

Documentación del Script de Inicio de la Aplicación de Envío de Correos

Este documento explica el funcionamiento del script de inicio de la aplicación web de envío de correos. Se proporcionarán detalles sobre cada línea del código y se indicará cómo realizar modificaciones en el mismo.

Instalación de Dependencias

Antes de ejecutar la aplicación, es necesario instalar las dependencias requeridas. Para ello, se debe ejecutar el siguiente comando en la terminal:

```
python3 install_requirements.py
```

Este script instala todas las librerías necesarias para que la aplicación funcione correctamente. Asegúrate de tener **Python 3** instalado en tu sistema antes de ejecutarlo.

Para Windows, se recomienda ejecutar este comando en **PowerShell** o en la terminal de **CMD**.

Inicio de la Aplicación

En sistemas Windows, después de instalar las dependencias, se debe ejecutar el siguiente archivo para iniciar la aplicación:

```
run_app.bat
```

Esto iniciará la aplicación de manera automática en un entorno Windows.

Para sistemas Linux o Mac, se debe utilizar el siguiente comando:

```
bash start_app.sh
```

Esto ejecutará el servidor y dejará la aplicación corriendo en segundo plano.

Organización del Código

Cada uno de los proyectos (**Estados**, **Sellos** y **Usabilidad**) está muy bien estructurado. Dentro de cada archivo principal (**estado.py**, **sellos.py**, **usabilidad.py**) se encuentran las funciones organizadas de manera clara y bien documentada, facilitando su comprensión y mantenimiento.

Las funciones están claramente separadas según su propósito, incluyendo:

- **Configuración de la aplicación:** Donde se inicializa **FastAPI** y se configuran rutas y variables necesarias.
- **Funciones auxiliares:** Métodos que realizan procesamiento de datos, validaciones o transformaciones.
- **Rutas principales:** Endpoints de la API para subir archivos, previsualizar datos y enviar correos.
- **Gestión de correo:** Funciones para la generación y envío de emails automáticos.

Cada módulo sigue una estructura clara y ordenada, con comentarios explicativos en cada parte importante del código, asegurando que cualquier desarrollador pueda entender y modificar el sistema con facilidad.

Proyecto de Estados

El proyecto cuenta con un módulo llamado **estado.py**, el cual se encarga de manejar la carga y procesamiento de archivos Excel, así como la generación y envío de correos electrónicos automáticamente.

Principales Características:

- Carga de archivos Excel y procesamiento de datos.
- Generación de informes de estado basados en los datos.
- Uso de **FastAPI** para proporcionar una interfaz web.
- Uso de **Jinja2** para renderizar plantillas HTML.
- Integración con SMTP para envío de correos automáticos.

Archivos y Carpetas Importantes:

- **estado.py:** Archivo principal que maneja la lógica del procesamiento de datos y envío de correos.
- **templates_estado/:** Carpeta donde se encuentran las plantillas HTML usadas para la interfaz web y los correos.
- **uploads/:** Directorio donde se almacenan temporalmente los archivos subidos por los usuarios.

Rutas Principales:

- `/`: Página principal con formulario para subir archivos.
 - `/status`: Verifica el estado del sistema.
 - `/upload`: Permite subir un archivo Excel y procesarlo.
 - `/preview`: Muestra los datos cargados antes de enviarlos.
 - `/select_account`: Permite seleccionar la cuenta desde la cual se enviarán los correos.
 - `/confirm_send`: Muestra la vista previa de los correos antes del envío.
 - `/send_emails`: Realiza el envío de correos a los destinatarios.
-

Proyecto de Sellos

El proyecto de **Sellos** tiene una estructura similar al de Estados, pero está enfocado en la asignación y validación de sellos en los reportes.

Características Principales:

- Carga y validación de archivos Excel.
 - Verificación y asignación de sellos.
 - Uso de **FastAPI** para manejar la aplicación web.
 - Integración con SMTP para el envío de correos.
-

Proyecto de Usabilidad

El proyecto de **Usabilidad** está diseñado para analizar y optimizar la experiencia de usuario a partir de datos recopilados.

Características Principales:

- Procesamiento y análisis de datos de usabilidad.
- Integración con **FastAPI** para visualización y envío de informes.
- Uso de **Jinja2** para generar reportes y vistas personalizadas.
- Sistema de envío de correos automáticos con **SMTP**.