá la ciudad por la que está interesado en conocer el nivel de polen. El programa descarga el archivo más reciente de los existentes en dropbox (almacenados por el programa anterior) y le devolverá el nivel de polen (alto, medio o bajo) por consola y, además, lo publicará en twitter (`https://twitter.com/MuleSteam`)¹ para que las personas que sigan la cuenta sepan el nivel de polen que hay en las ciudades que han buscado otros usuarios.

2. Almacenamiento y salida de datos

A la hora de almacenar y leer los datos obtenidos de `https://www.eltiempo.es/polen` en dropbox, lo hacemos mediante la librería oficial de python para dropbox.

También se creará una gráfica con el nivel de polen existente por cada provincia como información adicional para el usuario.

3. Tecnologías utilizadas

Para esta aplicación hemos utilizado python como lenguaje de programación base. Aparte de ello, hemos usado una cola de mensajes como es RabbitMQ debido a la simple interfaz que aporta a la hora de trabajar con ella.

También hemos usado la librería oficial de dropbox para python que permitir la comunicación entre la aplicación y la nube para así poder almacenar los datos que se irán recogiendo cada dos horas (cada minuto en las pruebas) y las lecturas a dichos datos que se harán cuando un usuario de la aplicación consulte acerca del nivel de polen existente en su provincia.

4. Problemas encontrados

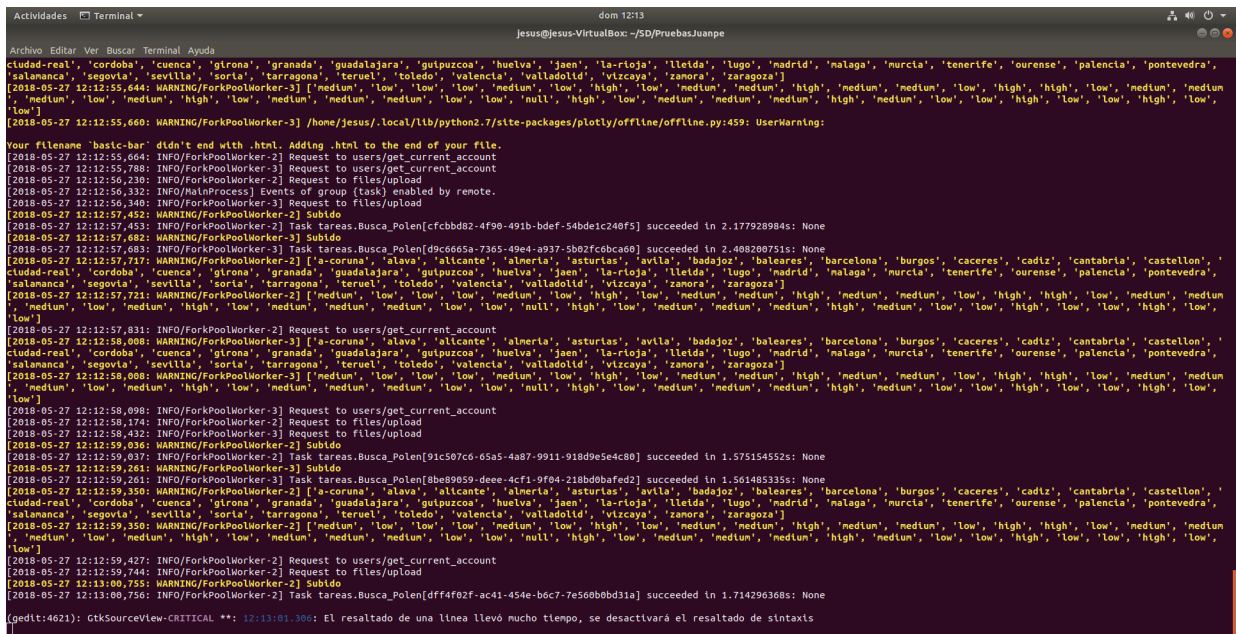
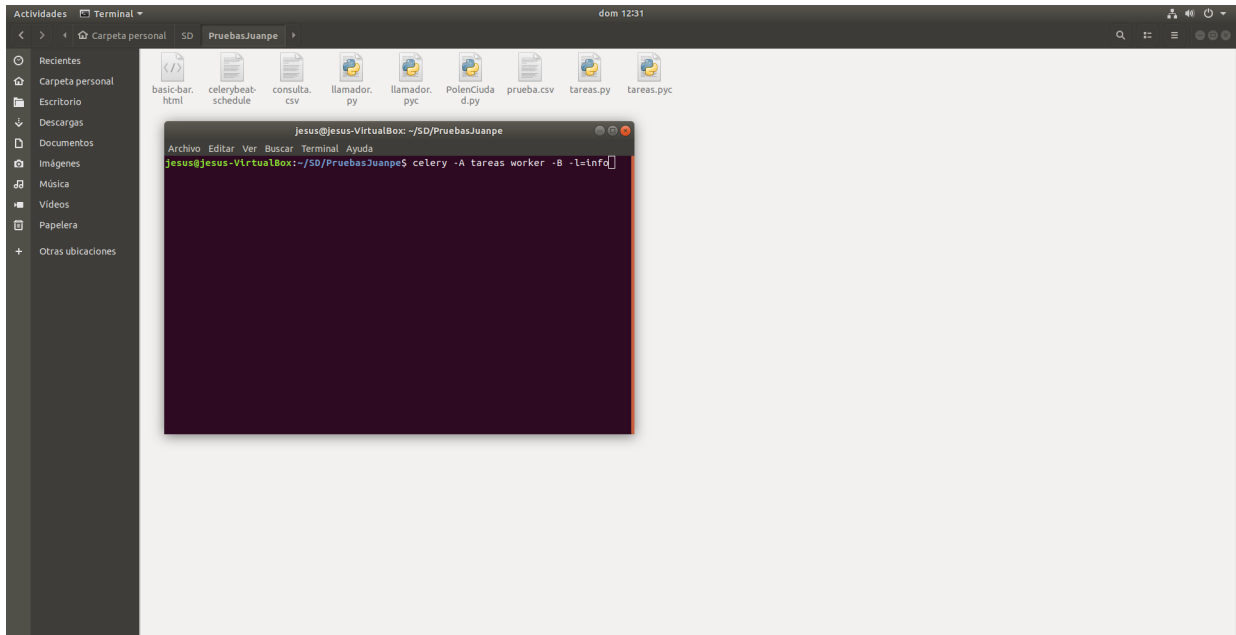
El problema principal que nos encontramos fue que no podíamos instalar RabbitMQ en nuestros portátiles, los cuales cuentan con el sistema operativo Kali Linux (basado en Debian), por lo que tuvimos que hacer el proyecto en una máquina virtual con Ubuntu.

