

Carrera: Tecnicatura en Programación

Proyecto: Bottled Message (botme)

Modelo Futuro

Autores:

Arregín Gonzalo

Arregín Leonardo

Fleming Nilo

Director y revisor:

Angel Leonardo Bianco

Fecha de creación: 20/09/2025

Fecha de actualización: 20/09/2025

Versión: 1.0

Índice

[**Resumen 2**](#_au51mny0sx6)

[**Introducción 3**](#_3at9u9s4e0vp)

[**Hitos 4**](#_yyrhu7ml5bea)

[I. Entregable 1: Modelado de clases y encapsulamiento 4](#_buwz1tcz7y35)

[II. Entregable 2: Estructura de datos y recursividad 4](#_p2nityf5kx5q)

[III. Entregable 3: Algoritmos y funcionalidades avanzadas 4](#_qsitov6j3m4w)

[IV. Entregable 4 (Final): Integración, presentación y defensa 5](#_xr06jmf74uf3)

[**Cronograma de entregas 5**](#_3etj4y9g1gtr)

[**Especificaciones 6**](#_4p7xi5bvhxdr)

[Clase InterfazMensajeria 6](#_56kfpodyq5td)

[Clase ServidorCorreo 6](#_kxqyslz3osyf)

[Clase Mensaje 7](#_y2o14wkxccs1)

[Clase Carpeta 7](#_ovymvtl6mbzu)

[Clase Usuario 7](#_dw3tfqq3gosa)

[**Puntos Pendientes 8**](#_638dmx9s5rzy)

[**Documentos relacionados 8**](#_59vttlvncprq)

[Documento de Especificación Técnica 8](#_w3kb6fwon5li)

[Documento de Pruebas Funcionales 8](#_uns4m3cguxf5)

[**Bibliografía 8**](#_umoei3bosjkx)

# Resumen

Este documento presenta el proyecto **Botted Message** (**botme**), que consiste en el diseño e implementación de un sistema de software, desarrollado en **Python**, para modelar una aplicación de mensajería.

El objetivo principal del desarrollo es aplicar los conceptos de la Programación Orientada a Objetos (POO), diversas estructuras de datos y diferentes algoritmos, para implementar con el mismo, las operaciones típicas de un entorno de mensajería.

Este software permitirá diferentes acciones, que contemplan desde el registro de usuarios, autenticación, envío y recepción de mensajes, organización por carpetas, uso de filtros y visualización de conversaciones.

El modelo incluirá clases encapsuladas como InterfazMensajeria, Usuario, Mensaje, Carpeta y ServidorCorreo, entre otras, cada una con responsabilidades bien definidas que serán detalladas en el documento de especificaciones.

La arquitectura modular utilizada permitirá escalar el sistema con funcionalidades avanzadas como filtros automáticos, prioridades, estructuras recursivas, pasibles de ser adaptadas para nuevos desafíos o mejoras.

Además, contará con una interfaz de línea de comandos intuitiva, que guíe al usuario en sus operaciones básicas.

# Introducción

A lo largo de este documento, se detallarán las decisiones de diseño y las implementaciones de cada componente del sistema, ordenados en los entregables establecidos para el proyecto.

Se explorará cómo la POO nos permite crear un código modular y escalable, cómo las estructuras de datos como los árboles y grafos, facilitan una organización jerárquica de la información, y cómo los algoritmos de búsqueda y priorización optimizan la eficiencia en el rendimiento.

El resultado es una aplicación que no solo cumple con las funcionalidades básicas de un cliente de correo, sino que también establece las bases para una herramienta avanzada y eficiente, demostrando que los principios de la POO son la mejor herramienta de desarrollo de software para este tipo de aplicaciones.

# Hitos

A continuación se enumeran los hitos o entregables relacionados al proyecto **botme**, según el cronograma y calendario establecido (informado en la página web de la universidad, y en los medios de comunicación de la cátedra - No informados en este documento -).

## Entregable 1: Modelado de clases y encapsulamiento

Este entregable presenta las definiciones de las clases principales, y el encapsulamiento de sus métodos y atributos, basados en los principios de la POO.

Por otro lado, se establece el diseño de alto nivel de sus interfaces principales y la documentación de referencia que sustenta el modelo elegido para el desarrollo de la solución.

## Entregable 2: Estructura de datos y recursividad

En el segundo entregable se definen las estructuras de datos y recursividad de los procesos para la implementación de Carpetas y Subcarpetas con una estructura diseñada con el modelo de “árbol”.

