

---

## Mensajería y Sentimientos

---

202201947 – Pablo Andres Rodriguez Lima

### Resumen

El proyecto consiste en el desarrollo de una API que permite la gestión de correos electrónicos, así como el análisis de hashtags, usuarios y sentimientos específicos previamente configurados en un archivo de configuración. Esta API proporciona una solución versátil y poderosa para las empresas y organizaciones que desean automatizar y optimizar la administración de su correo electrónico, así como realizar un seguimiento y análisis de datos relevantes en función de etiquetas, usuarios y emociones definidas en la configuración. Este enfoque permite una mayor eficiencia en la organización de la comunicación por correo electrónico y ofrece una comprensión más profunda de la interacción con los usuarios a través de las redes sociales y otras plataformas.

### Palabras clave

API  
Gestión de correos electrónicos  
Hashtags  
Análisis de usuarios  
Análisis de sentimientos

### Abstract

*The project involves the development of an API that enables the management of emails and the analysis of specific hashtags, users, and sentiments defined in a configuration file. This API provides a versatile and powerful solution for businesses and organizations looking to automate and optimize email management while tracking and analyzing relevant data based on labels, users, and emotions defined in the configuration. This approach allows for greater efficiency in organizing email communication and provides a deeper understanding of user interactions across social media and other platforms.*

### Keywords

API  
Email management  
Hashtags  
User analysis  
Sentiment analysis

## Introducción

Este proyecto tiene como objetivo fundamental la creación de una innovadora tecnología para el envío de mensajes encriptados, garantizando su seguridad frente a interceptaciones y desciframientos por parte de personal o instituciones no autorizadas.

El sistema diseñado consta de dos componentes clave: un componente emisor de mensajes y un componente receptor de mensajes expuestos en un archivo tipo XML.

Se presenta un listado de temas los cuales pueden ser de beneficio para entender el proyecto y un esquema del tipo diagrama de clases para entender el manejo del proyecto.

## Desarrollo del tema

### 1. ¿Qué es Python?

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

#### 1.1) ¿Cómo se utiliza Python?

El lenguaje Python se aplica a varios casos de uso en el desarrollo de aplicaciones, incluidos los ejemplos siguientes:

Desarrollo web del lado del servidor

El desarrollo web del lado del servidor incluye las funciones complejas de backend que los sitios web llevan a cabo para mostrar información al usuario.

Por ejemplo, los sitios web deben interactuar con las bases de datos, comunicarse con otros sitios web y proteger los datos cuando se los envía a través de la red.

Python es útil para escribir código del lado del servidor debido a que ofrece muchas bibliotecas que constan de código prescrito para crear funciones de backend complejas. Los desarrolladores también utilizan un amplio rango de marcos de Python que proporcionan todas las herramientas necesarias para crear aplicaciones web con mayor rapidez y facilidad. Por ejemplo, los desarrolladores pueden crear la aplicación web esqueleto en segundos porque no deben escribirla desde cero. Pueden probarla por medio de las herramientas de prueba del marco, sin depender de herramientas de prueba externas.

#### 1.2) Automatización con scripts de Python

Un lenguaje de scripting es un lenguaje de programación que automatiza las tareas que suelen llevar a cabo las personas. Los programadores utilizan ampliamente los scripts de Python para automatizar muchas tareas diarias.

#### 1.3) XML tree and elements

XML es un formato de datos inherentemente jerárquico, y la forma más natural de representarlo es con un árbol. ET tiene dos clases para este propósito: ElementTree representa todo el documento XML como un árbol, y Element representa un solo nodo en este árbol. Las interacciones con todo el documento (lectura y escritura desde/hacia archivos) generalmente se realizan en el nivel de ElementTree. Las interacciones con un solo elemento XML y sus subelementos se realizan en el nivel de Element.

## 2.1 Componentes de una API

### Flask

Flask es un microframework web de Python que se basa en la filosofía de "con menos es más". Esto significa que Flask proporciona una funcionalidad básica para crear aplicaciones web, pero deja que los desarrolladores se encarguen de implementar la lógica de negocio y la presentación.

Flask es una buena opción para desarrolladores que quieren crear aplicaciones web simples y eficientes. Es un framework ligero que es fácil de aprender y usar.

### Django

Django es un framework web de Python que es más completo que Flask. Django proporciona una gran cantidad de funcionalidad incorporada, como autenticación, autorización, administración y ORM.

Django es una buena opción para desarrolladores que quieren crear aplicaciones web complejas. Es un framework maduro y robusto que proporciona una base sólida para el desarrollo de aplicaciones web.

### API

Una API es un conjunto de funciones y procedimientos que permiten a los desarrolladores de software interactuar con un sistema o aplicación. Las APIs se utilizan para una amplia gama de propósitos, como la integración de sistemas, el desarrollo de aplicaciones móviles y el acceso a datos.

Hay dos tipos principales de APIs: APIs REST y APIs SOAP. Las APIs REST son las más comunes y se basan en el protocolo HTTP. Las APIs SOAP se basan en el protocolo SOAP, que es más complejo que HTTP.

## Backend

El backend es la parte de una aplicación web que se ejecuta en el servidor. El backend es responsable de procesar las solicitudes de los usuarios, acceder a los datos y generar las respuestas.

