



TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE OAXACA

Carrera:

Ingeniería en Sistemas Computacionales

Materia:

Diseño e Implementación de software con patrones

Documentación de patrón Template

Docente:

Espinoza Pérez Jacob

Equipo:

Ordaz Pacheco Ruudvan

Santos Manuel Julia Marlenny

Vera Acevedo Héctor Aramís

Grupo:

7SB

Fecha de entrega:

11/05/2025

Documentación del Patrón Template en el Sistema de Ventas

Introducción

Este documento explica la implementación del **patrón Template Method** en el módulo de generación de reportes del sistema de ventas. El objetivo es estandarizar la creación de reportes en PDF, eliminando duplicación de código y facilitando el mantenimiento.

Clases Principales

3.1. ReporteTemplate (Clase Abstracta)

Responsabilidad: Define la estructura común para todos los reportes.

Métodos Principales

Método	Tipo	Descripción
generarReporte()	final	Esqueleto del algoritmo: Controla el flujo de generación del PDF.
configurarDocumento()	private	Establece tamaño y márgenes del PDF.
agregarCabecera()	private	Añade logo y título genérico.
agregarTitulo()	private	Inserta el título específico del reporte.
generarContenido()	abstract	Paso variable: Cada reporte implementa su contenido.
crearTabla()	abstract	Paso variable: Crea la tabla con columnas personalizadas.
llenarTabla()	abstract	Paso variable: Llena la tabla con datos de la base de datos.

Flujo de Generación

1. Configura el documento PDF.
2. Agrega cabecera y título.
3. Delega la creación del contenido a las subclases.

```

public abstract class ReporteTemplate {
    protected Document documento;
    protected String tituloReporte;
    protected String nombreArchivo;

    public final void generarReporte() {
        documento = new Document();
        try {
            String ruta = System.getProperty("user.home");
            PdfWriter.getInstance(documento, new FileOutputStream(ruta + "/Desktop/" + nombreArchivo + ".pdf"));

            // Configuración común del documento
            configurarDocumento();

            documento.open();

            // Agregar cabecera común
            agregarCabecera();

            // Agregar título específico
            agregarTitulo();

            // Generar contenido específico del reporte
            generarContenido();

            documento.close();

            JOptionPane.showMessageDialog(null, "Reporte creado: " + nombreArchivo);

        } catch (DocumentException | FileNotFoundException e) {
            System.out.println("Error al crear el documento: " + e);
        } catch (IOException e) {
            System.out.println("Error de IO: " + e);
        }
    }

    private void configurarDocumento() throws DocumentException, IOException {
        // Configuración común para todos los reportes
        documento.setPageSize(PageSize.A4);
        documento.setMargins(30, 30, 30, 30);
    }

    private void agregarCabecera() throws DocumentException, IOException {
        Image header = Image.getInstance("src/img/header1.jpg");
        header.scaleToFit(650, 1000);
        header.setAlignment(Chunk.ALIGN_CENTER);
        documento.add(header);

        Paragraph parrafo = new Paragraph();
        parrafo.setAlignment(Paragraph.ALIGN_CENTER);
        parrafo.add("Reporte creado por \nMiscelanea Calicanto\n\n");
        parrafo.setFont(FontFactory.getFont("Tahoma", 18, Font.BOLD, BaseColor.DARK_GRAY));
        documento.add(parrafo);
    }

    private void agregarTitulo() throws DocumentException {
        Paragraph titulo = new Paragraph(tituloReporte + "\n\n");
        titulo.setAlignment(Paragraph.ALIGN_CENTER);
        titulo.setFont(FontFactory.getFont("Tahoma", 16, Font.BOLD, BaseColor.BLACK));
        documento.add(titulo);
    }

    // Métodos abstractos que deben implementar las subclases
    protected abstract void generarContenido() throws DocumentException;
    protected abstract PdfPTable crearTabla() throws DocumentException;
    protected abstract void llenarTabla(PdfPTable tabla) throws DocumentException, SQLException;
}

```

Clases Concretas

ReporteClientes

```
public class ReporteClientes extends ReporteTemplate {

    public ReporteClientes() {
        this.tituloReporte = "Reporte de Clientes";
        this.nombreArchivo = "Reporte_Clientes";
    }

    @Override
    protected void generarContenido() throws DocumentException {
        PdfPTable tabla = crearTabla();
        try {
            llenarTabla(tabla);
        } catch (SQLException e) {
            System.out.println("Error en la consulta SQL: " + e);
        }
        documento.add(tabla);
    }

    @Override
    protected PdfPTable crearTabla() throws DocumentException {
        PdfPTable tabla = new PdfPTable(5);
        tabla.addCell("Codigo");
        tabla.addCell("Nombres");
        tabla.addCell("Cedula");
        tabla.addCell("Telefono");
        tabla.addCell("Direccion");
        return tabla;
    }

    @Override
    protected void llenarTabla(PdfPTable tabla) throws DocumentException, SQLException {
        Connection cn = Conexion.conectar();
        PreparedStatement pst = cn.prepareStatement(
            "select idCliente, concat(nombre, ' ', apellido) as nombres, cedula, telefono, direccion from tb_cliente");
        ResultSet rs = pst.executeQuery();

        while (rs.next()) {
            tabla.addCell(rs.getString(1));
            tabla.addCell(rs.getString(2));
            tabla.addCell(rs.getString(3));
            tabla.addCell(rs.getString(4));
            tabla.addCell(rs.getString(5));
        }
    }
}
```