Gestión y búsqueda de mensajes con estructuras recursivas, y análisis de eficiencia de las operaciones implementadas.

## Entregable 3: Algoritmos y funcionalidades avanzadas

Implementación de reglas definidas por el usuario para la utilización de filtros y diccionarios que ayudarán a la organización de mensajes dentro de las carpetas existentes.

Utilización de grafos para diseñar la red de servidores de correo, y simulación de envío de mensajes entre servidores.

## Entregable 4 (Final): Integración, presentación y defensa

Integración de todas las funcionalidades mediante una interfaz de usuario para la interacción de los mismos con software.

Documentación final del diseño del proyecto (diagramas, código, pruebas realizadas, etc.) mediante la herramienta GitHub para facilitar posibles modificaciones futuras.

# Cronograma de entregas

| Etapa | Fecha de entrega | Estado |
| --- | --- | --- |
| Entregable 1 | 20-09-2025 | Pendiente |
| Entregable 2 | 11-10-2025 | No iniciada |
| Entregable 3 | 01-11-2025 | No iniciada |
| Entregable 4 (final) | 15-11-2025 | No iniciada |

# Especificaciones

A continuación se enumeran las clases definidas dentro de la solución, con la especificación funcional y la justificación de diseño de cada una de ellas, la especificación técnica se encuentra desarrollada en el documento de especificaciones.

## Clase InterfazMensajeria

La clase InterfazMensajeria será abstracta, al igual que sus métodos, y hará de interfaz de envío, recepción y listado de mensajes de otras clases, para cumplir con las funcionalidades brindadas.

Las clases dedicadas a realizar estos procesos, son las que serán instanciadas y contendrán el código necesario de cada uno de los procedimientos a realizar.

Esto asegura la neutralidad de la interfaz, haciéndola reutilizable y adaptable a posibles cambios, sin la necesidad de tener que realizar modificaciones sustanciales en la misma.

## Clase ServidorCorreo

La clase ServidorCorreo será responsable de realizar la registración de nuevos usuarios, realizando las validaciones correspondientes, para que no se dupliquen datos a la hora del registro del alta, almacenando nombre de usuario (como clave única) y contraseña.

También contendrá la funcionalidad de la autenticación de usuarios existentes, cuando los mismos quieran acceder a la aplicación de mensajería.

Con esto se logra garantizar la seguridad y la integridad de los datos de usuario de manera eficiente.

## Clase Mensaje

La clase Mensaje será responsable de gestionar toda la información relacionada a los mensajes creados por los usuarios, y representa cada mensaje individualmente dentro del sistema.

Esto permite el encapsulamientos de los datos relacionados a los mismos de forma individual, cómo por ejemplo remitente, destinatario, asunto, contenido, etc.

Con esto se logra que todos los mensajes creados dentro del sistema tengan la misma estructura, sean coherentes para la interpretación dentro del mismo y puedan ser creados en cualquier momento sin necesidad de lógicas diferenciadas.

## Clase Carpeta

La clase Carpeta hará de contenedor de mensajes para facilitar la organización de los mismos, existirán carpetas por defecto como por ejemplo Enviados, Recibidos, Papelera, Borradores, pero además cada usuario podrá crear y gestionar carpetas personalizadas de acuerdo a cada necesidad.

## Clase Usuario

La clase Usuario representará un usuario registrado dentro del sistema, administrará todos los datos relacionados al mismo como por ejemplo el inicio de sesión, o las carpetas personales que cada uno tenga.

Será la responsable de recibir y enviar los mensajes que le lleguen mediante la interfaz desde los diferentes usuarios de la aplicación.

Con esto se logra centralizar la lógica de gestión de mensajes enviados y recibidos, haciendo la solución modular y escalable

# Puntos Pendientes

| ID | Fecha resolución | Responsable | Estado | Comentario |
| --- | --- | --- | --- | --- |
| 1 | 20-11-2025 | Equipo 3 | Pendiente | Se creará al momento de comienzo de pruebas |
|  |  |  |  |  |

# Documentos relacionados

## Documento de Especificación Técnica

Especificacion\_tecnica\_botme\_v1.0

## Documento de Pruebas Funcionales

Pendiente de creación

# Bibliografía

* Martin Fowler y Scott Kendall (1999), UML Gota a gota, Pearson Education.