El backend suele estar escrito en un lenguaje de programación del lado del servidor, como Python, Java o PHP. El backend también puede utilizar una base de datos para almacenar datos.

## Frontend

El frontend es la parte de una aplicación web que se ejecuta en el navegador del usuario. El frontend es responsable de mostrar la interfaz de usuario de la aplicación y de interactuar con el backend.

El frontend suele estar escrito en HTML, CSS y JavaScript. HTML define la estructura de la página web, CSS define el diseño de la página web y JavaScript agrega interacción a la página web.

## Postman

Postman es una herramienta de desarrollo de API que se utiliza para probar y documentar APIs. Postman permite a los desarrolladores enviar solicitudes a una API, recibir respuestas y visualizar los resultados.

Postman es una herramienta gratuita y de código abierto que está disponible para Windows, macOS y Linux.

## HTML

HTML es el lenguaje de marcado que se utiliza para crear el contenido de una página web. HTML define la estructura y el diseño de una página web.

HTML es un lenguaje sencillo que es fácil de aprender. Se basa en etiquetas que se utilizan para definir el contenido de la página web.

## JavaScript

JavaScript es un lenguaje de programación que se utiliza para agregar interacción a las páginas web. JavaScript se utiliza para crear efectos visuales, animaciones y juegos.

JavaScript es un lenguaje potente que se puede utilizar para una amplia gama de propósitos. Es un lenguaje de programación del lado del cliente, lo que significa que se ejecuta en el navegador del usuario.

### 2.2 Configuración

Para configurar un widget, simplemente llamamos a `.config()` y pasamos los argumentos que queramos modificar. Algunas opciones son:

`bg`: modifica el color de fondo. Se puede indicar con el color en inglés (incluyendo modificadores, como “darkgreen”) o su código RGB en hexadecimal (“#aaaaaa” para blanco). Ojo: en MacOS no se puede modificar el color de fondo de los botones; aunque indiquemos un nuevo color, se mostrará en blanco. Lo más parecido que podemos hacer es configurar el `highlightbackground`, que pintará el fondo alrededor del botón del color que indiquemos.

`fg`: cambia el color del texto.

`cursor`: modifica la forma del cursor. Algunos de los más utilizados son “gumby”, “pencil”, “watch” o “cross”.

`height`: altura en líneas del componente.

`width`: anchura en caracteres del componente.

`font`: nos permite especificar, en una tupla con nombre de la fuente, tamaño y estilo, la fuente a utilizar en el texto del componente. Por ejemplo, `Font(“Times New Roman”, 24, “bold underline”)`.

`bd`: modificamos la anchura del borde del widget.

`relief`: cambiamos el estilo del borde del componente. Su valor puede ser “flat”, “sunken”, “raised”, “groove”, “solid” o “ridge”.

`state`: permite deshabilitar el componente (`state=DISABLED`); por ejemplo, una Label en la que no se puede escribir o un Button que no se puede clicar.

`padding`: espacio en blanco alrededor del widget en cuestión.

`command`: de cara a que los botones hagan cosas, podemos indicar qué función ejecutar cuando se haga click en el mismo.

### 2.4 Gestión de la composición

Es muy importante que, cuando tengamos configurado el componente, utilicemos un gestor de geometría de componentes. Si no, el widget quedará creado, pero no se mostrará.

## Conclusiones

El proyecto tiene como vitalidad el uso correcto del manejo de los drones de tal manera que cada uno pueda tener su propio contenido.

Cada dron puede tener sus instrucciones de subir y bajar de tal manera que pueda ser mas optimo los tiempos de rastreo de mensajes.

La interfaz grafica puede ayudar y que el uso se de una manera mas intuitiva ayudando que todo sea mas amigable con el usuario.

## Referencias bibliográficas

Gomez, O. A. (s/f). Visualizando Grafos usando Graphviz. OSiUX. Recuperado el 4 de septiembre de

2023, de <https://osiux.com/visualizando-grafos-graphviz.org>

Londoño, P. (2023, abril 26). Qué son las clases en Python, para qué sirven y cómo funcionan. Hubspot.es. <https://blog.hubspot.es/website/clases-python>

Pérez, J. R. C. (2020, noviembre 21). Implementando una lista doblemente ligada en Python. Medium. <https://a01153884.medium.com/implementando-una-lista-doblemente-ligada-en-python-b8def2b8df73>

xml.etree.ElementTree — The ElementTree XML API. (s/f). Python documentation. Recuperado el 4 de septiembre de 2023, de <https://docs.python.org/3/library/xml.etree.elementtree.html>

(S/f). Amazon.com. Recuperado el 4 de septiembre de 2023, de <https://aws.amazon.com/es/what-is/python/>

## APENDICES

p3\_202201947

### SECCION CARGA DE ARCHIVOS:

Seleccionar archivo Sin archivos seleccionados Cargar Archivo Mensaje  
Seleccionar archivo Sin archivos seleccionados Cargar Archivo Sentimiento

### OPERACIONES DE CONSULTA:

Consultar Hashtag Consultar Menciones Consultar Sentimiento Grafica Hashtag Grafica Menciones Grafica Sentimiento

### OPERACIONES:

Documentacion Info Del estudiante Inicializar Sistema

