



La mini aplicación fue desarrollada en cuatro sesiones, es por ello, que la memoria se divide en cuatro capítulos.

CAPÍTULO 1:

El trabajo empieza con la incógnita de la posibilidad de enviar Gmail a través de Python y su interfaz (en este caso un módulo “import” de Python). Entonces nos reunimos mediante videoconferencia a través de la plataforma Discord. A continuación, buscamos en el navegador si existe algún módulo importable externo al propio Python que nos permitiera enlazar el servicio Gmail de Google con Python. De esta manera Óscar halla el módulo “smtplib” junto con el “ssl”, dos librerías conjuntas entre sí que permiten a través de unos parámetros conectar un módulo cualquiera Python a Gmail. Luego Diego creó el logo visible en la parte superior de este pdf, y en común elegimos el nombre de la aplicación. Con la idea pensada, se envía el correo con la estructura del trabajo y solo quedaba la parte de hacer el proyecto.

CAPÍTULO 2:

En nuestra segunda reunión telemática nos dedicamos a desarrollar la función más sencilla, que sería enviar un correo. Entonces nos pusimos a investigar cómo se hace, puesto que Gmail no acepta su uso a través de aplicaciones externas con la misma clave de la contraseña, por tanto, una vez dada con esta información creamos una cuenta de Gmail con el único propósito de ser usada durante el tiempo de vida de esta mini aplicación, en donde desde ajustes hicimos la verificación en dos pasos y generamos una contraseña mediante la cual, Sí se puede acceder a esa cuenta desde el módulo. Acto seguido, introducimos los datos en la función de cuenta y probamos a enviar un correo, que fue enviado satisfactoriamente. Para mayor seguridad, pedimos la dirección de varios amigos y confirmamos su uso. Una vez que tenemos la función para enviar correos nos pusimos a innovar distintas funciones, como la de que si pones distintos correos separados con un espacio envía al mismo correo a los distintos usuarios que se escribieron mediante el “input” descubierta por Óscar, además en un ataque de malicia Diego descubrió que mediante un bucle infinito “while” podíamos con únicamente la dirección de correo electrónico de alguien poder llenarle la bandeja de entrada del

mismo correo. Después de unas risas concluyó la segunda sesión por necesidades de estudio para los posteriores exámenes.

CAPÍTULO 3:

Una vez concertada la siguiente quedada para el desarrollo del proyecto, nos dividimos el trabajo en 3 partes. Diego creo los módulos base vacíos, los cuales serían repartidos para su posterior desarrollo. Óscar desarrollo el módulo “Login” y junto a Diego debatieron la posibilidad de hacer un “guardado de sesión” a través de ficheros con una idea primitiva de una idea de “banca” que íbamos a hacer antes de elegir hacer el “PyMail”. Jose Antonio mientras se informó acerca de interfaces gráficas para poder representar la imagen de algo parecido a “Pymail” en Python, que junto a la opinión consensuada conseguimos darle el diseño gráfico a través de “tkinter”, importado previamente de otra librería de Python. Diego mientras desarrollo el “SendMail” en un intento fallido de conseguir enviar un gran volumen de correos al unísono mediante bucles, idea rechazada posteriormente. Una vez avanzados los 3 módulos, se unieron todos en una misma carpeta, y se creó el botón “enviar” de la ventana de la interfaz. Óscar se dispuso entonces a darle una función a ese botón, descubriendo que no se podía enlazar a la función creada anteriormente para enviar correos debido a las restricciones del módulo en el que se basa el enlace Gmail-Python.

CAPÍTULO 4:

En la última sesión de trabajo, Óscar y Diego intentaron solucionar el problema adyacente de manera un tanto desesperada y agonizante. Óscar encontró un nuevo módulo llamado “email.message” con menos restricciones mediante el cual podíamos definir la nueva función. A partir de ahí definieron las nuevas funciones para enviar correos y se inició el delirio por enlazarlo a la interfaz. Tanto Óscar como Diego generaron anomalías cutáneas debido a que el orden de ejecución de funciones, los problemas con las variables de la interfaz que no podían ser llamadas más haya que en el módulo interfaz “Gmail.py”, y las múltiples ideas fracasadas que se probaron para conseguir enviar los correos no nos amenizaron la mañana. Después de múltiples intentos a base de ensayo y error, conseguimos que el programa enviase el correo, con la información de manera correcta y sin líneas de código que se enviaban junto al cuerpo del correo sin saber el porqué. Tras este hallazgo, tan solo nos quedaba enlazar ahora el nuevo botón “cerrar sesión”, que lo conseguimos sin mayor complicación. Ahora solo quedaba un pequeño detalle, retomar la idea de enviar el mismo correo a varias personas, pero esta vez a través de la interfaz.

Esto era gracias a la función Split que permite a través de poner espacios entre los correos electrónicos hacer que envíe el correo a cada uno de ellos. En este momento hubo ideas opuestas entre que el “.split()” iba antes de la función “send” por parte de

Diego y por otra parte Óscar defendía que el “.split()” iba dentro de la misma función. Funciono la idea sugerida por Diego, pero no podía ir todo bien, ya que solo envía la primera línea de lo que escribas en el apartado de “cuerpo”. El error se solucionó rápidamente, pero igual de rápido nos encontramos con otro error, solucionado corrigiendo la sintaxis de los bucles y... ahora sí, PyMail se había completado con éxito.

Una vez que por fin podíamos celebrar que habíamos logrado crear “PyMail”.

Una vez terminado el proyecto, Diego se encargó de escribir esta memoria además de el otro pdf para las normas de uso de la aplicación. Cervero mientras se encargó de escribir las descripciones de los módulos dentro de Python y el pdf equivalente en la carpeta adjunta.