¹Hemos utilizado la misma cuenta de twitter que para la práctica de Mule debido a que ya teníamos las claves y no hemos visto necesario el hecho de cambiar de cuenta, ya que es, solamente, para fines académicos.

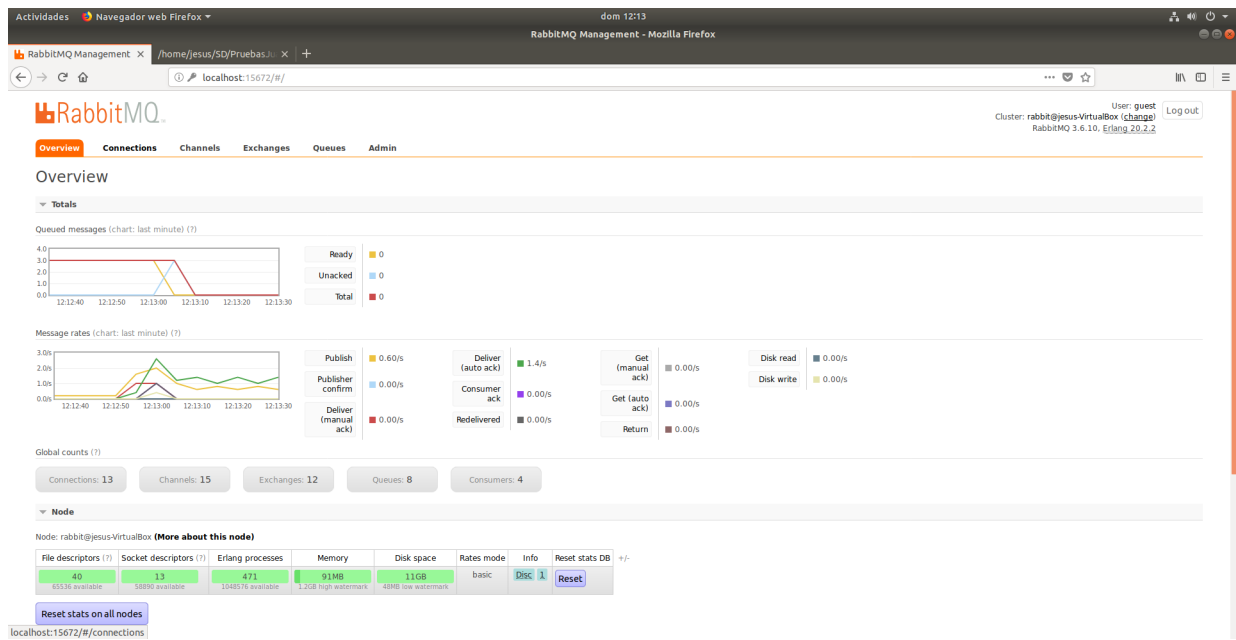
También nos hemos encontrado otros problemas menores a la hora de trabajar con python y con las librerías de dropbox y twitter.

