Contenido

[Instrucciones para la ejecución de la aplicación 2](#_Toc62572633)

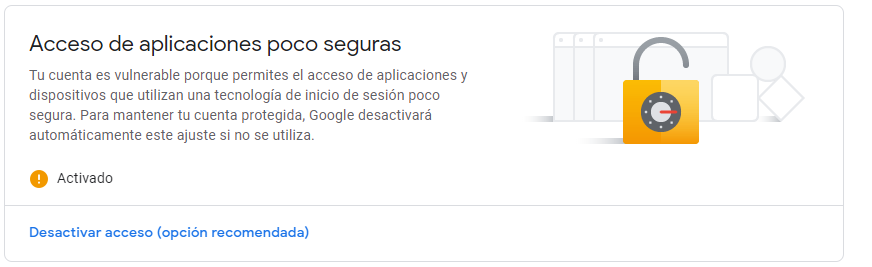
[Descripción de la aplicación 5](#_Toc62572634)

## Instrucciones para la ejecución de la aplicación

1. El primer paso es tener la casilla de mail a validar, la utilizada para el caso es: [*casomlac@gmail.com*](mailto:casomlac@gmail.com)

La contraseña es: Prueba321\*

Y a su vez, verificar que este activada la opción de ingreso mediante aplicaciones poco seguras, esto se puede validar desde Seguridad.



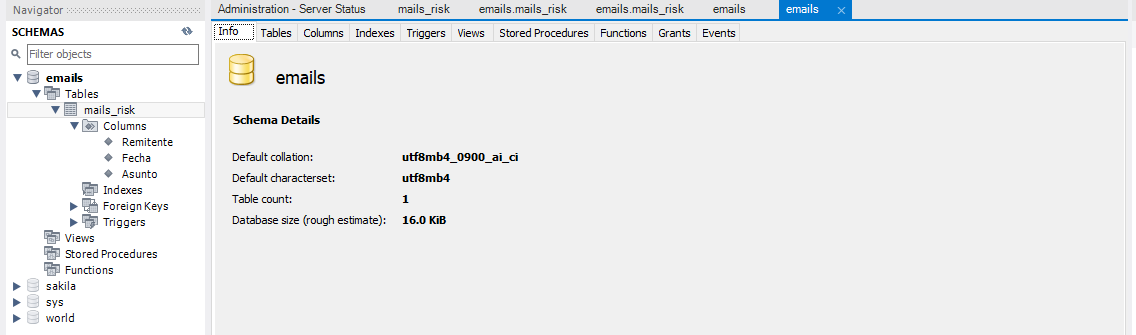
1. Para ejecutar el programa debe tener instalada la aplicación de Visual Studio Code. Una vez instalada, se deben agregar tres extensiones 🡪 

* Jupyter
* Pip-packages
* Python (python-3.8.1)

1. Y a su vez, se debe instalar Mysql (mysql-installer-community-8.0.23.0). La instalación debe hacerse con esa versión, una vez instalada se debe crear un servidor, y luego la Base de Datos (ver paso 5).
2. Luego de instalar todo lo mencionado anteriormente, desde Visual Studio Code procedemos a la sección “Explorer”🡪 “New file” 🡪 pegar el código del archivo “Email-listener2.txt” o agregar el archivo “Email-listener2.py”.
3. Una vez que el código esta listo en Visual Studio Code, volvemos a Mysql, y procedemos a la creación de la Base de Datos.

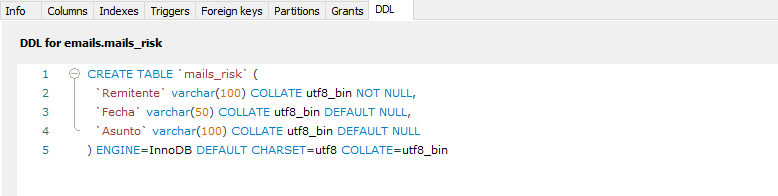
Para crear la base de datos debemos:

* Crear un servidor, con un usuario y contraseña.
* Crear la base

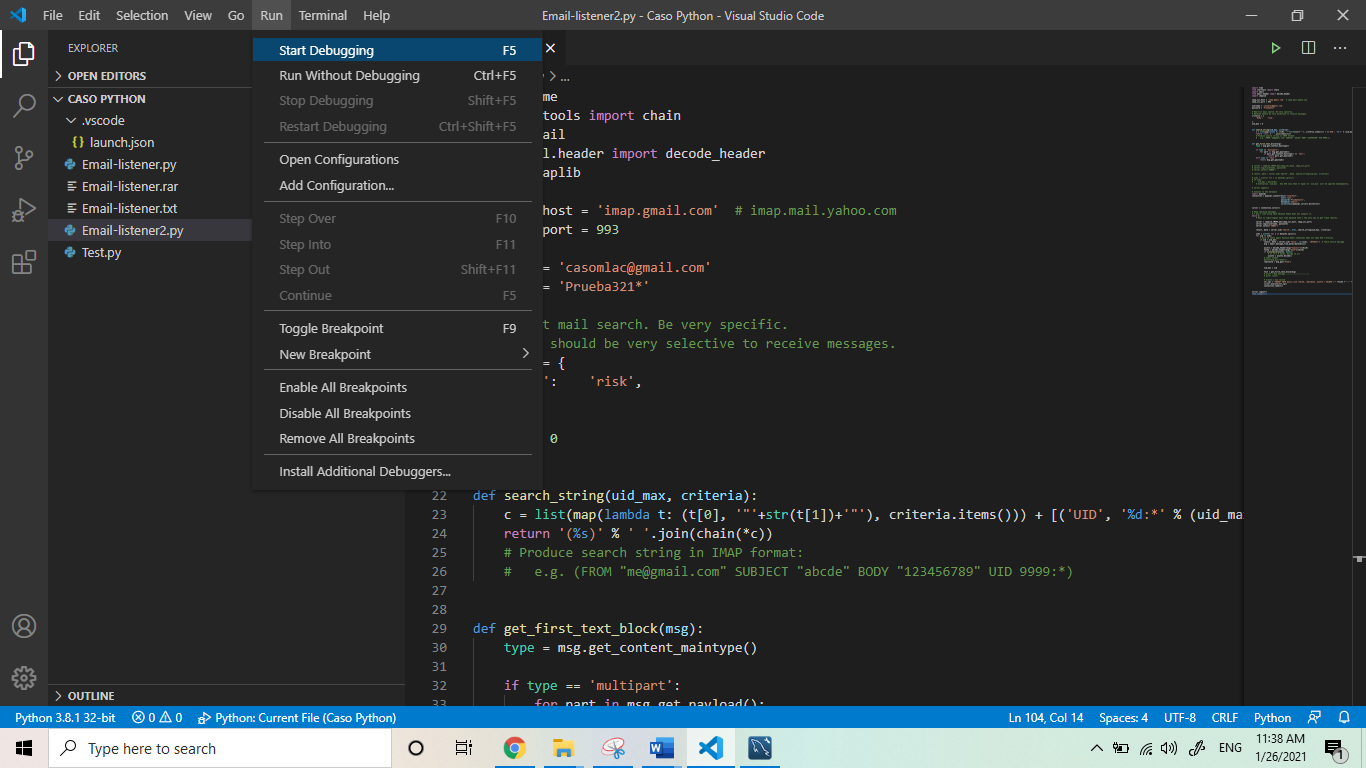
Detalle para la creación de la Base de Datos:

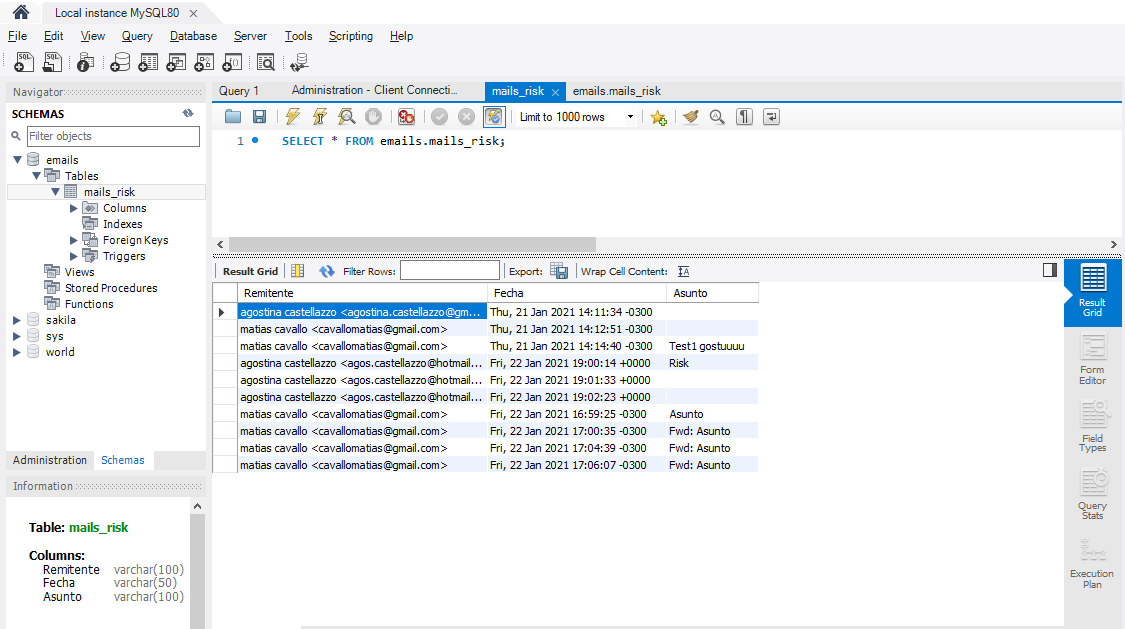
* Crear la tabla donde se almacenan los datos fecha, remitente del mail y asunto del mail.

Código para la creación de la tabla “mail\_risk”:



1. Luego de validar que el código funcione, que no haya errores de sintaxis, se debe ejecutar el programa (f5) desde Visual Studio Code.



1. Luego de que ejecute, desde MySQL, podemos visualizar el resultado de los mail que contengan la palabra “risk” en el cuerpo del mail.

## Descripción de la aplicación

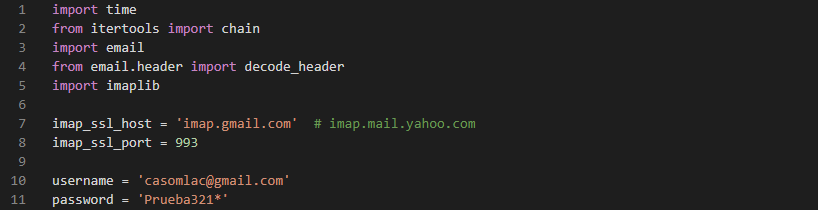
El primer problema con el me encontré fue sobre las configuraciones del mail, para que el programa funcione, se debe activar el ingreso por medio de aplicaciones poco seguras. Este dato no lo tenía en mente, y lo descubrí por leer foros sobre cómo lograr que una aplicación pueda leer mails. Entiendo que esta opción es poco segura, y dentro de una compañía se debería tener desactivada la misma.

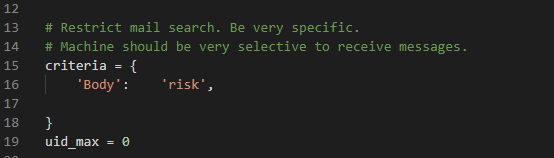
Una vez superada las instalaciones de las distintas aplicaciones y sus extensiones, más la creación del mail y su buen funcionamiento, llego el momento de empezar a desarrollar el código en Visual Studio Code.

Entre las líneas 1 y 11, se describen los datos a utilizar, y se configura el mail que va a ser revisado por el programa.

