

## Tarea Individual 25 - Uso de POP3, lectura de correos

La salida muestra:

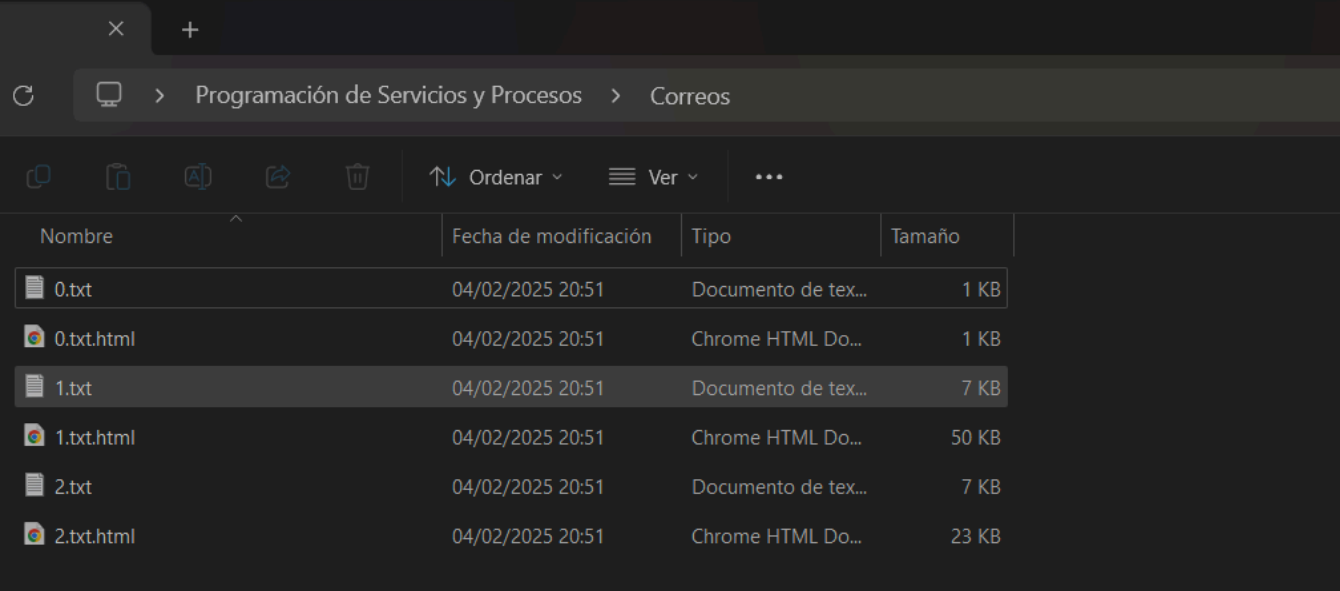
- La conexión al servidor POP3 y la autenticación.
- La lista de mensajes obtenidos del buzón de entrada y el proceso de almacenamiento de los correos en archivos.
- El cierre de sesión y desconexión del servidor.

He limitado la descarga a tres mensajes.

La app de java me devuelve la salida sin ningún error:

```
-----[CONEXION ABIERTA]-----
Numero de mensajes: 251
Numero de mensajes no leidos: 251
MENSAJE #1:
Guardando... La mejor chaqueta impermeable que hemos hecho nunca The North Face <thenorthface@newsletter.thenorthface.com>
Nombre del archivo: C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\0
... C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\0.txt
... C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\0.txt.html
MENSAJE #2:
Guardando... Aún más descuentos en el Mes de las Marcas: SIN IVA en SAMSUNG, 20% Dto. en HP y muchos más Novedades Worten <novedades@worten.es>
Nombre del archivo: C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\1
... C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\1.txt
... C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\1.txt.html
MENSAJE #3:
Guardando... Sin IVA en Samsung ¡¡ULTIMAS HORAS!! Novedades Worten <novedades@worten.es>
Nombre del archivo: C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\2
... C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\2.txt
... C:\Users\ [REDACTED] \Desktop\Programación de Servicios y Procesos\Correos\2.txt.html
-----[SESION CERRADA, SERVIDOR DESCONECTADO]-----
```

Me guarda los archivos:



Nombre	Fecha de modificación	Tipo	Tamaño
0.txt	04/02/2025 20:51	Documento de tex...	1 KB
0.txt.html	04/02/2025 20:51	Chrome HTML Do...	1 KB
1.txt	04/02/2025 20:51	Documento de tex...	7 KB
1.txt.html	04/02/2025 20:51	Chrome HTML Do...	50 KB
2.txt	04/02/2025 20:51	Documento de tex...	7 KB
2.txt.html	04/02/2025 20:51	Chrome HTML Do...	23 KB

Codigo y manejo de los dos errores:

package Tema4.correo;

```

import java.io.*;
import java.util.*;
import javax.mail.*;
import javax.mail.internet.MimeBodyPart;

public class EjemploPOP3 {

    private static final String HOST = "pop.gmail.com";
    private static final String USERNAME = "*****@gmail.com";
    private static final String PASSWORD = "**** *";
    //Aqui personalizo una carpeta destino
    private static final String CARPETA_DESTINO = "C:\\Users\\*****\\Desktop\\Programación de
Servicios y Procesos\\Correos";

    public static void main(String[] args) throws MessagingException, IOException{
        //ReceiveMailPOP3.dois();
        Folder carpeta = null;
        Store store = null;

        try {
            Properties props = new Properties();
            //Google usa POP3S (SEGURO)
            props.put("mail.store.protocol", "pop3s");
            //Uso de TLS 1.2
            props.put("mail.pop3s.ssl.protocols", "TLSv1.2 TLSv1.3");
            props.put("mail.pop3s.ssl.enable", "true");

            //Clase que representa una sesion de correo
            //Encapsula la configuracion y propiedades necesarias para conectarse a un
servidor de correo
            Session session = Session.getDefaultInstance(props);
            //La clase store proporciona metodos para conectarse, desconectarse y acceder
a buzones de un servidor de correo
            store = session.getStore();
            store.connect(HOST, USERNAME, PASSWORD);
            System.out.println("-----[CONEXION
ABIERTA]-----");

            //Obtenemos todo el contenido del buzón de entrada
            carpeta = store.getDefaultFolder().getFolder("INBOX");
            carpeta.open(Folder.READ_ONLY);
            //Obtenemos los mensajes del buzón de entrada
            Message[] mensajes = carpeta.getMessages();
            System.out.println("Numero de mensajes: " + carpeta.getMessageCount());
            System.out.println("Numero de mensajes no leidos: " +
carpeta.getUnreadMessageCount());

```

```

//Recorremos todos los mensajes
for (int i = 0; i < 3; ++i) { //for (int i = 0; i < mensajes.length; ++i) {

    System.out.println("MENSAJE #" + (i + 1) + ":");
    Message mensaje = mensajes[i];

    String remitente = "Desconocido";
    //Si el mensaje es de respuesta cogemos el primero para nuestro
    remitente

    if(mensaje.getReplyTo().length >= 1) {
        remitente = mensaje.getFrom()[0].toString();
    }
    //Si no, cogemos el primer remitente de la lista de remitentes
    else if(mensaje.getFrom().length >= 1) {
        remitente = mensaje.getFrom()[0].toString();
    }

    //Obtenemos el asunto del mensaje
    String asunto = mensaje.getSubject();
    System.out.println("Guardando... " + asunto + " " + remitente);
    //Eliminamos caracteres
    especiales-----

    String nombreFichero = CARPETA_DESTINO + "\\" + i;
    System.out.println("Nombre del archivo: " + nombreFichero);
    //Guardamos el archivo en nuestra carpeta
    guardaEnFichero(mensaje.getContent(), nombreFichero);

    }
    }finally {
        //Cerramos los objetos
        if(carpeta != null) {
            carpeta.close(true); }

        if(store != null) {store.close(); }
        System.out.println("-----[SESION CERRADA, SERVIDOR
DESCONECTADO]-----");
    }

}

    public static void guardaEnFichero(Object contenido, String nombreFichero) throws
IOException, MessagingException {

    OutputStream flujoSalida = null;
    InputStream flujoEntrada = null;

    try {

```

```

if(contenido instanceof Multipart) {
    //Hacemos un cast para obtener las diferentes partes del email
    Multipart multi = ((Multipart)contenido);
    int parts = multi.getCount();

    //Recorremos todas las partes del email
    for(int i=0; i < parts; ++i) {

        //Sacamos una parte del email
        MimeBodyPart parte = (MimeBodyPart)multi.getBodyPart(i);
        //Si esta parte tiene mas partes hacemos una recursion
        if(parte.getContent() instanceof Multipart) {
            guardaEnFichero(parte.getContent(), nombreFichero);

        }else {
            // Si no tiene mas partes
            String extension = "";
            //Fijamos la extension del fichero segun el tipo de la parte
            if(parte.isMimeType("text/html")) {
                extension = "html";
            }else if(parte.isMimeType("text/plain")) {
                extension="txt";
            }else { //Si no sabemos el tipo intentamos obtener la
                extension del nombre

                if (parte.getDataHandler() != null) {
                    extension = parte.getDataHandler().getName();
                }

            }
            //Añadimos la extension al fichero
            nombreFichero = nombreFichero + "." +extension;
            System.out.println("... " + nombreFichero);
            flujoSalida = new FileOutputStream(new
            File(nombreFichero));

            // Sacamos todo el contenido de la parte del mensaje
            flujoEntrada = parte.getInputStream();
            // Leemos el contenido de la parte byte a byte y lo
            escribimos en el fichero

            int j;
            while ((j = flujoEntrada.read()) != -1) {
                flujoSalida.write(j);
            }
        } //else
    } //for
} // if

```

```
}finally { //cerramos los objetos

    if(flujoEntrada!=null) {flujoEntrada.close(); }
    if(flujoSalida !=null) {flujoSalida.close(); }

}

}

}
```