

DOCUMENTACIÓN CHALLENGE

CASO - KEYWORD EN Gmail

Autor: Erika Ganchozo



1. Índice

1.	Índice.....	1
2.	Contexto.....	2
3.	Objetivo.....	2
4.	Información necesaria para la ejecución.....	2
4.1.	Librerías.....	2
4.2.	Módulos y funciones:	2
4.3.	Acceso a Gmail	3
5.	Descripción de la aplicación, problemas y soluciones.....	3
5.1.	Descripción de la aplicación.....	3
5.2.	Problemas y soluciones	3
6.	Ejecución de la Aplicación.....	4

2. Contexto

Como parte del proceso de selección de la compañía Mercado Libre, se encuentra la actividad denominada “Challenge”, a través de la cual se evalúan los conocimientos en programación de aquellos aspirantes a formar parte del equipo de Seguridad de la Información.

3. Objetivo

Armar un programa en Python, Go, Groovy o Java que pueda acceder, de manera automática, a una cuenta de Gmail, leer los correos e identificar aquellos que tengan la palabra "DevOps" en el Body. De estos correos identificados, se deberá guardar en una base de datos MySQL los siguientes campos:

- fecha de recepción del correo
- to (remitente que ha enviado el correo)
- subject

Tener en cuenta que solo deberá guardar aquellos correos que ya no haya guardado en alguna corrida anterior.

4. Información necesaria para la ejecución

4.1. Librerías

Todas las librerías utilizadas se encuentran integradas en la biblioteca estándar de Python, por lo que no es necesario realizar la instalación por separado. A continuación, se agrega un breve detalle de cada una:

- **Imaplib**: Protocolo del cliente IMAP4
- **Email**: Paquete de manejo de correo electrónico y MIME
- **Sqlite3**: DB-API 2.0 interfaz para bases de datos SQLite

4.2. Módulos y funciones:

Se utilizó la función `decode_header` que es proporcionada por el módulo `email.header`, la cual es una API heredada de la librería `email` (mencionada en el apartado 4.1). A continuación, se agrega un breve detalle de la función utilizada:

- **`decode_header`**: Decodifica un valor de encabezado de mensaje sin convertir el juego de caracteres.

4.3. Acceso a Gmail

El acceso a la cuenta de Gmail creada para el presente desarrollo, se realiza a través de los siguientes datos:

- Email: challengekeyword@gmail.com
- Contraseña: Ch@llenge2021

5. Descripción de la aplicación, problemas y soluciones

5.1. Descripción de la aplicación

Al iniciar la ejecución de la aplicación se solicita al usuario la cuenta de Gmail y contraseña para iniciar la sesión, validando si éstas son correctas para continuar la ejecución. Una vez iniciada la sesión, realiza la conexión al Servidor de correo entrante (IMAP) de Gmail y busca si existen mails nuevos en la bandeja de entrada.

En caso de identificar nuevos mails, se realiza la búsqueda de la palabra “DevOps” en cada uno de ellos y solo si la encuentra, agrega el registro (campos mencionados en el apartado 3 del presente documento) a la tabla “mails” de la base de datos “KeywordBD”.

En caso de no identificar nuevos mails, la aplicación finaliza su ejecución.

5.2. Problemas y soluciones

Durante el desarrollo de la aplicación surgieron varios errores los cuales fueron solucionados a través del análisis del código e investigación en varias fuentes de casos similares. Los principales fueron los siguientes:

- Contar los mails nuevos: Se definió un contador para recuperar el número de mails que se encontraban como no leídos en la bandeja de entrada, pero en principio realizaba un conteo de todos los mails de la bandeja (leídos y no leídos).

Luego se creó una lista con todos los datos recuperados mediante el método `search()` de `imaplib` (mails marcados como “unseen”), se usó el método `split()` en dicha lista para separar cada elemento dentro de ella y luego mediante `len()` se contaron estos elementos.

- Encontrar palabra “DevOps”: Para lograr identificar la palabra en el cuerpo del mail se realizaba la búsqueda de coincidencia directamente y no lograba identificarlo correctamente.

Para solucionarlo se usó el método `split()` para separar por líneas, luego se agregó un `for` para ir recorriendo cada palabra de cada línea y buscar la coincidencia de “DevOps”.