ReporteCategoria

```
public class ReporteCategorias extends ReporteTemplate {

    public ReporteCategorias() {
        this.tituloReporte = "Reporte de Categorías";
        this.nombreArchivo = "Reporte_Categorias";
    }

    @Override
    protected void generarContenido() throws DocumentException {
        PdfPTable tabla = crearTabla();
        try {
            llenarTabla(tabla);
            documento.add(tabla);
        } catch (SQLException e) {
            System.out.println("Error en la consulta SQL: " + e);
        }
    }

    @Override
    protected PdfPTable crearTabla() throws DocumentException {
        PdfPTable tabla = new PdfPTable(3);
        tabla.addCell("Codigo");
        tabla.addCell("Descripcion");
        tabla.addCell("Estado");
        return tabla;
    }

    @Override
    protected void llenarTabla(PdfPTable tabla) throws DocumentException, SQLException {
        Connection cn = Conexion.conectar();
        PreparedStatement pst = cn.prepareStatement("select * from tb_categoria");
        ResultSet rs = pst.executeQuery();

        while (rs.next()) {
            tabla.addCell(rs.getString(1));
            tabla.addCell(rs.getString(2));
            tabla.addCell(rs.getString(3));
        }
    }
}
```

ReporteProductos

```
public class ReporteProductos extends ReporteTemplate {

    public ReporteProductos() {
        this.tituloReporte = "Reporte de Productos";
        this.nombreArchivo = "Reporte_Productos";
    }

    @Override
    protected void generarContenido() throws DocumentException {
        float[] columnsWidths = {3, 5, 4, 5, 7, 5, 6};
        PdfPTable tabla = new PdfPTable(columnsWidths);
        tabla.addCell("Codigo");
        tabla.addCell("Nombre");
        tabla.addCell("Cant.");
        tabla.addCell("Precio");
        tabla.addCell("Descripcion");
        tabla.addCell("Por. Iva");
        tabla.addCell("Categoria");

        try {
            Connection cn = Conexion.conectar();
            PreparedStatement pst = cn.prepareStatement(
                "select p.idProducto, p.nombre, p.cantidad, p.precio, p.descripcion, "
                + "p.porcentajeIva, c.descripcion as categoria, p.estado "
                + "from tb_producto as p, tb_categoria as c "
                + "where p.idCategoria = c.idCategoria;");
            ResultSet rs = pst.executeQuery();

            while (rs.next()) {
                tabla.addCell(rs.getString(1));
                tabla.addCell(rs.getString(2));
                tabla.addCell(rs.getString(3));
                tabla.addCell(rs.getString(4));
                tabla.addCell(rs.getString(5));
                tabla.addCell(rs.getString(6));
                tabla.addCell(rs.getString(7));
            }

            documento.add(tabla);
        } catch (SQLException e) {
            System.out.println("Error en la consulta SQL: " + e);
        }
    }

    // Implementación de métodos abstractos no utilizados en este caso
    @Override
    protected PdfPTable crearTabla() throws DocumentException {
        return null;
    }

    @Override
    protected void llenarTabla(PdfPTable tabla) throws DocumentException, SQLException {
    }
}
```

ReporteVentas

```
public class ReporteVentas extends ReporteTemplate {

    public ReporteVentas() {
        this.tituloReporte = "Reporte de Ventas";
        this.nombreArchivo = "Reporte_Ventas";
    }

    @Override
    protected void generarContenido() throws DocumentException {
        float[] columnsWidths = {3, 9, 4, 5, 3};
        PdfPTable tabla = new PdfPTable(columnsWidths);
        tabla.addCell("Codigo");
        tabla.addCell("Cliente");
        tabla.addCell("Tot. Pagar");
        tabla.addCell("Fecha Venta");
        tabla.addCell("Estado");

        try {
            Connection cn = Conexion.conectar();
            PreparedStatement pst = cn.prepareStatement(
                "select cv.idCabeceraVenta as id, concat(c.nombre, ' ', c.apellido) as cliente, "
                + "cv.valorPagar as total, cv.fechaVenta as fecha, cv.estado "
                + "from tb_cabecera_venta as cv, tb_cliente as c "
                + "where cv.idCliente = c.idCliente;");
            ResultSet rs = pst.executeQuery();

            while (rs.next()) {
                tabla.addCell(rs.getString(1));
                tabla.addCell(rs.getString(2));
                tabla.addCell(rs.getString(3));
                tabla.addCell(rs.getString(4));
                tabla.addCell(rs.getString(5));
            }

            documento.add(tabla);
        } catch (SQLException e) {
            System.out.println("Error en la consulta SQL: " + e);
        }
    }

    @Override
    protected PdfPTable crearTabla() throws DocumentException {
        return null; // No se usa porque se crea directamente en generarContenido()
    }

    @Override
    protected void llenarTabla(PdfPTable tabla) throws DocumentException, SQLException {
        // No se usa porque se maneja todo en generarContenido()
    }
}
```


