

Código de un Módulo del programa

Módulo elegido:

Creación de Reportes e Incidencias.

Parte 1:

```

public class ReporteSoporte {

    // Atributos
    private String correo;
    private String tipoReporte;
    private String mensaje;
    private String estado; // Por ejemplo: "Pendiente", "En proceso", "Resuelto"
    private String fechaEnvio;

    // Constructor
    public ReporteSoporte(String correo, String tipoReporte, String mensaje, String fechaEnvio) {
        this.correo = correo;
        this.tipoReporte = tipoReporte;
        this.mensaje = mensaje;
        this.estado = "Pendiente";
        this.fechaEnvio = fechaEnvio;
    }

    // Método implementado
    public String generarConfirmacion(String nombreUsuario) {
        return "Hola " + nombreUsuario + ", hemos recibido tu reporte sobre \""
            + tipoReporte + "\". Nuestro equipo de soporte te contactará al correo "
            + correo + " en un plazo de 48 horas.";
    }

    // Getters y setters (no es obligatorio implementarlos todos para este ejercicio)
    public String getCorreo() {
        return correo;
    }

    public void setCorreo(String correo) {
        this.correo = correo;
    }

    public String getTipoReporte() {
        return tipoReporte;
    }

    public void setTipoReporte(String tipoReporte) {
        this.tipoReporte = tipoReporte;
    }

    public String getMensaje() {
        return mensaje;
    }

    public void setMensaje(String mensaje) {
        this.mensaje = mensaje;
    }
}

```

```

    }

    public String getEstado() {
        return estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public String getFechaEnvio() {
        return fechaEnvio;
    }

    public void setFechaEnvio(String fechaEnvio) {
        this.fechaEnvio = fechaEnvio;
    }
}

```

Cuando el usuario accede a la sección de "Compartir reportes", rellena un formulario con su correo, el tipo de problema y un mensaje descriptivo. Al pulsar el botón de "Enviar", los datos se envían al servidor, donde se procesa la información.

En la parte del backend, se crea una instancia de la clase **ReporteSoporte** utilizando los datos del formulario. Esta clase contiene atributos como el correo electrónico, el tipo de reporte, el mensaje del usuario, la fecha de envío y el estado del reporte.

Una vez creado el objeto, se puede invocar el método **generarConfirmacion**, que recibe el nombre del usuario y devuelve un mensaje confirmando la recepción del reporte. Este mensaje puede mostrarse en la interfaz o enviarse por correo como acuse de recibo.

Después de eso, el reporte puede almacenarse en el sistema para que el equipo de soporte lo revise y gestione según su estado. Aunque esta parte no está implementada, la clase está preparada para integrarse con procesos de almacenamiento, seguimiento y respuesta.

Parte 2:

```

import static org.junit.Assert.assertEquals;
import org.junit.Test;

public class ReporteSoporteTest {

    @Test
    public void testGenerarConfirmacion() {
        // Datos de entrada simulados
        ReporteSoporte reporte = new ReporteSoporte(
            "usuario@ejemplo.com",
            "Problemas inicio de sesión y permisos",
            "No puedo iniciar sesión desde ayer.",
            "2025-05-29"
        );

        // Ejecución del método con parámetro
        String resultado = reporte.generarConfirmacion("Carlos");

        // Resultado esperado
        String esperado = "Hola Carlos, hemos recibido tu reporte sobre \"Problemas inicio de sesión y permisos\". Nuestro equipo de soporte te contactará al correo usuario@ejemplo.com en un plazo de 48 horas.";

        // Verificación
        assertEquals(esperado, resultado);
    }
}

```

Prueba Junit caja negra. El resultado es el esperado, al llamarlo crea un reporte con los datos que se introduzcan en el formulario, y podemos verificar que la salida es exactamente lo esperado.

Prueba Junit caja blanca. El método al ser muy sencillo no tiene ramas secundarias con bucles o condiciones, por lo tanto tiene un esquema muy básico por lo que sería:



