

**Alumnos—**

JORGE ALAN ZAMORA MEJIA

JORGE EDUARDO ELIAS CAZAREZ

SEBASTIAN MARTINEZ SOQUI

**Asignación—**

Patrones de Diseño

**Fecha—**

15 de septiembre de 2024

**Materia—**

Arquitectura de Software

**Profesor—**

Felix Daniel Campoa Toledo

**Documento de Reconfiguración del Componente de Envío de Correos Electrónicos**

**1. Introducción**

El documento tiene como objetivo el describir el proceso para reconfigurar el componente de envío de correos electrónicos. El componente está diseñado para ser flexible y permitir la reconfiguración sin necesidad de modificar o regenerar el archivo .jar. La reconfiguración puede realizarse al vuelo y cuenta con la intención de ser transparente para el cliente.

**2. Estructura del Componente**

El componente de envío de correos electrónicos se compone de las siguientes partes principales:

* **Clases de Configuración**: MailConfig y su MailConfigBuilder.
* **Estrategias de Envío**: MailSenderStrategy, MailSenderApache, MailSenderJava, MailSenderSendgrid.
* **Conector**: ConectorMailSender que implementa SubsistemaMailSender.

**3. Configuración Dinámica (Configuración de Servidores de Correo)**

**Definición del Servidor:**

El servidor se define en la clase MailConfig mediante el atributo server.

**Configuración del Servidor:**

* host: Dirección del servidor SMTP.
* port: Puerto a utilizar (por ejemplo, 465 para SSL, 587 para TLS).
* encryption: Tipo de cifrado (por ejemplo, SSL o TLS).
* protocol: Protocolo utilizado para el envío (SMTP, etc.).
* password: Contraseña para la autenticación del servidor.

**Actualización de Configuración:**

Para actualizar la configuración, se debe crear una nueva instancia de MailConfig utilizando el MailConfigBuilder con los nuevos parámetros. Ejemplo de código para actualizar configuración:

MailConfig mailConfig = new MailConfig.MailConfigBuilder()

.setProtocol("SMTP")

.setServer("smtp.example.com")

.setHost("smtp.example.com")

.setPort(587)

.setEncryption("STARTTLS")

.setPassword("newpassword")

.build();

**Selección del Mecanismo de Envío**

**Definición del Mecanismo:**

Se puede elegir entre diferentes mecanismos de envío implementados por MailSenderApache, MailSenderJava, y MailSenderSengrid.

**Actualización del Mecanismo:**

La clase MailSender utiliza un patrón Builder para seleccionar el mecanismo de envío adecuado. Se debe configurar el MailSender con la instancia correspondiente. Ejemplo de código para seleccionar un mecanismo de envío:

MailSender mailSender = new MailSender.MailSenderBuilder()

.setBuilder(new ConcreteBuilder())

.setMailConfig(mailConfig)

.setEmisor("emisor@example.com")

.setRemitente("destinatario@example.com")

.setAsunto("Asunto del correo")

.setMensaje("Contenido del correo")

.build();

**4. Interfaz de Reconfiguración**

**Configuración desde la Aplicación Cliente**

La aplicación cliente debe interactuar con el componente a través de la interfaz SubsistemaMailSender, utilizando el método enviarCorreo. El método acepta parámetros que definen el correo y la configuración, permitiendo la reconfiguración dinámica sin necesidad de cambiar el .jar.

**Ejemplo de Uso**

**Configuración Inicial:**

Ejemplo de configuración en la aplicación cliente:

ConectorMailSender conector = new ConectorMailSender();

boolean resultado = conector.enviarCorreo(

"emisor@example.com",

"destinatario@example.com",

"Asunto del correo",

"Contenido del correo",

"SMTP",

"smtp.example.com",

"smtp.example.com",

"STARTTLS",

"password",

587

);

**Reconfiguración Dinámica:**

Para cambiar la configuración, simplemente se deben proporcionar los nuevos parámetros al método enviarCorreo sin necesidad de modificar el código fuente del componente.

**5. Conclusión**

Este documento proporciona una guía completa sobre cómo reconfigurar el componente de envío de correos electrónicos. La flexibilidad del componente permite ajustes que facilitan su integración y uso en diferentes contextos y con distintos servidores de correo electrónico.