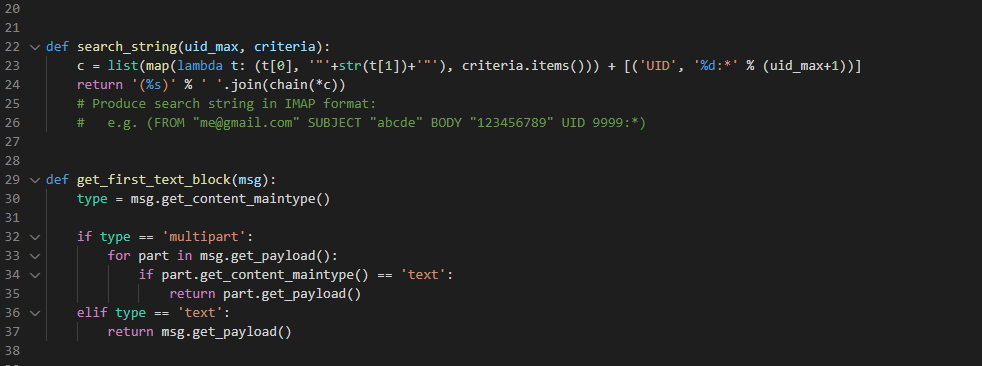
Parámetros: El puerto por defecto para Gmail es IMAP4\_SSL\_PORT es el 993.

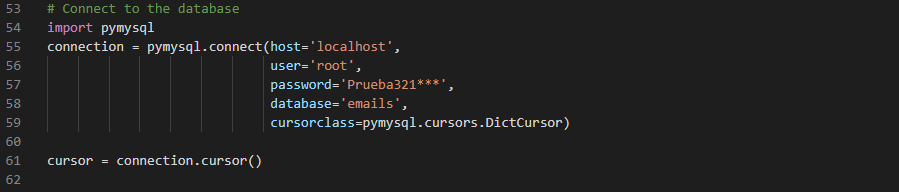
El servidor stmp de gmail mail = imap.gmail.com



Entre las líneas 12 y 19, se define el criterio de búsqueda, en este caso que se ecuentre la palabra “risk” en el cuerpo de los mails.

Entre las líneas 20 y 38, la sentencia DEF la use para crear objeto funcional, en este caso el buscar/search string, en base a *uid\_max* y el *criterio* definido anteriormente, tambien se define como es el formato de la busqueda en los mails, lo cual se aclara en comentarios (en verde).

A su vez, se define que extraiga el cuerpo del mensaje en formato text.

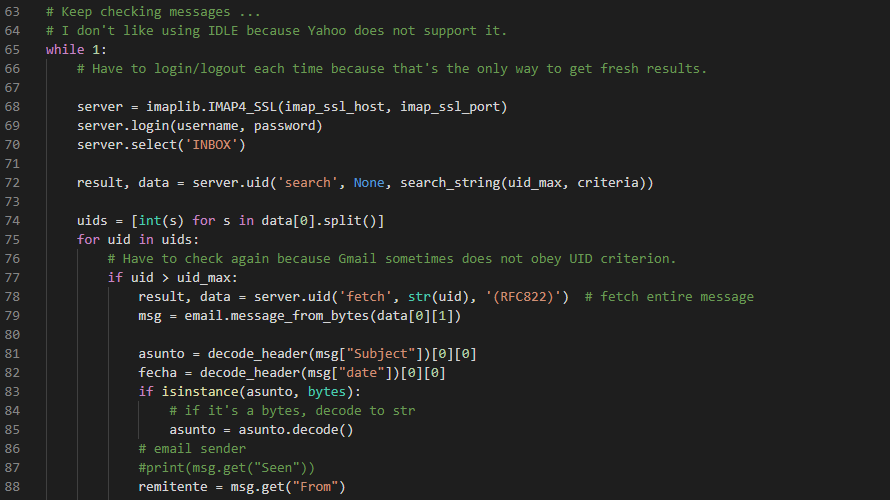
Entre las líneas 53 y 62, se realiza la conexión a la Base de Datos creada en Mysql, es por eso que primero se coloca “import pymysql” para realiza la conexión.

Entre las líneas 63 y 72, mientras se continua verificando los mails que se encuentran en la bandeja de entrada, para realizar esto el programa se va conectando y desconectado para refrescar la busqueda.

