



INSTITUTO TECNOLÓGICO DE COSTA RICA  
INGENIERÍA EN COMPUTACIÓN  
Redes

## **Documentación Tarea 1: e-Mail**

Ricardo Murillo Jiménez - 2018173697

Profesor:  
Kevin Moraga García

**Sede Inter-Universitaria Alajuela  
5 de Septiembre 2021**

# Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Ambiente de desarrollo</b>	<b>4</b>
2.1	Lenguaje Programación: Python . . . . .	4
2.2	Principal Biblioteca a usar . . . . .	4
2.3	Navegador: Predeterminado de Ubuntu . . . . .	4
2.4	Sistema Operativo . . . . .	4
2.5	Especificaciones de Computadora . . . . .	4
2.6	Bash . . . . .	5
2.7	Editor de texto . . . . .	5
2.8	Forma de Debugging . . . . .	5
2.9	Control de Versiones . . . . .	5
<b>3</b>	<b>Estructuras de datos usadas y funciones</b>	<b>6</b>
<b>4</b>	<b>Instrucciones para ejecutar el programa</b>	<b>7</b>
<b>5</b>	<b>Actividades realizadas por estudiante</b>	<b>8</b>
<b>6</b>	<b>Autoevaluación</b>	<b>9</b>
6.1	Estado final del programa . . . . .	9
6.2	Problemas encontrados . . . . .	9
6.3	Limitaciones adicionales . . . . .	9
6.4	Reporte de commits de git . . . . .	9
6.5	Evaluación . . . . .	10
6.6	AutoEvaluación . . . . .	10
<b>7</b>	<b>Lecciones Aprendidas</b>	<b>11</b>

# 1 Introducción

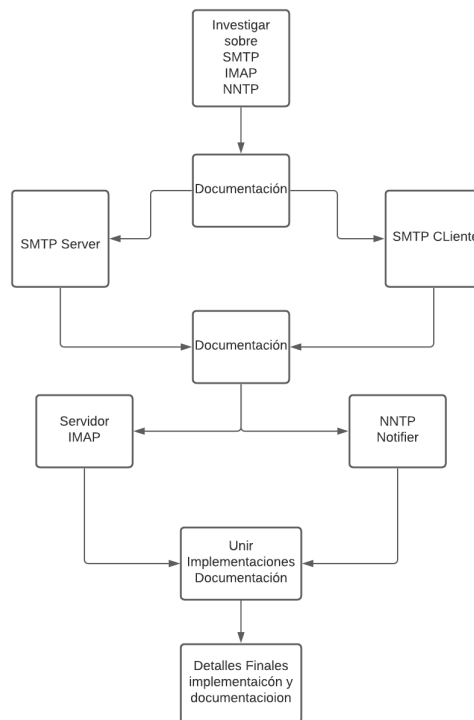
El uso del correo postal para el envío de cartas o mensajes entre 2 o varias personas, con la llegada de Internet, se puede decir que quedó como un método poco práctico, debido a la cantidad de mensajes que se transmiten en cada momento y la necesidad de inmediatez que se empezó a crear entre los usuarios de Internet.

Es por esto que se creó el protocolo SMTP o "Protocolo para transferencia simple de correo". Con este lo que se pretende es tener un protocolo que sirva para enviar correos desde un dispositivo hasta un servidor de correos remoto. Se creó además otro protocolo llamado IMAP con el fin de lograr leer o recuperar los archivos almacenados en los servidores y que fueron enviados probablemente mediante el SMTP. Por lo que se creará una herramienta con la cual se podrá realizar el envío y recibo de correos electrónicos mediante el protocolo SMTP

Por lo que para la resolución de esta tarea se tratará de seguir el siguiente esquema de trabajo(Gráfico):

- Investigar sobre las tecnologías, protocolos y software que se podría llegar a utilizar; poniendo especial atención a: SMTP(cliente-servidor), IMAP y NNTP que son los principales protocolos para la realización de la presente tarea.

Seguidamente se comenzará a trabajar en la documentación a la vez que en la implementación del cliente y el servidor del SMTP. Una vez bien avanzado esto, se trabajará de nuevo en la documentación. Para después seguir con los otros dos requerimientos Funcionales: Servidor IMAP y NNTP Notifier. Finalmente se finalizarán detalles de la implementación y la documentación misma.



## 2 Ambiente de desarrollo

### 2.1 Lenguaje Programación: Python

Python 3.6.8

### 2.2 Principal Biblioteca a usar

Twisted 21.2.0

### 2.3 Navegador: Predeterminado de Ubuntu

Mozilla Firefox 91.0 (64 bits) con google como motor de búsqueda

### 2.4 Sistema Operativo

ubuntu-20.04.2.0-desktop-amd64

### 2.5 Especificaciones de Computadora

PC Desktop

CPU: CPU AMD 5 3600

GPU: Radeon™ RX 580

RAM: 8000MB

## **2.6 Bash**

5.0.17(1)-release (x86\_64 - pc - linux - gnu)

## **2.7 Editor de texto**

Visual Studio Code - versión 1.59

## **2.8 Forma de Debugging**

Consola de depuración de Visual Studio Code

## **2.9 Control de Versiones**

Se usará la herramienta Git.

Primeramente se creará una rama "dev" para llevar los principales cambios y versión principal de la tarea. Una vez se tenga una versión más estable se Mergeará de "dev" a "main".

Se crearán además ramas hijas de "dev" para cada subproblema importante, o para tratar con problemas que vayan apareciendo en el desarrollo.

### 3 Estructuras de datos usadas y funciones

- Función CONNECT: Esta función esta presente en ambos clientes, es la encargada de conectar al servidor en base al url y el puerto que tomó de la entrada de usuario
- Función attend client: Presente en ambos tipos de servidores, se encarga de crear un hilo (si es prethread) o un proceso (si es preforked) para atender a un client.
- Función cd: El FTPServer implementa el comando cd para buscar en un directorio.
- Función get: El FTPServer implementa el comando get para obtener un archivo.
- Función put: El FTPServer implementa el comando put para poner un archivo.
- Función quit: El FTPServer implementa el comando quit para salir del FTPServer.
- Función receive clients: Presente en ambos servidores, se encarga de manejar los procesos o hilos (dependiendo si es prethread o preforked) para que acepten a los clientes
- STRUCT command: Este struct contiene un id, una dirección y un nombre de archivo, con estos datos se podía comunicar al servidor cual comando esta solicitando y con que datos se va a realizar desde la entrada del usuario.
- STRUCT packet: Este struct contiene entre sus datos una bandera, el id del comando, el largo de los datos, y los datos. Con esta información tanto el cliente como el servidor, se lograban comunicar datos de entrada y salida.
- Struct sockaddr\_in: Proveniente de socket.h, este facilitaba la creación de la dirección y el puerto, a donde el socket se va a conectar.

## 4 Instrucciones para ejecutar el programa

Las instrucciones para ejecutar son las siguientes:

- Inicializar el Servidor, escogiendo las variables para el storage que se decida.
- Correr el cliente, mediante la GUI, completar los datos, y envíar el correo.

## 5 Actividades realizadas por estudiante

Lista de Actividades		
Actividad	Día	Cantidad Horas
Investigación inicial	25/08/2021	1 hrs
SMTP/IMAP/NNTP/SSL		
Desarrollo ideas Kick-Off	25/08/2021	15 min
Desarrollar el Kick-Off-Diagrama	26/4/2021	1 hrs
Detalles y entrega	26/4/2021	20 min
Base de Documentación	30/08/2021	20 min
Investigación SMTP-servers-clientes	30/08/2021	1 hrs
Inicio SMTP-Server y verificación ambiente desarrollo	30/08/2021	3 hrs
Corrección errores y server-SMTP esperando conexión	31/08/2021	1 hrs
Investigación SMTP-Clientes	31/08/2021	1 hrs
Inicio código SMTP-Clientes	31/08/2021	2 hrs
Errores y continuación Cliente(se logra conexión servidor)	01/09/2021	5 hrs
Investigación IMAP	02/09/2021	1 hrs
Inicio código IMAP	02/09/2021	3 hrs
Comentar Código y arreglar Hard-Codeo server-cliente	03/09/2021	3 hrs
Documentación	04/09/2021	2 hrs
Código IMAP	04/09/2021	2 hrs
Tratar Errores IMAP	05/09/2021	4 hrs



## **6 Autoevaluación**

### **6.1 Estado final del programa**

Básicamente se logró implementar el servidor SMTP y cliente SMTP. Se logró el desarrollo de una interfaz para el cliente SMTP y la correcta presentación de la presente documentación.

### **6.2 Problemas encontrados**

Entre los problemas encontrados se tienen:

Debido a la vejez de la biblioteca, muchos ejemplos están solamente para Python 2, versión poco usada en la actualidad. Y aunque Twisted se está moviendo a Python 3, no todo está como en las versiones anteriores para python 3, lo que complica un poco el informarse correctamente.

### **6.3 Limitaciones adicionales**

La falta de tiempo debido a otras materias, actividades de la Universidad y extracurriculares, junto a un poco de falta de organización, derivaron en la falta de tiempo para la total implementación del sistema.

### **6.4 Reporte de commits de git**

Los commits realizados para la presente Tarea se pueden encontrar en el siguiente repositorio de GitHub:

<https://github.com/ricardomj0499/Tarea1-Redes-e-Mail>

## 6.5 Evaluación

Evaluación	
Kick-off	5
SMTP-Server	15
SMTP-Client	15
IMAP-Server	0
NNTP-Notifier	0
Modo SSL IMAP y SMTP server	0
Documentación	20
OPC1: envío, recepción, revisión con PGP/MIME	2
OPC2: GUI para Cliente- SMTP	5

## 6.6 AutoEvaluación

Autoevaluación	
Aprendizaje	Puntaje
Aprendizaje del protocolo SMTP	9
Aprendizaje del protocolo IMAP	6
Aprendizaje del protocolo NNTP	7
Aprendizaje de la capa SSL/TLS	2
Organización de Tiempo	5

## 7 Lecciones Aprendidas

Gracias a la implementación de la presente Tarea logré conocer desde cero como es el funcionamiento de un e-Mail básico mediante el protocolo SMTP y los otros mencionados a lo largo de la tarea: IMAP, NNTP. Esto ya que realmente a lo largo de la carrera nunca ví algo relacionado directamente a lo que son las redes de correo electrónico o protocolos de este estilo.

Adicionalmente, gracias a la implementación de la biblioteca Twisted de python, la cual aunque ayuda en gran manera a la implementación de la tarea, pude observar la complejidad que hay detrás de todo este tema.

Hay que mencionar también a los otros dos protocolos citados(IMAP, NNTP), y los cuales ayudan a comprender mejor todo los procesos necesarios para lo que es el envío y recibo de correos a lo largo de red, así como tampoco olvidar todo lo relacionado a lo que es la seguridad necesaria para estos temas.

Más personalmente puedo decir que tuve que aprender a controlar un poco mejor el tiempo que se le dedica a cada tarea en este semestre. Ya que surgieron problemas debido a mala organización del tiempo.

## References

- [1] “Historia de la tecnología: el lenguaje de programación c,” 2011. Web; accedido 18-12-2020.
- [2] “Programación estructurada,” 2020. Web; Accedido el 18-12-2020.
- [3] “Allegro,” 2020. Web; Accedido el 19-12-2020.