

Uso del protocolo SMTP para enviar un email

En este apartado pondrás en práctica el uso del protocolo SMTP desde un programa Java, enviando un correo desde una cuenta de Gmail.



Utilizaremos el API *JavaMail*, que forma parte del estándar Java EE pero que se integra como un paquete aparte.

<https://es.wikipedia.org/wiki/JavaMail>

WIKIPEDIA

La enciclopedia libre

Buscar en Wikipedia

Buscar

Donaciones

JavaMail

9 idiomas

Contenidos

ocultar

Inicio

Características

Referencias

Enlaces externos

Artículo

Discusión

Leer

Editar

Ver historial

Herramientas

JavaMail es una **API Java** que facilita el envío y recepción de correo electrónico desde código java a través de protocolos **SMTP**, **POP3** y **IMAP**. JavaMail está integrado en la plataforma **Java EE**, pero también proporciona un paquete opcional para su uso en Java SE.¹

La última versión liberada bajo la identificación de Java EE es la 1.6.2, publicada en agosto de 2018. Existe otra implementación JavaMail de **código abierto** - GNU JavaMail - aunque sólo soporta la versión 1.3 de la especificación JavaMail, además solo proporciona un único backend gratuito de **NNTP**, que permite utilizar esta tecnología para leer y enviar artículos de **grupos de noticias**.

A partir del 14 de septiembre de 2018 el proyecto de JavaMail fue movido de Oracle a **Eclipse Foundation** como parte de **EE4J project**², en el cual la última versión estable es la versión 1.6.3, liberada el 26 de noviembre de 2018.

JavaMail

Información general

Tipo de programa

biblioteca de software

Autor

Oracle Corporation

Desarrollador

Eclipse Project

Lanzamiento inicial

21 de marzo de 2002

Licencia

(EPL) V2.0 y (GPL) v2 with Classpath Exception

Información técnica

Programado en

Java

Plataformas admitidas

multiplataforma

Versiones

Última versión estable

1.6.3 (info) (26 de noviembre de 2018)

← → ↺

https://www.oracle.com/java/technologies/javamail-releases.html

ORACLE

Products Industries Resources Customers Partners Developers C

Java / Technologies /
JavaMail API

Downloads

April 28, 2017

Starting with JavaMail 1.5.0, the latest JavaMail releases are available from the [JavaMail project page on GitHub](#)

JavaMail Documentation

The JavaMail documentation can be accessed online [here](#).

- [PDF format](#) (225763 bytes)

Previous Releases

Recent releases can be found in the [java.net Maven repository](#).

Very old releases are listed below:


JavaMail 1.4.5
JavaMail API 1.4.5, you can find it [here](#)

JavaMail 1.4.4
JavaMail API 1.4.4, you can find it [here](#)


JavaMail 1.4.3
JavaMail API 1.4.3, you can find it [here](#)

JavaMail 1.4.2


<https://www.oracle.com/java/technologies/javamail-releases.html>

Product / File Description	File Size	Download
JavaMail API 1.4.5	2.2 MB	 javamail1_4_5.zip

JavaMail API 1.4.4

Product / File Description	File Size	Download
JavaMail API 1.4.4	2.2 MB	 javamail1_4_4.zip

JavaMail API 1.4.3

Product / File Description	File Size	Download
JavaMail 1.4.3	2.73 MB	 javamail-1.4.3.zip

You must accept the [Oracle License Agreement](#) to download this software.

☐ I reviewed and accept the Oracle License Agreement
Required

You will be redirected to the login screen in order to download the file.

Download javamail1_4_5.zip 

Sigue los pasos para crear tu programa Java

1

Crea el proyecto Java.

Crea un nuevo proyecto Eclipse de tipo *Java Project* con el nombre *Proyecto JavaMail*.

2

Importa las librerías de *JavaMail*.

Importa dentro del proyecto los archivos JAR del API de *JavaMail* que anteriormente has descargado y descomprimido. Recuerda que para importar librerías externas debes:

Agregar manualmente

1.Descarga:

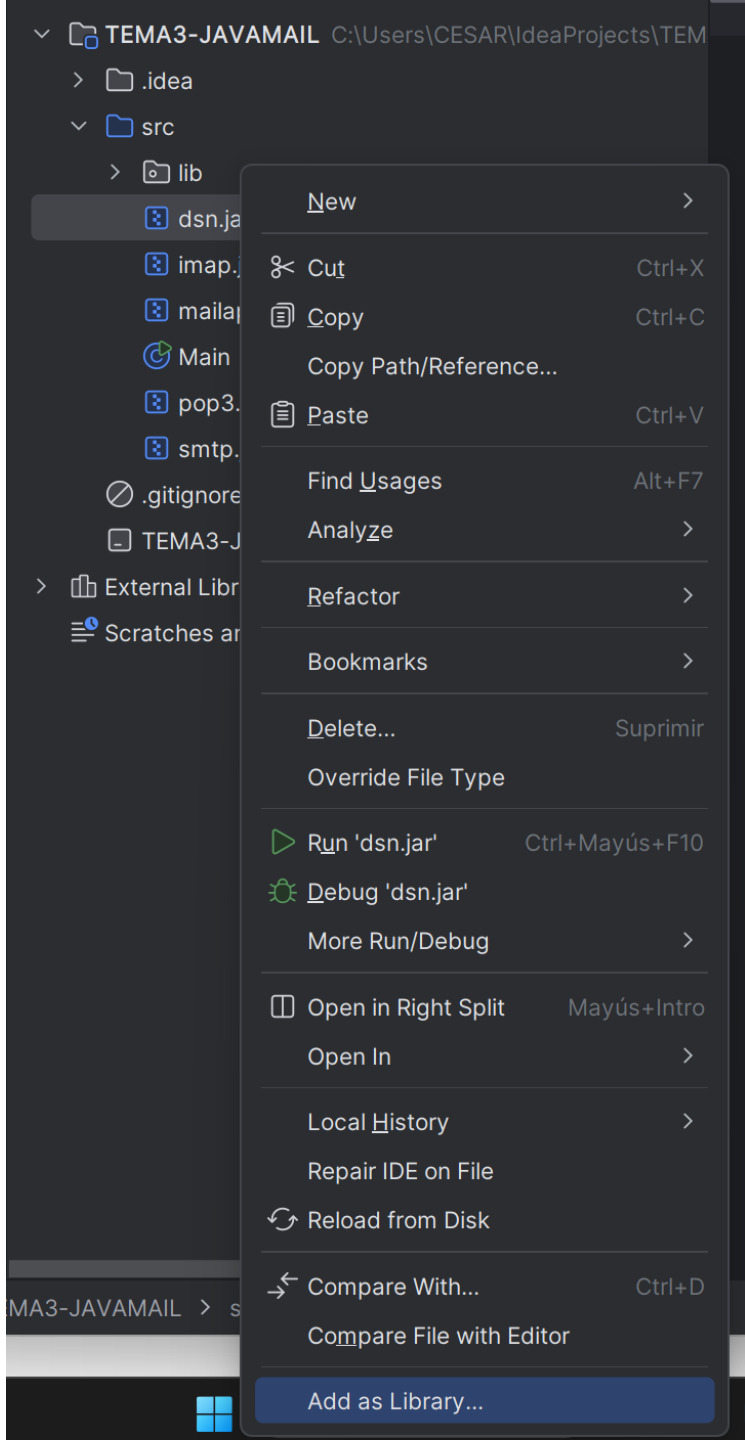
- Ve al repositorio de JavaMail y descarga el archivo ZIP que contiene todas las librerías .jar.

2.Agrega a tu proyecto:

- Copia el archivo .jar a un directorio dentro de tu proyecto (por ejemplo, libs).
- En IntelliJ, haz clic derecho en el archivo .jar > Add as Library.

