



Envío de Correos Electrónicos con la API de JavaMail

Pilar García Lugo - pilargarcialugo@gmail.com



Enunciado

Se requiere el diseño y desarrollo de una aplicación que permita hacer envío de correos electrónicos por medio de una interfaz de usuario y que permita la navegación en la interfaz de manera que el usuario tenga la posibilidad de ver los correos que ya ha enviado y, de la misma forma, volver a la ventana de inicio en donde pueda redactar un nuevo correo. el usuario también debe poder limpiar los campos del correo a enviar.

Requerimientos Funcionales

- Enviar Correo Electrónico
- Navegar al anterior correo
- Navegar al siguiente correo
- Limpiar pantalla del mensaje

Revisar el documento [RequerimientosJavaMail.PDF](#) para ver la descripción detallada de los requerimientos

Requerimientos No Funcionales

- Apariencia de la Aplicación

The image shows a wireframe of a JavaMail application window. The window has a title bar with the text "Correo Electrónico con JavaMail" and standard window controls (minimize, maximize, close). The main content area is divided into two sections. The first section, titled "Iniciar Sesión", contains two input fields labeled "Correo" and "Contraseña". The second section, titled "Correo", contains two input fields labeled "Destinatario" and "Asunto", followed by a large text area labeled "Mensaje". At the bottom of the window, there are three buttons: a left arrow button, a button labeled "Enviar Mensaje", and a right arrow button.

Figura: Ventana Inicial del Mensaje. Fuente: Elaboración propia realizada en Balsamiq Mockups

Requerimientos No Funcionales (Cont.)

- Apariencia de la Aplicación

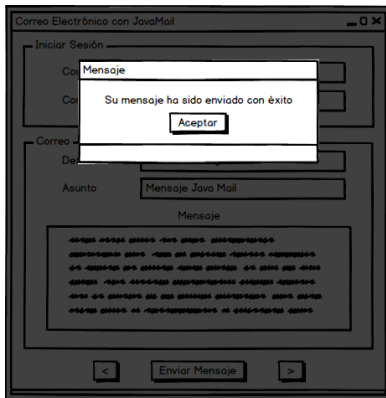


Figura: Ventana de Confirmación del envío del mensaje. Fuente: Elaboración propia realizada en Balsamiq Mockups

Requerimientos No Funcionales (Cont.)

- Apariencia de la Aplicación

Correo Electrónico con JavaMail

Iniciar Sesión

Correo: correo@gmail.com

Contraseña: *****

Correo

Destinatario: destinatario@gmail.com

Asunto: Mensaje Java Mail

Mensaje Enviado

< Reenviar Mensaje >

Figura: Ventana de un mensaje enviado. Fuente: Elaboración propia realizada en Balsamiq Mockups

Diagrama de Clases

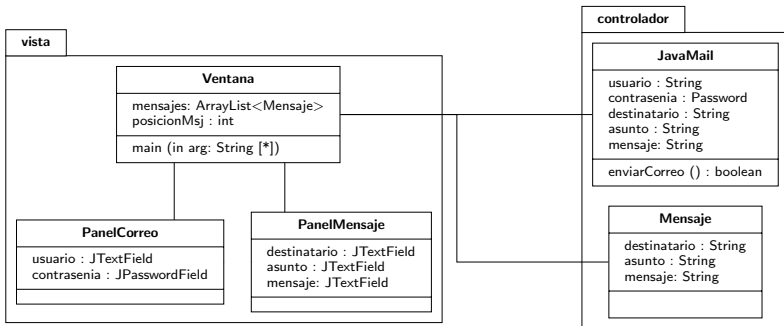


Figura: Diagrama de Clases para el envío de correos electrónicos. Fuente: Elaboración propia.

Agenda

- 1 ¿Qué es JavaMail?
- 2 Requerimientos de Configuración
- 3 Requerimientos de Configuración (Cont.)
- 4 Clases y configuración de JavaMail
 - Clases de JavaMail
 - Creación y configuración de la clase Session
 - Properties Específicas para Session
 - Construcción del Mensaje a enviar
 - Envío del Mensaje
- 5 Referencias

¿Qué es JavaMail?

- La API de JavaMail proporciona un marco independiente de la plataforma y del protocolo para crear aplicaciones de correo y mensajería.
- Está disponible como un paquete opcional para usar con la plataforma Java SE (Standard Edition) y también está incluida en la plataforma Java EE (Enterprise Edition).
- **Enlace de descarga de la librería Twitter4J:** <https://goo.gl/xRDfr5>



Requerimientos de Configuración



Requerimientos

- Usuario y contraseña.
- Protocolo SMTP.
- Puerto 587, (el puerto por defecto de SMTP es el 25)^a.
- TLS si está disponible.

