

Kick-off: Tarea 1 de Redes

Jesús Gabriel Cordero Díaz

March 5, 2025

1 Introducción

1.1 ¿Qué es?

Este documento *Kick-off* presenta de forma breve el problema a resolver y la estrategia general para la Tarea 1 de Redes. El objetivo es construir un servidor de correo (que soporte **SMTP**, **IMAP** y **NNTP**) utilizando la librería **Twisted** en Python, añadiendo también una capa segura **SSL/TLS** y manejo de adjuntos **MIME**.

1.2 ¿Qué incluye?

- **Descripción general del proyecto:** Se explica la necesidad de un servidor de correo, con énfasis en la lógica de envío (SMTP) y recepción (IMAP).
- **Alcance:** Protocolos y funcionalidades requeridas (SMTP, IMAP, NNTP, adjuntos, notificaciones y dominio propio).
- **Justificación:** Comprender la implementación de estos protocolos y la configuración de correo seguro (capa SSL/TLS) enriquece la formación y competencias en ingeniería de redes.

1.3 Estrategia inicial

La idea principal es dividir el desarrollo en módulos sencillos que intercambian información:

- **SMTP_Server:** Para recibir y guardar correos.
- **SMTP_Client:** Lector de CSV para envíos masivos y personalizados.
- **IMAP_Server:** Para que el usuario pueda leer su buzón.
- **NNTP_Notifier:** Para notificaciones en un servicio de noticias.

A continuación, se muestra el diagrama simplificado de la estructura del sistema:

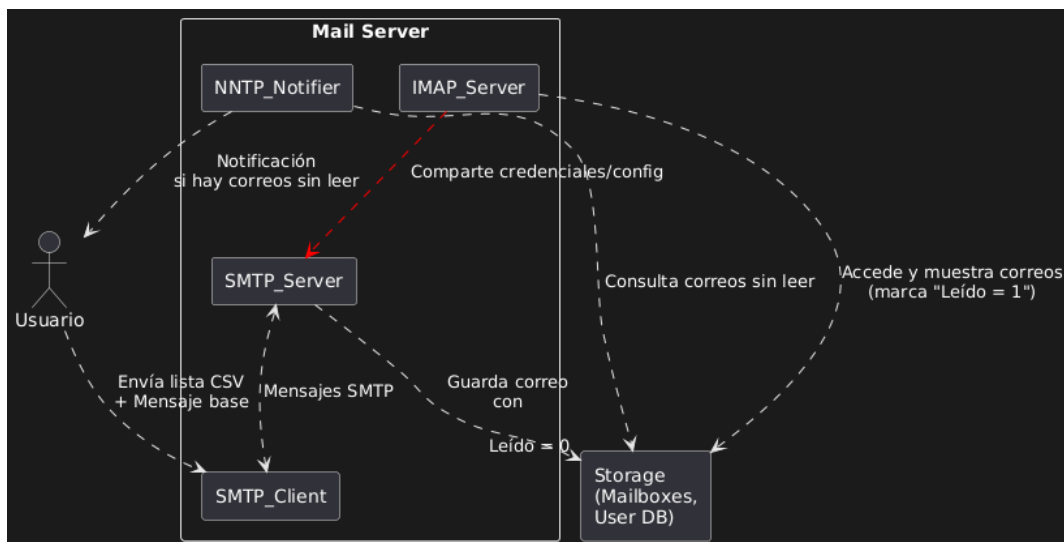


Figure 1: Diagrama simplificado del sistema de correo.

2 Ambiente de desarrollo

2.1 Herramientas

- **Lenguaje:** Python 3.x
- **Bibliotecas:** Twisted (para servicios de red), OpenSSL (para SSL/TLS)
- **Otros:** Posible uso de `pytest` para pruebas unitarias y un cliente de correo (Thunderbird, por ejemplo) para validar IMAP.

2.2 IDE / Debugging

Se emplearán los siguientes entornos de desarrollo:

- **VSCode:** Editor principal y depuración rápida.
- **PyCharm:** Opcionalmente, para debugging más avanzado y pruebas unitarias.

2.3 Flujo de trabajo

El desarrollo será incremental y manual en su mayoría:

1. Implementar pequeña funcionalidad (ej. `SMTP_Server`).
2. Probar manualmente enviando correos de prueba o `scripts` sencillos.
3. Integrar con otras capas (IMAP, NNTP) y repetir el proceso.
4. Ajustar y refinar de acuerdo con los resultados.

3 Control de versiones

3.1 Repositorio

Se usará **Git** como sistema de control de versiones.

3.2 Enlace al repo

https://github.com/jesusgcd/redes_tarea1

Se ha otorgado acceso al usuario **kmoragas**.

3.3 Ventajas

- Seguimiento detallado de cambios (commits).
- Colaboración más ágil, incluso si el proyecto crece.
- Posibilidad de revertir a versiones anteriores en caso de errores serios.
- Historial de trabajo que evidencia el progreso y la dedicación al proyecto.