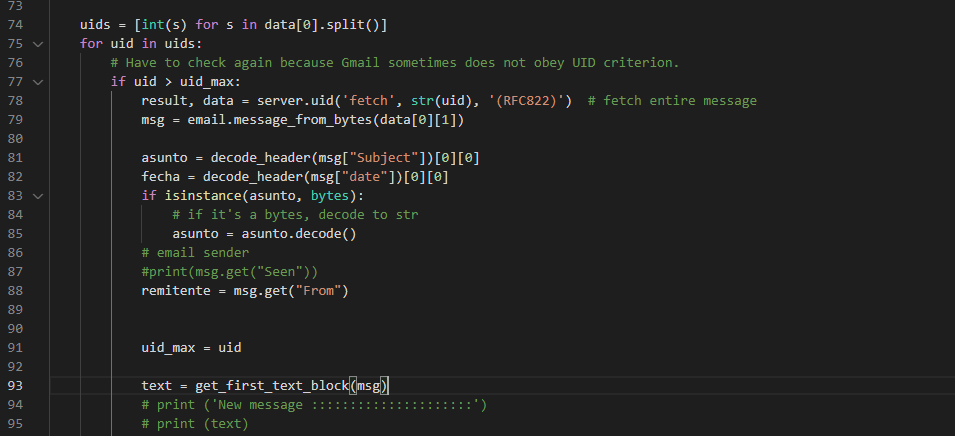
Esta conexión la hace mediante la colocacion del servidor stmp de gmail = **imaplib.IMAP4\_SSL(imap\_ssl\_host, imap\_ssl\_port)**

Luego, se indica que la busqueda continue sobre la carpeta “inbox”.

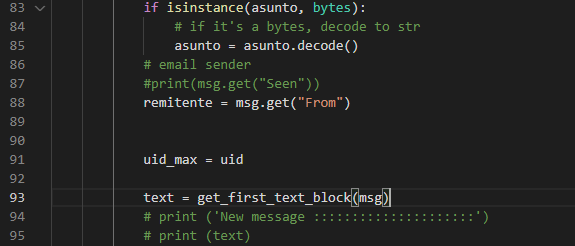
*\*\*Para obtener la informacion de como realizar la conexión a Gmail, consulte la pagina stackoverflow.*



Entre las lineas 73 y 82, se define que para los UIDS (los UIDS son los id que el programa le asigna a cada mail), que contengan la palabra “risk” en el cuerpo del mail, se obtenga los datos de remitente, la fecha y el asunto del mail, mediante la funcion fetch, la cual funciona tomando un argumento (la ruta del recurso que quieres obtener, en este caso el mensaje completo) y devuelve un objeto conteniendo la respuesta. Luego se detalla en cada variable *msg, asunto y fecha* lo que se debe extraer de cada mail.

*\*\*Para codificar esta parte del codigo tambien consulte en la pagina stackoverflow. Dado que no tenia del todo claro como realizar el filtro de la busqueda que va recorriendo los diferentes UIDS.*

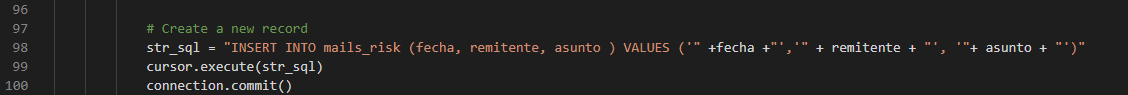
Entre las lineas 83 y 95, se utiliza la funcion “isinstance” para el dato *asunto*, la cual se usa para revisar los tipos de variable, en este caso se especifica que el tipo de dato sea string.

Luego, se especifica que sobre el dato *remitente,* se hace mediante la funcion “get”.

Entre las lineas 96 y 100, se realiza la conexión a la base de datos para grabar un nuevo registro en la base de datos anteriormente, según los parametros especificados.

Se utiliza la funcion “Insert Into” con el detalle de la tabla creada “mails\_risk”, especificando los campos a completar (los datos solicitados) y se aclaran los valores, respetando los nombres mencionados a lo largo del codigo.

*\*\*Me paso que varias veces me dio error el codigo por no respetar todo en espanol o todo en ingles.*

Mediante “cursor.execute()” se da la orden de ejecutar.

Entra las lineas 101 y 104, se desconecta del servidor y se utiliza la funcion ” time.sleep(1)” para suspender la ejecución durante 1 segundo.