5. Demostración

- Lanzamos el worker `tareas.py` con Celery:



- Se lanzan peticiones a RabbitMQ:



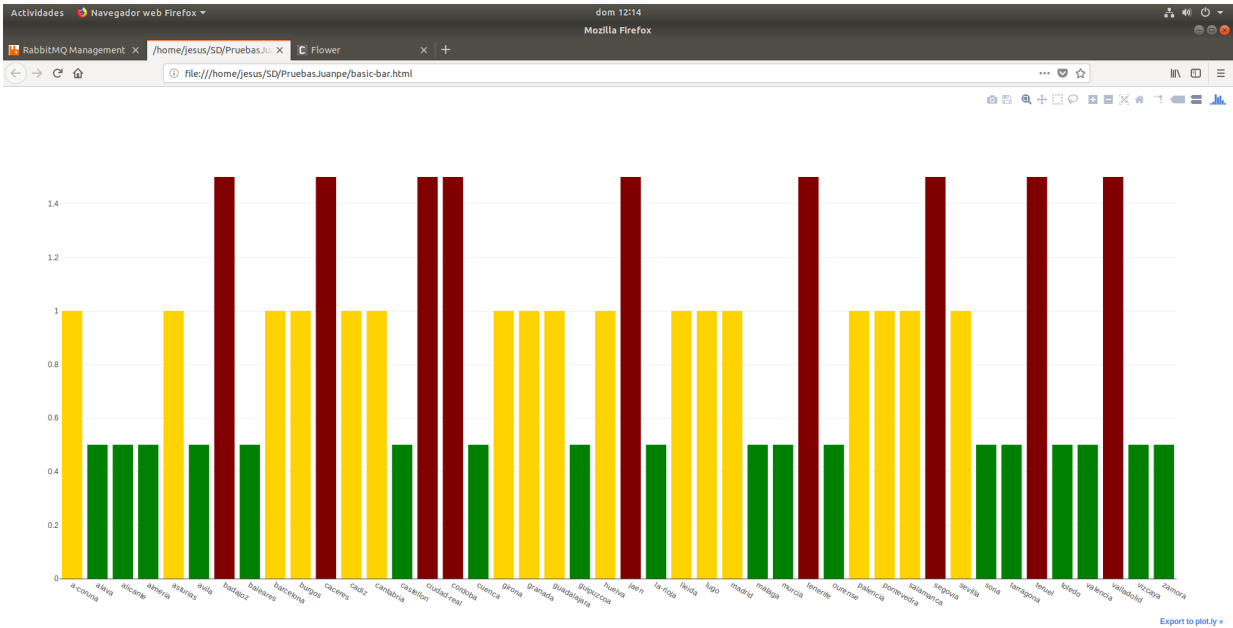
- También las podemos ver en la interfaz de flower:

The screenshot shows the Flower web interface in a Mozilla Firefox browser. The address bar indicates the URL is localhost:5555. The interface includes a top navigation bar with tabs: Dashboard, Tasks, Broker, and Monitor. The Dashboard tab is active, showing a summary of worker statistics: Active: 0, Processed: 1, Failed: 0, Succeeded: 6, Retried: 0. Below this, there is a table with columns: Worker Name, Status, Active, Processed, Failed, Succeeded, Retried, and Load Average.