- Marcar mails como “leídos”: Cada que se ejecutaba la aplicación leía los mails, pero no los marcaba como leídos en el buzón, por lo cual los seguía leyendo en cada ejecución. La solución a esto fue muy sencilla, ya que solo se agregó la instrucción `readonly = False` cuando se selecciona el buzón (Inbox) dentro de la función `select()` de `imaplib`.

6. Ejecución de la Aplicación

A continuación, se detallan los pasos para iniciar la aplicación “Keyword” y su ejecución:

1. Abrir consola de comandos
2. Cambiar al directorio donde se encuentra el archivo Keyword.py (En este caso se realizó en el directorio “C:\Users\ErikaG\Desktop”)
3. Ingresar “python Keyword.py” y luego presionar “Enter”.

```
C:\Users\ErikaG\Desktop>python Keyword.py
```

4. Ingresar cuenta de Gmail y contraseña.

```
C:\Users\ErikaG\Desktop>python Keyword.py
Introducir Email: challengekeyword@gmail.com
Introducir Contraseña:
```

En caso de ingresar las credenciales incorrectas se muestra un mensaje de error y finaliza la ejecución.

```
C:\Users\ErikaG\Desktop>python Keyword.py
Introducir Email: challengekeyword2021@gmail.com
Introducir Contraseña: Ch@llenge000
No fue posible iniciar sesión. Ingrese credenciales correctas
C:\Users\ErikaG\Desktop>
```

5. Realizar la búsqueda de nuevos mails.

Si no encuentra nuevos mails, muestra el mensaje “No se encontraron emails nuevos”.

```
C:\Users\ErikaG\Desktop>python Keyword.py
Introducir Email: challengekeyword@gmail.com
Introducir Contraseña: Ch@llenge2021
Iniciando aplicación....
No se encontraron emails nuevos
```

6. Guardar nuevos mails en la base de datos

Si encuentra nuevos mails, la aplicación muestra la cantidad de mails identificados, y solo agrega a la base de datos aquellos que contienen en el body la palabra “DevOps”. Crea la tabla “mails” (en caso de no existir) e imprime el registro de dicha tabla.

```
C:\Users\ErikaG\Desktop>python Keyword.py
Introducir Email: challengekeyword@gmail.com
Introducir Contraseña: Ch@llenge2021
Iniciando aplicación....
Se encontraron 2 emails nuevos
¡Se creó la tabla mails!

-----Contenido de Tabla mails-----
(1, 'Sun, 28 Feb 2021 19:40:15 +0000', 'Erika Ganchozo <sorygan_1@hotmail.com>', 'Otra prueba 4')
(2, 'Sun, 28 Feb 2021 19:40:07 +0000', 'Erika Ganchozo <sorygan_1@hotmail.com>', 'PRUEBA ')
C:\Users\ErikaG\Desktop>
```

Asimismo, crea el archivo “KeywordBD.db” en el mismo directorio donde se encuentra el script “Keyword.py”.

Nombre	Fecha de modificación	Tipo
 KeywordBD	28/02/2021 16:43	Data Base File
 Keyword	28/02/2021 16:38	Python File

Si la tabla “mails” ya existe, se muestra un mensaje indicándolo, agrega el nuevo mail identificado y muestra el registro de la tabla “mails”.

```
C:\Users\ErikaG\Desktop>python Keyword.py
Introducir Email: challengekeyword@gmail.com
Introducir Contraseña: Ch@llenge2021
Iniciando aplicación....
Se encontraron 1 emails nuevos
¡La tabla mails ya existe!

-----Contenido de Tabla mails-----
(1, 'Sun, 28 Feb 2021 19:40:15 +0000', 'Erika Ganchozo <sorygan_1@hotmail.com>', 'Otra prueba 4')
(2, 'Sun, 28 Feb 2021 19:40:07 +0000', 'Erika Ganchozo <sorygan_1@hotmail.com>', 'PRUEBA ')
(3, 'Sun, 28 Feb 2021 19:42:35 +0000', 'Erika Ganchozo <sorygan_1@hotmail.com>', 'PRUEBA 200')
C:\Users\ErikaG\Desktop>
```