^aAlgunas IP deniegan el uso del puerto 25 ya que es afectado por gran cantidad de spam y tráfico basura: el puerto se bloquea.

Requerimientos de Configuración

SMTP

- SMTP (Protocolo simple de transferencia de correo).
- Protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

TLS

- TLS (Transport Layer Security)
- Permite confiar información personal a sitios web, ya que los datos se ocultan a través de métodos criptográficos mientras se navega en sitios seguros.
- Está basado en SSL (Secure Sockets Layer) Capa de Conexiones Seguras: protocolo que hace uso de certificados digitales para establecer comunicaciones seguras a través de Internet.

Clases de JavaMail

Clases de JavaMail implicadas para el envío de mensajes:

- **Session:** Representa la conexión con el servidor Gmail de correo. Hay que obtenerla pasándole los parámetros de configuración para el servidor de correo.
- **MimeMessage:** El mensaje^a.
- **Transport:** Clase para el envío de mensajes. Se utiliza el método `send()` para el envío.

^aMIME viene de Multipurpose Internet Mail Extensions. Define el contenido de lo que se está transfiriendo: el formato de los mensajes.



Creación y configuración de la clase Session

La clase Session representa la conexión con el servidor de correo (Gmail). Se obtiene con el método **Session.getInstance()**. Dicho método recibe un parámetro **Properties** con las propiedades de la conexión.

Properties:

- Clase de Java estándar (java.util.Properties).
- Admite guardar parejas “clave-valor” (datos de tipo String, dándoles un nombre a cada uno para luego poder recuperarlos). Esto se hace por medio del método setProperty().

Creación y configuración de la clase Session (Cont.)

```
Properties p = new Properties();  
p.setProperty("un_nombre", "un_valor");  
p.setProperty("PI", "3.1416");
```



Properties Específicas para Session

```
Properties props = new Properties();

// Si requiere o no usuario y password para conectarse.
props.setProperty("mail.smtp.auth", "true");

// Nombre del host de correo, es smtp.gmail.com
props.setProperty("mail.smtp.host", "smtp.gmail.com");

// Puerto de gmail para envio de correos
props.setProperty("mail.smtp.port", "587");

// TLS si esta disponible
props.setProperty("mail.smtp.starttls.enable", "true");
```

Properties Específicas para Session (Cont.)



```
// Instancia de Session
Session session = Session.getInstance(props ,
new javax.mail.Authenticator() {
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(usuario , contrasenia);
    }
});
```

Construcción del Mensaje a enviar

Instanciar la clase **MimeMessage**. En el constructor se pasa el objeto **Session** obtenido anteriormente.

```
Message message = new MimeMessage( session );
```

Definir el Remitente y el Destinatario (FROM y TO) del mensaje.

- Se hace con los métodos `setFrom()` y `setRecipients()` de **MimeMessage**.

Construcción del Mensaje a enviar (Cont.)

- **Destinatarios - Método setRecipients() de MimeMessage:** Este método admite dos parámetros (el tipo de destinatario y el InternetAddress: El constructor de InternetAddress corresponde a la dirección de correo.).

El tipo de destinatario puede ser:

- **Message.RecipientType.TO:** Destinatario principal del mensaje.
 - **Message.RecipientType.CC:** Destinatario al que se envía copia del mensaje.
 - **Message.RecipientType.BCC:** Destinatario al que se envía copia, pero sin que los demás destinatarios puedan verlo.
-
- **Definir el Asunto y Cuerpo del Mensaje:** Se define el asunto (subject) del mensaje y el cuerpo con los métodos de la clase MimeMessage setSubject(asunto) y setText(mensaje).

Envío del Mensaje

- Para enviar se utiliza el método **send()** de la clase **Transport** que recibe como parámetro el mensaje.



Referencias



I. Oracle America, *JavaMail API Design Specification*, 1st ed.,
I. Oracle America, Ed. 500 Oracle Parkway Redwood City, California
94065, U.S.A.: Oracle America, Inc, 2013. [Online]. Available:
<https://goo.gl/zStXPY>



Ejemplos java y c/linux. [Online]. Available: <https://goo.gl/khEva1>



Javamail. Oracle. [Online]. Available: <https://goo.gl/4R85h3>



Envío de Correos Electrónicos con la API de JavaMail

Pilar García Lugo - pilargarcialugo@gmail.com