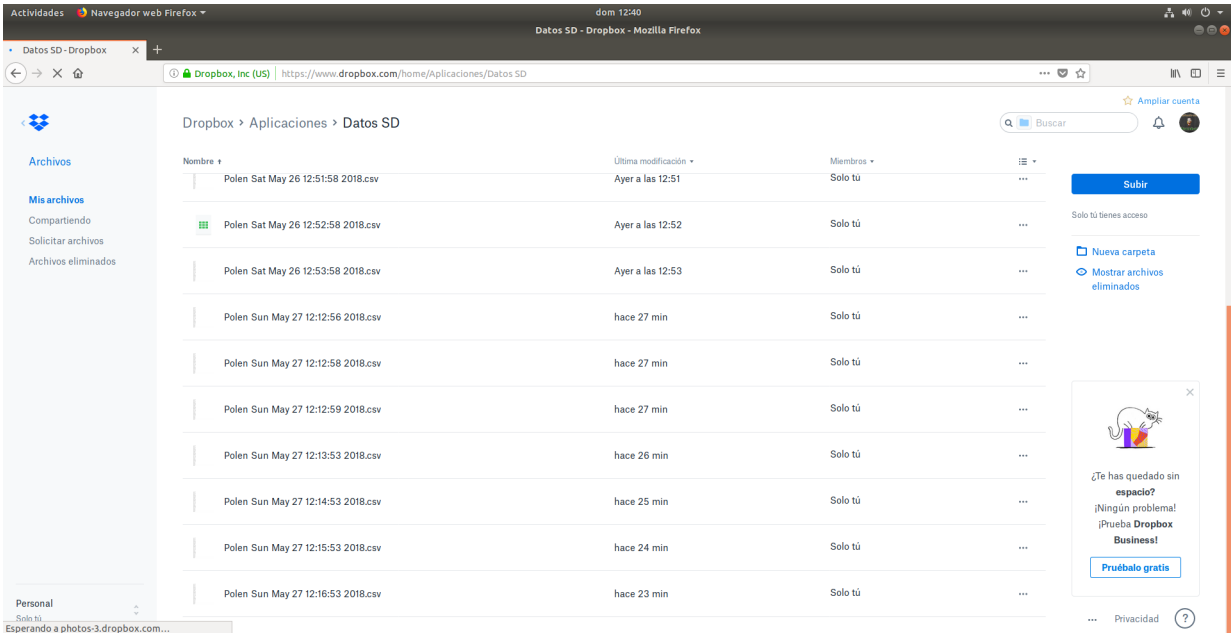
Worker Name	Status	Active	Processed	Failed	Succeeded	Retried	Load Average
celery@jesus-VirtualBox	Online	0	1	0	6	0	1.46, 0.7, 0.47

Below the table, it says 'Showing 1 to 1 of 1 entries'.

- Gráfica generada por `tasks.py`:



- Resultado en Dropbox de la ejecución de `tareas.py` (cada minuto en las pruebas, pero la finalidad es cada una o dos horas):



