



PSP - U5 Actividades Mail

[Descargar estos apuntes](#)

- Para cada actividad, crea un proyecto nuevo con el nombre de la clase principal.
- Crea una/varias nuevas clases dentro del paquete `psp.u5.actividades` para cada ejercicio.
- Las clases deben tener el prefijo `psp.u5.actividades.U5AXX_XXXXXXX` y darles un nombre identificativo de la funcionalidad que tienen en el diseño (por ejemplo `psp.actividades.U5A20_FTPClient`).

Es importante que las salidas, tanto de los programas cliente, como del servidor, indiquen claramente de donde viene la información que están mostrando.

1. Actividad TestMail

Creación del proyecto

Este proyecto lo tienes que crear como **Java with Maven > Java Application**

1. Crea una clase TestMail de utilidad de correo

Esta clase va a tener como propiedades todos los objetos necesarios para la composición de un email, y va a tener como métodos las distintas formas de enviar un correo.

Se puede ampliar con todos los métodos que necesites y/o creas convenientes.

Os pongo un ejemplo de la especificación mínima debería tener la clase, añadiendo los constructores y métodos que necesites para tu implementación.

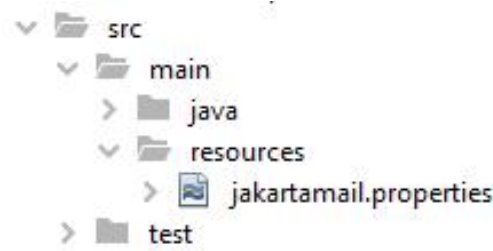
C TestFTP
<div>□ FTPClient ftpClient</div> <div>□ propiedades_que_necesites</div>
<div>○ void</div>
<div>● TestFTP(String propertiesFile)</div> <div>● void connect()</div> <div>● void disconnect()</div> <div>● void uploadFile(String localFilePath, String remoteFilePath)</div> <div>● void downloadFile(String remoteFilePath, String localFilePath)</div> <div>● void uploadBinaryFile(String localFilePath, String remoteFilePath)</div> <div>● void downloadBinaryFile(String remoteFilePath, String localFilePath)</div> <div>● void setRemoteWorkingDirectory(String remoteDirPath)</div> <div>● void createDirectory(String remoteDirPath)</div> <div>● void deleteFile(String remoteFilePath)</div> <div>● métodos_privados_que_necesites()</div>

Esta es la estructura básica de la clase TestMail. Además de los métodos indicados, puedes añadir otros que consideres necesarios para completar la funcionalidad y facilitar el uso de la clase. También puedes cambiar los tipos de datos de los parámetros según tus necesidades (por ejemplo, usar listas para múltiples destinatarios) y el tipo de retorno de los métodos (por ejemplo, boolean para indicar éxito o fracaso).

Las propiedades con la configuración del servidor SMTP, deberán leerse de un archivo `jakartamail.properties`, con el contenido de las variables que se muestra a continuación.

```
# jakartamail.properties
# Data to send emails from a GMAIL account
mail.from=cuenta@iesdoctorbalmis.com
mail.fromtext = "Actividades PSP"
mail.username=cuenta@iesdoctorbalmis.com
mail.password=XXXXX
mail.smtp.host=mail-eu.smtp2go.com
mail.smtp.port=2525
mail.smtp.auth=true
mail.smtp.starttls.enable=true
#mail.debug=true
```

El archivo lo vamos a guardar en la carpeta **src/main/resources** del proyecto, tal y como hemos visto en los apuntes.



El acceso al archivo lo haremos de la siguiente forma:

```
InputStream input = this.getClass().getResourceAsStream("/" + propertiesFile)
```

En los apuntes tienes varios recursos para ver cómo trabajar con este tipo de archivos, que no deja de ser una subclase de `HashMap`.

Credenciales para el envío de correo

Recuerda usar las credenciales y la cuenta de correo que hayas configurado en SMTP2GO. Si lo has hecho bien, no tienen que coincidir con las credenciales de tu cuenta de correo.

Las propiedades `mail.from` y `mail.fromtext` nos permiten modificar lo que se muestra en la cabecera del correo enviado, creando el objeto `InternetAddress` para el método **`Message.setFrom`** con dos parámetros.

2. Crea la clase principal `U5A01_TestMail`

Crea una clase con un `main` que instancie un objeto de tipo `TestMail` y envíe mensajes de diferentes tipos utilizando los métodos de la clase `TestMail`.

Ofrece un menú, en el `main`, que permita probar las diferentes opciones de la clase `TestMail`.

Para las rutas de los archivos, se pueden indicar tanto rutas relativas al proyecto como rutas absolutas del sistema.

Funcionalidad de la clase `U5A01_TestMail`

Ten en cuenta que esta clase debe permitir enviar un correo electrónico de forma que la librería `JakartaMail` sea transparente para el desarrollador que la use.

Para ello, puedes añadir cuantas propiedades y métodos consideres necesarios a la clase `TestMail`.

El código debe consistir básicamente en:

- Seleccionar el tipo de correo a enviar (simple, HTML, con adjuntos, con imágenes embebidas).
- Pedir los datos necesarios para cada tipo de correo (destinatario, asunto, cuerpo del mensaje, ruta del archivo adjunto, etc.).
- Llamar al método correspondiente de la clase `TestMail` para enviar el correo.
- Mostrar mensajes de éxito o error según corresponda.