Icono

Descripción generada automáticamente

Ejercicio1. Viola el Principio de Responsabilidad Única (SRP)

**public class AuditTransfMonet {**

**public void transferenciaRealizada(Transferencia transferencia) {**

**if(this.esTransferenciaImportante(transferencia)) {**

**String auditor = this.obtenerDireccionMailAuditor();**

**String mensaje = this.componerMensajeAviso(transferencia);**

**ConexionMail conexionMail = null;**

**try {**

**conexionMail = this.abrirConexionMail();**

**conexionMail.enviar(new Mail().to(auditor).withBody(mensaje));**

**} finally {**

**if(conexionMail != null)**

**conexionMail.cerrar();**

**}**

**}**

**}**

**private boolean esTransferenciaImportante(Transferencia transferencia) {**

**return transferencia.importe() > 50000;**

**}**

**private String obtenerDireccionMailAuditor() {**

**return "mail-auditor";**

**}**

**private String componerMensajeAviso(Transferencia transferencia) {**

**return "aviso-transferencia-importante";**

**}**

**private ConexionMail abrirConexionMail() {**

**return ConexionMail.getInstance();**

**}**

**}**

*Código corregido*

**public class AuditService {**

**private TransferenciaValidator validator;**

**private MailService mailService;**

**public AuditService(TransferenciaValidator validator, MailService mailService) {**

**this.validator = validator;**

**this.mailService = mailService;**

**}**

**public void transferenciaRealizada(Transferencia transferencia) {**

**if (validator.esTransferenciaImportante(transferencia)) {**

**String auditor = mailService.obtenerDireccionMailAuditor();**

**String mensaje = mailService.componerMensajeAviso(transferencia);**

**mailService.enviarMail(auditor, mensaje);**

**}**

**}**

**}**

**public class TransferenciaValidator {**

**public boolean esTransferenciaImportante(Transferencia transferencia) {**

**return transferencia.importe() > 50000;**

**}**

**}**

**public class MailService {**

**public String obtenerDireccionMailAuditor() {**

**return "mail-auditor";**

**}**

**public String componerMensajeAviso(Transferencia transferencia) {**

**return "aviso-transferencia-importante";**

**}**

**public void enviarMail(String auditor, String mensaje) {**

**ConexionMail conexionMail = null;**

**try {**

**conexionMail = this.abrirConexionMail();**

**conexionMail.enviar(new Mail().to(auditor).withBody(mensaje));**

**} finally {**

**if (conexionMail != null)**

**conexionMail.cerrar();**

**}**

**}**

**private ConexionMail abrirConexionMail() {**

**return ConexionMail.getInstance();**

**}**

**}**

### Problema inicial:

En el código original de la clase AuditTransfMonet, se estaban manejando múltiples responsabilidades dentro de una misma clase:

* Validar si una transferencia es importante.
* Obtener la dirección de correo del auditor.
* Componer el mensaje de notificación.
* Gestionar la conexión de correo y enviar el mail.

Esto **viola el SRP**, ya que una clase debería tener solo una razón para cambiar. Si en algún momento quisiéramos cambiar la lógica de validación, la forma de componer el mensaje o la manera de enviar el correo, tendríamos que modificar esta clase, lo cual la hace propensa a errores y de difícil mantenimiento.

### Corrección aplicada:

Para corregir el código según el **SRP(Principio de responsabilidad unica)**, lo que hice fue **dividir las responsabilidades** en clases separadas:

1. **AuditService**: Esta clase es responsable únicamente de coordinar el proceso de auditoría y notificación cuando una transferencia importante es realizada. No hace el trabajo directo de validar o enviar correos.
2. **TransferenciaValidator**: Se encarga de validar si una transferencia es importante o no. Esta clase cumple una única función: la validación.
3. **MailService**: Es la clase responsable de todo lo relacionado con los correos electrónicos: obtener la dirección del auditor, componer el mensaje y enviar el mail. Toda la lógica relacionada con el envío de correos está contenida en esta clase.