- Uso de PolenCiudad.py por parte de un usuario:

```

Actividades Terminal
dom 12:14
Jesus@Jesus-VirtualBox: ~/SD/PruebasJuanpe$ python PolenCiudad.py
Jesus@Jesus-VirtualBox:~/SD/PruebasJuanpe$ python PolenCiudad.py
Sat May 26 12:28:22 2018
Sat May 26 12:35:20 2018
Sat May 26 12:38:27 2018
Sat May 26 12:38:53 2018
Sat May 26 12:47:49 2018
Sat May 26 12:50:02 2018
Sat May 26 12:51:00 2018
Sat May 26 12:51:58 2018
Sat May 26 12:52:58 2018
Sat May 26 12:53:58 2018
Polen Sat May 26 12:53:58 2018.csv
Introduzca la ciudad donde quiere ver el nivel de polen: sevilla
El nivel de polen en sevilla es alto

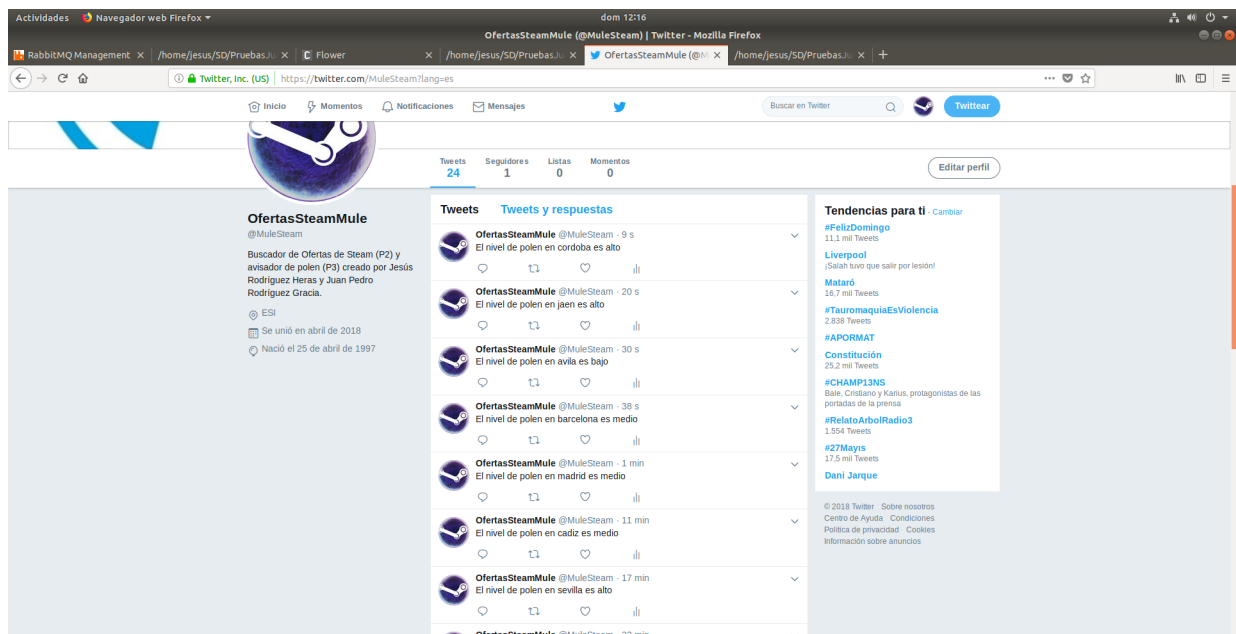
Jesus@Jesus-VirtualBox:~/SD/PruebasJuanpe$ python PolenCiudad.py
Sat May 26 12:28:22 2018
Sat May 26 12:35:20 2018
Sat May 26 12:38:27 2018
Sat May 26 12:38:53 2018
Sat May 26 12:47:49 2018
Sat May 26 12:50:02 2018
Sat May 26 12:51:00 2018
Sat May 26 12:51:58 2018
Sat May 26 12:52:58 2018
Sat May 26 12:53:58 2018
Polen Sat May 26 12:53:58 2018.csv
Introduzca la ciudad donde quiere ver el nivel de polen: dz
Introduzca la ciudad donde quiere ver el nivel de polen: dz
El nivel de polen en cadiz es medio

Jesus@Jesus-VirtualBox:~/SD/PruebasJuanpe$ python PolenCiudad.py
Sat May 26 12:28:22 2018
Sat May 26 12:35:20 2018
Sat May 26 12:38:27 2018
Sat May 26 12:38:53 2018
Sat May 26 12:47:49 2018
Sat May 26 12:50:02 2018
Sat May 26 12:51:00 2018
Sat May 26 12:51:58 2018
Sat May 26 12:52:58 2018
Sat May 26 12:53:58 2018
Sun May 27 12:12:56 2018
Sun May 27 12:12:58 2018
Sun May 27 12:13:53 2018
Polen Sun May 27 12:13:53 2018.csv
Introduzca la ciudad donde quiere ver el nivel de polen: madrid
El nivel de polen en madrid es medio

Jesus@Jesus-VirtualBox:~/SD/PruebasJuanpe$

```

- Resultados en twitter generados por la búsqueda del usuario:



6. Referencias

- <https://www.eltiempo.es/polen>
- <https://www.rabbitmq.com/>
- <http://www.celeryproject.org/>
- <http://flower.readthedocs.io/en/latest/>

- <http://dropbox-sdk-python.readthedocs.io/en/latest/>