

Manual de Usuario – **Buscador de Correos** **(Buscador.cpp)**

1. Introducción

Este programa permite gestionar y buscar correos electrónicos almacenados en un archivo de texto o cargados directamente desde el programa. Ofrece una interfaz por consola con colores ANSI para una visualización más clara.

El usuario puede: - Ver correos ordenados por fecha. - Buscar correos por remitente. - Buscar correos por palabra clave. - Abrir y leer un correo específico.

Se organiza internamente usando: - Árbol binario de búsqueda (para ordenar por fecha). - Mapas hash (para búsqueda por remitente e ID). - Matriz dispersa (para búsqueda por palabras clave en asunto/cuerpo).

2. Requisitos del Sistema

- Sistema operativo: Windows (usa `cls` para limpiar pantalla).
- Compilador C++ compatible con C++11 o superior.
- Archivo externo opcional: `correos.txt` con formato:

`remitente; asunto; cuerpo; fecha`

3. Ejecución del Programa

Compila el archivo como:

```
g++ Buscador.cpp -o Buscador
```

Ejecuta:

```
./Buscador
```

Al iniciar, se cargan correos predeterminados y luego se intenta leer el archivo `correos.txt`.

4. Menú Principal

Al ejecutar el programa, se muestra:

[MENU PRINCIPAL]

1. Ver correos ordenados
2. Buscar por remitente
3. Buscar por palabra clave
0. Salir

El usuario selecciona una opción ingresando el número correspondiente.

5. Funciones del Menú

5.1 Ver correos ordenados por fecha

Muestra todos los correos ordenados ascendentemente según la fecha.

Vista general:

ID	Remitente	Asunto	Fecha
----	-----------	--------	-------

El usuario puede: - Ingresar el **ID** para abrir el correo. - Ingresar **0** para volver al menú.

5.2 Buscar por remitente

Permite ingresar el nombre del remitente.

Si existe: - Se listan todos los correos enviados por ese remitente. - Se puede abrir un correo ingresando su ID.

Si no existe: - Se muestra el mensaje: «No se encontraron correos de ese remitente».

5.3 Buscar por palabra clave

La búsqueda se hace sobre **asunto y cuerpo del mensaje**, usando coincidencia exacta.

Acciones: - Ingresar la palabra clave. - Visualizar la lista de correos donde aparece. - Abrir un correo escribiendo su ID.

Si no hay resultados: «No se encontraron coincidencias».

5.4 Abrir un correo

La visualización incluye: - ID - Remitente - Asunto - Fecha - Cuerpo del mensaje

El usuario vuelve a la pantalla anterior presionando **ENTER**.

6. Archivo Externo (correos.txt)

Para cargar automáticamente correos, el archivo debe seguir este formato:

remitente;asunto;cuerpo;fecha
remitente2;asunto2;cuerpo2;fecha2
...

Ejemplo:

ana@correo.com;Reunión;Reunión mañana a las 10;2025-11-12
pepe@correo.com;Recordatorio;Enviar informe;2025-11-15

Cada línea genera un correo con un ID único.

7. Cómo Funcionan las Búsquedas Internamente

7.1 Árbol binario de correos

Los correos se insertan en un árbol binario ordenados por fecha. Esto permite generar una lista ordenada eficientemente.

7.2 Mapas hash

- correosPorRemitente → permite obtener correos asociados a un remitente.
- correosPorID → permite abrir cualquier correo de forma inmediata.

7.3 Matriz dispersa (mapa de palabras por ID)

Cada correo se tokeniza (separan palabras) y se guardan en:

`matrizDispersa[ID][palabra] = 1`

Esto permite búsqueda rápida de palabras clave.

8. Manejo de Errores Comunes

1. **No carga el archivo correos.txt**
 - Verifique que exista en la misma carpeta del ejecutable.
2. **La pantalla no se limpia correctamente**
 - El comando `cls` funciona solo en Windows. En Linux usaría `clear`.
3. **Las búsquedas no encuentran coincidencias**
 - La búsqueda por palabra ignora mayúsculas/minúsculas.
 - Solo detecta palabras exactas.

9. Salida del Programa

Seleccione 0 en el menú principal para cerrar la ejecución.

10. Conclusiones

Este programa permite gestionar y consultar correos de forma ordenada, rápida y eficiente gracias al uso de: - Árbol binario para ordenamiento. - Mapas hash para accesos directos. - Matriz dispersa para búsquedas semánticas.

El diseño modular facilita su ampliación y mantenimiento.