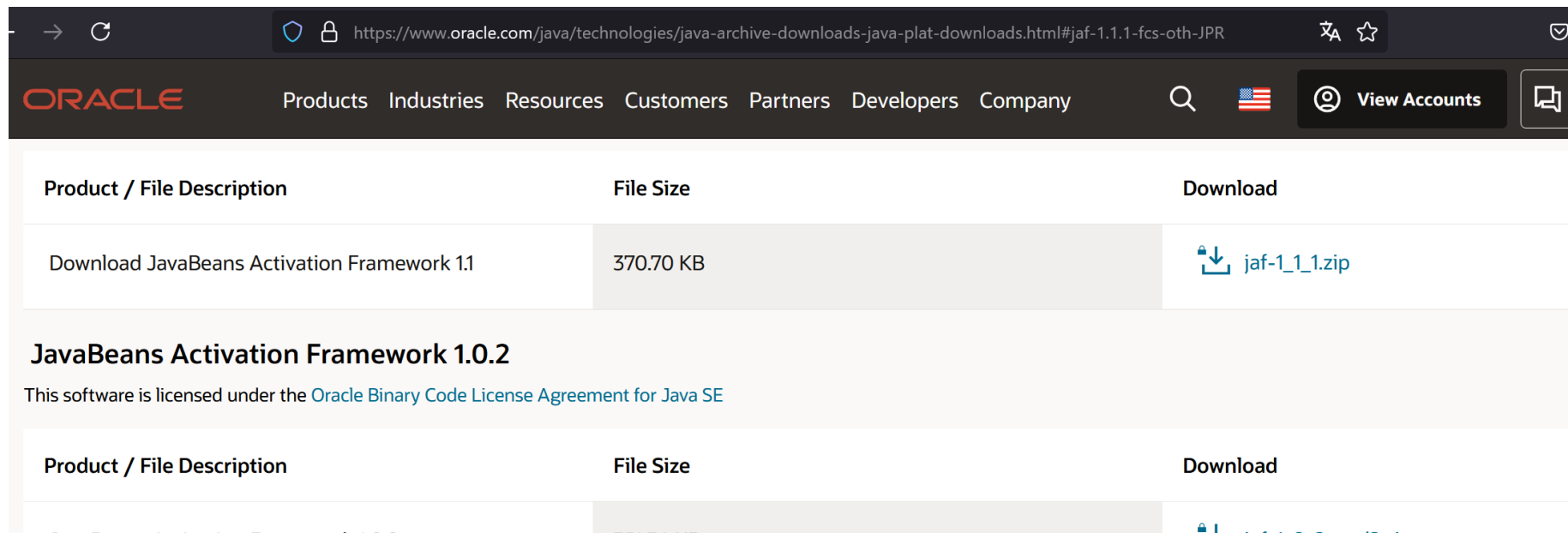
3.Configura el classpath:

- Asegúrate de que IntelliJ incluya el archivo en el classpath del proyecto.



La versión 1.4.5 de JavaMail es bastante antigua y puede no ser compatible con las versiones modernas de Java (11 en adelante), donde el paquete javax.activation ya no está incluido en el JDK. Si decides usarla, debes agregar manualmente las dependencias faltantes.

La versión 1.4.5 de JavaMail requiere el JavaBeans Activation Framework (JAF) para manejar clases como javax.activation.DataSource. Descarga el archivo .jar de JAF desde:



The screenshot shows the Oracle Java Archive Downloads page. The browser address bar displays the URL: <https://www.oracle.com/java/technologies/java-archive-downloads-java-plat-downloads.html#jaf-1.1.1-fcs-oth-JPR>. The Oracle logo is visible in the top left, and navigation links for Products, Industries, Resources, Customers, Partners, Developers, and Company are in the top center. A search icon, a US flag, and a 'View Accounts' button are on the right. Below the navigation bar, there is a table with three columns: 'Product / File Description', 'File Size', and 'Download'. The first row shows 'Download JavaBeans Activation Framework 1.1' with a file size of '370.70 KB' and a download link for 'jaf-1_1_1.zip'. Below this table, the section 'JavaBeans Activation Framework 1.0.2' is visible, followed by a note: 'This software is licensed under the [Oracle Binary Code License Agreement for Java SE](#)'. Another table with the same columns is partially visible at the bottom.

Product / File Description	File Size	Download
Download JavaBeans Activation Framework 1.1	370.70 KB	jaf-1_1_1.zip

JavaBeans Activation Framework 1.0.2

This software is licensed under the [Oracle Binary Code License Agreement for Java SE](#)

Product / File Description	File Size	Download
Download JavaBeans Activation Framework 1.0.2	354.74 KB	jaf-1_0_2.zip

El error Could not convert socket to TLS indica un problema al establecer la conexión segura con el servidor SMTP. Esto suele deberse a una incompatibilidad entre las versiones de protocolo TLS soportadas por Java y el servidor de correo (en este caso, Gmail).

Pasos para solucionar el error

Actualizar Java:

Asegúrate de estar usando una versión actualizada de Java, preferiblemente la última versión de Java LTS.

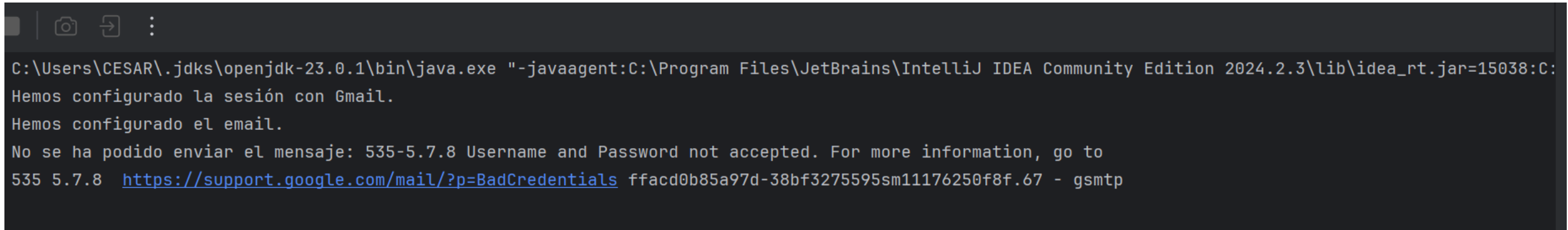
Habilitar protocolos TLS modernos: Algunas versiones de Java deshabilitan por defecto ciertos protocolos. Puedes habilitar TLS 1.2 o 1.3 de la siguiente manera:

Agrega esta línea antes de establecer la conexión:

```
propiedades.put("mail.smtp.ssl.protocols", "TLSv1.2");
```

Configuración TLS moderna:

- Se añadió `mail.smtp.ssl.protocols` para forzar el uso de TLS 1.2.
- Propiedad `mail.smtp.ssl.trust`:
- Evita errores con certificados SSL no confiables.
- Contraseña de aplicación:
- Gmail requiere contraseñas de aplicación para conexiones SMTP seguras. Genera una desde tu cuenta de Google.
- Uso de `Session.getInstance`:
- Más seguro que `Session.getDefaultInstance` para manejar propiedades únicas por sesión.
- Mensajes claros en la salida:
- Mejora la depuración en cada etapa del envío de correo.



```
C:\Users\CESAR\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=15038:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\bin" -Dfile.encoding=UTF-8
Hemos configurado la sesión con Gmail.
Hemos configurado el email.
No se ha podido enviar el mensaje: 535-5.7.8 Username and Password not accepted. For more information, go to
535 5.7.8 https://support.google.com/mail/?p=BadCredentials ffacd0b85a97d-38bf3275595sm11176250f8f.67 - gsmtip
```

¿Qué es una contraseña de aplicación?

Es una contraseña generada por Gmail que puedes usar en lugar de tu contraseña habitual para permitir que aplicaciones externas (como tu programa Java) se conecten de forma segura a tu cuenta.

Pasos para generar una contraseña de aplicación:

Habilita la autenticación en dos pasos:

Ve a tu cuenta de Google.

En el menú de navegación, selecciona Seguridad.

Busca la sección Inicio de sesión en Google y activa la verificación en dos pasos. Sigue las instrucciones para configurarla (puedes usar tu teléfono para recibir códigos).

Accede a las contraseñas de aplicación:

Una vez activada la verificación en dos pasos, regresa a la sección Seguridad.

En la misma sección de Inicio de sesión en Google, busca la opción Contraseñas de aplicación.

Haz clic en Contraseñas de aplicación (puede que te pida iniciar sesión de nuevo).

Crea una nueva contraseña de aplicación:

En el menú desplegable, selecciona:

Aplicación: Selecciona Correo.

Dispositivo: Selecciona Otro (nombre personalizado) y escribe algo como JavaMail.

Haz clic en Generar.

Obtén la contraseña generada:

Gmail mostrará una contraseña de 16 caracteres (por ejemplo: abcd efgh ijkl mnop).

Copia esta contraseña y úsala en lugar de tu contraseña normal en el código.



Inicio



Información personal



Datos y privacidad



Seguridad



Contactos y compartir



Pagos y suscripciones



Información general

Actividad relacionada con la seguridad reciente

Ha habido un intento sospechoso de inicio de sesión con tu contraseña

8 ene · España



[Revisar actividad de seguridad](#)

Cómo inicias sesión en Google

Asegúrate de poder acceder siempre a tu cuenta de Google manteniendo al día esta información



Verificación en dos pasos



Activa desde: 14 dic 2021



Llaves de acceso y llaves de seguridad

3 llaves de acceso



Contraseña

Última modificación: 26 jun 2024



Saltar contraseña cuando sea posible



Activada



Notificación de Google

2 dispositivos



Clase MimeMessage en JavaMail

- La clase MimeMessage es una implementación de un mensaje de correo electrónico basado en el formato MIME (Multipurpose Internet Mail Extensions). Se utiliza para construir, leer y enviar correos electrónicos que admiten texto, HTML, adjuntos y contenido multimedia.

Principales características de MimeMessage

- Encapsulación de datos MIME:
- Permite construir correos electrónicos con varios tipos de contenido (texto, HTML, imágenes, etc.).
- Compatibilidad con cabeceras MIME estándar:
- Soporta cabeceras como To, From, Subject, Content-Type, entre otras.
- Flexibilidad:
- Se puede usar para correos simples (texto plano) o complejos (con HTML, adjuntos y contenido incrustado).

Métodos principales de MimeMessage

1. Configuración de cabeceras

•setFrom(Address address)

Establece el remitente del mensaje.

•setFrom()

Establece el remitente predeterminado de la sesión.

•addFrom(Address[] addresses)

Agrega una lista de remitentes.

•setRecipients(Message.RecipientType type, String addresses)

Establece los destinatarios del mensaje (TO, CC, BCC) a partir de una cadena de direcciones.

•addRecipients(Message.RecipientType type, String addresses)

Agrega más destinatarios al mensaje.

•setRecipients(Message.RecipientType type, Address[] addresses)

Establece los destinatarios a partir de un arreglo de direcciones.

•addRecipients(Message.RecipientType type, Address[] addresses)

Agrega más destinatarios a partir de un arreglo de direcciones.

•setSubject(String subject)

Establece el asunto del mensaje.

•setSubject(String subject, String charset)

Establece el asunto con un conjunto de caracteres específico.

•setSentDate(Date date)

Establece la fecha de envío del mensaje.

•addHeader(String name, String value)

Agrega una cabecera personalizada al mensaje.

•setHeader(String name, String value)

Establece o reemplaza el valor de una cabecera.

•removeHeader(String name)

Elimina una cabecera específica.

•getHeader(String name)

Obtiene el valor de una cabecera.

Configuración del contenido

`setText(String text)`

Establece el cuerpo del mensaje en texto plano.

`setText(String text, String charset)`

Establece el cuerpo del mensaje en texto plano con un conjunto de caracteres específico.

`setText(String text, String charset, String subtype)`

Establece el cuerpo del mensaje con un tipo MIME específico (e.g., text/html).

`setContent(Object content, String mimeType)`

Establece el contenido del mensaje con un tipo MIME específico.

`setContent(Multipart multipart)`

Establece el contenido del mensaje como una estructura MIME multipart.

`setDisposition(String disposition)`

Establece la disposición del contenido (e.g., inline, attachment).

Manipulación de mensajes existentes

`saveChanges()`

Guarda los cambios realizados en el mensaje.

`writeTo(OutputStream os)`

Escribe el contenido completo del mensaje en un flujo de salida.

`writeTo(OutputStream os, String[] ignoreList)`

Escribe el contenido del mensaje excluyendo las cabeceras especificadas.

`updateHeaders()`

Actualiza automáticamente las cabeceras basadas en el contenido del mensaje.

`getAllRecipients()`

Obtiene todos los destinatarios (TO, CC, BCC) del mensaje.

`getContent()`

Obtiene el contenido del mensaje (puede ser texto, HTML o una estructura MIME multipart).

`getContentType()`

Obtiene el tipo MIME del mensaje.

La línea `Transport t = sesion.getTransport("smtp");` en JavaMail se utiliza para obtener una instancia del objeto `Transport`, que es el responsable de enviar mensajes de correo electrónico a través de un protocolo específico, en este caso, SMTP (Simple Mail Transfer Protocol).

Parámetros

`protocol`: Especifica el protocolo que se utilizará para enviar el correo. En este caso, "smtp" se usa para el envío de correos electrónicos.

Valor retornado

Retorna un objeto de tipo `Transport` que permite enviar mensajes utilizando el protocolo especificado.

Excepción

`NoSuchProviderException`: Se lanza si el protocolo especificado no está configurado o no es compatible con la sesión.

Flujo típico usando Transport

1. Obtener la instancia de Transport

La instancia se obtiene a través de la sesión configurada:

```
Transport t = sesion.getTransport("smtp");
```

2. Conectarse al servidor SMTP

Usa el método connect para autenticarte en el servidor SMTP. Ejemplo:

```
t.connect("smtp.gmail.com", "tuCorreo@gmail.com", "tuContraseña");
```

Primer parámetro (host): El servidor SMTP al que te estás conectando (e.g., "smtp.gmail.com").

Segundo parámetro (user): Nombre de usuario para autenticar.

Tercer parámetro (password): Contraseña (o contraseña de aplicación, si aplica).

3. Enviar el mensaje

Una vez conectado, usa el método sendMessage para enviar el mensaje:

```
t.sendMessage(email, email.getAllRecipients());
```

Primer parámetro (message): El objeto MimeMessage que contiene el correo configurado.

Segundo parámetro (addresses): Los destinatarios del mensaje.

4. Cerrar la conexión

Cierra la conexión con el servidor SMTP:

```
t.close();
```