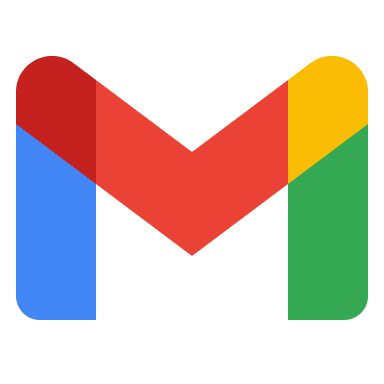
**Correo Electrónico con Maven.**

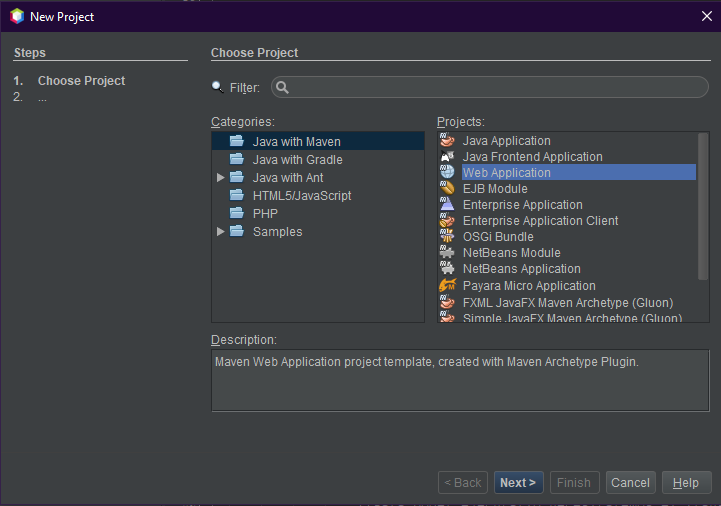
****

**ÍNDICE.**

[**Crear e instalar.**](#_3dln9jry95ou) **3**

[**Estructura básica.**](#_js35fc6c3500) **4**

# Crear e instalar.

****

Para empezar tendremos que crear un proyecto Maven que sería un Web Application y pondremos el nombre que queramos.

Ahora para poder generar un correo electrónico tendremos que importar la dependencia que corresponde al generador, la versión que he utilizado es la 1.6.0 que es una de las versiones más recientes. Para ello necesitaremos poner la dependencia en el pom.xml y añadiremos la siguiente línea de comando.

**<dependency>**

**<groupId>com.sun.mail</groupId>**

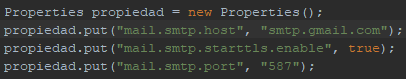
**<artifactId>javax.mail</artifactId>**

**<version>1.6.0</version>**

**</dependency>**

# Estructura básica.

Primero tendremos que crear y establecer las propiedades del correo que vamos a generar, en este caso voy a usar el gmail que es el más cómodo y el que menos problemas da a la hora del uso.



La primera propiedad es el host que quiero acceder que sería el de gmail.

El siguiente es la extensión a los protocolos de comunicación de texto plano (TLS), que es la seguridad de la capa de transporte.

El último es el puerto de entrada al que vamos a acceder.

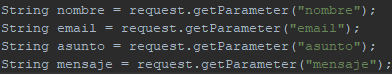
Ahora crearemos la sesión que con las propiedades que hemos creado nos permitirá conectarnos a una cuenta de gmail.



Establecemos el correo desde el que estamos enviando el email (remitente), tiene que ser un correo y una contraseña verdadero.



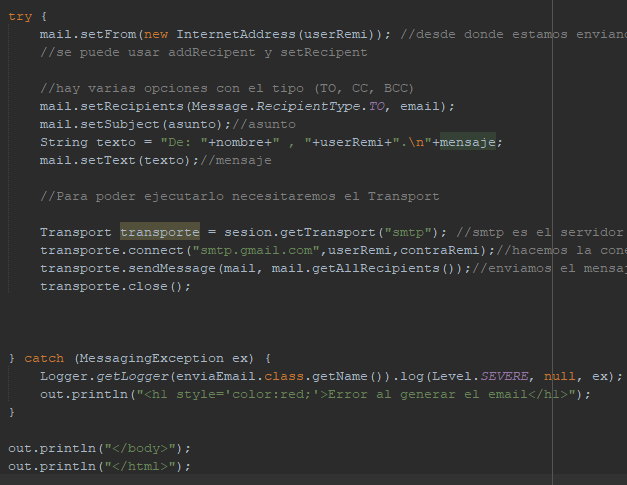
Obtenemos los parámetros recibidos del formulario.



Y creamos el objeto de tipo MimeMessage que sería el mensaje que vamos a enviar con las propiedades ya establecidas de la sesión.



Establecemos todas las propiedades que iré explicando por partes.



setFrom: establecemos desde donde se va a enviar el correo.

setRecipents / addRecipent: podemos usar ambos y dentro de él establecemos a quien vamos a enviarle el correo y de qué forma con el *RecipientType*

-TO: es el envío hacia un único destinatario

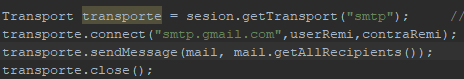
-CC: se envía con una copia

-BCC: para varios destinatarios

setSubject: el asunto

setText: el contenido del correo

El objeto Transport es el responsable de que se pueda enviar correctamente el correo al destinatario.



Geteamos de la sesión el tipo de transporte que habíamos establecido que es el smtp, que es el servidor de salida, luego hacemos la conexión con el correo y contraseña que habíamos creado y por último enviamos el mensaje y usamos el allRecipents para getear todos los tipos de envío de correo y cerramos el transporte.

# Requisito

Para que todo funcione correctamente desde nuestra aplicación, deberemos aplicar un ajuste en nuestra cuenta de correo.

Nos iremos a gestionar cuenta de Google, apartado de Seguridad y en Acceso de aplicaciones pocas seguras lo activaremos.