ReporteVentasDias

```
public class ReporteVentasDia extends ReporteTemplate {

    public ReporteVentasDia() {
        this.tituloReporte = "Reporte de Ventas del Día";
        this.nombreArchivo = "Reporte_Ventas_Dia";
    }

    @Override
    protected void generarContenido() throws DocumentException {
        PdfPTable tabla = crearTabla();
        double totalVentasDia = 0.0;

        try {
            Connection cn = Conexion.conectar();
            PreparedStatement pst = cn.prepareStatement(
                "SELECT cv.idCabeceraVenta AS id, "
                + "CONCAT(c.nombre, ' ', c.apellido) AS cliente, "
                + "cv.valorPagar AS total, cv.fechaVenta AS fecha, cv.estado "
                + "FROM tb_cabecera_venta AS cv "
                + "JOIN tb_cliente AS c ON cv.idCliente = c.idCliente "
                + "WHERE DATE(cv.fechaVenta) = CURRENT_DATE;");

            ResultSet rs = pst.executeQuery();

            if (rs.next()) {
                do {
                    tabla.addCell(rs.getString(1));
                    tabla.addCell(rs.getString(2));
                    tabla.addCell(rs.getString(3));
                    tabla.addCell(rs.getString(4));
                    tabla.addCell(rs.getString(5));

                    totalVentasDia += rs.getDouble(3);
                } while (rs.next());

                documento.add(tabla);
                documento.add(new Paragraph("\n\nTotal de Ventas del Día: " + totalVentasDia + " MXN"));
            } else {
                documento.add(new Paragraph("\n\nNo se han realizado ventas hoy.));
            }
        } catch (SQLException e) {
            System.out.println("Error en la consulta SQL: " + e);
        }
    }

    @Override
    protected PdfPTable crearTabla() throws DocumentException {
        float[] columnsWidths = {3, 9, 4, 5, 3};
        PdfPTable tabla = new PdfPTable(columnsWidths);
        tabla.addCell("Codigo");
        tabla.addCell("Cliente");
        tabla.addCell("Tot. Pagar");
        tabla.addCell("Fecha Venta");
        tabla.addCell("Estado");
        return tabla;
    }

    @Override
    protected void llenarTabla(PdfPTable tabla) throws DocumentException, SQLException {
        // Implementación vacía porque ya se maneja en generarContenido()
    }
}
```

Uso del Sistema

4.1. Generar un Reporte

```
ReporteTemplate reporte = new ReporteClientes();  
reporte.generarReporte();
```

4.2. Clase Reportes (Facade)

Simplifica la interacción con los reportes:

```
public class Reportes {  
    public void generarReporteClientes() {  
        new ReporteClientes().generarReporte();  
    }  
    // ... métodos para otros reportes  
}
```

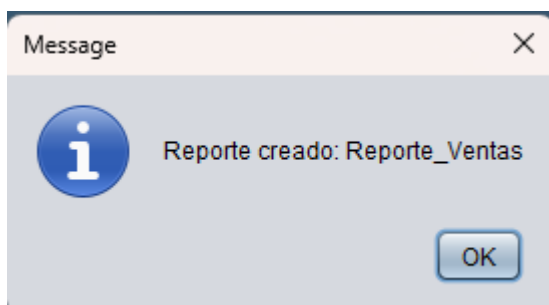
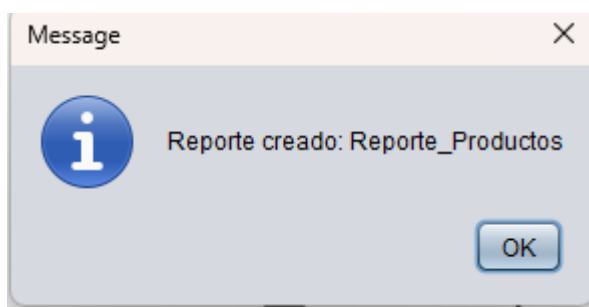
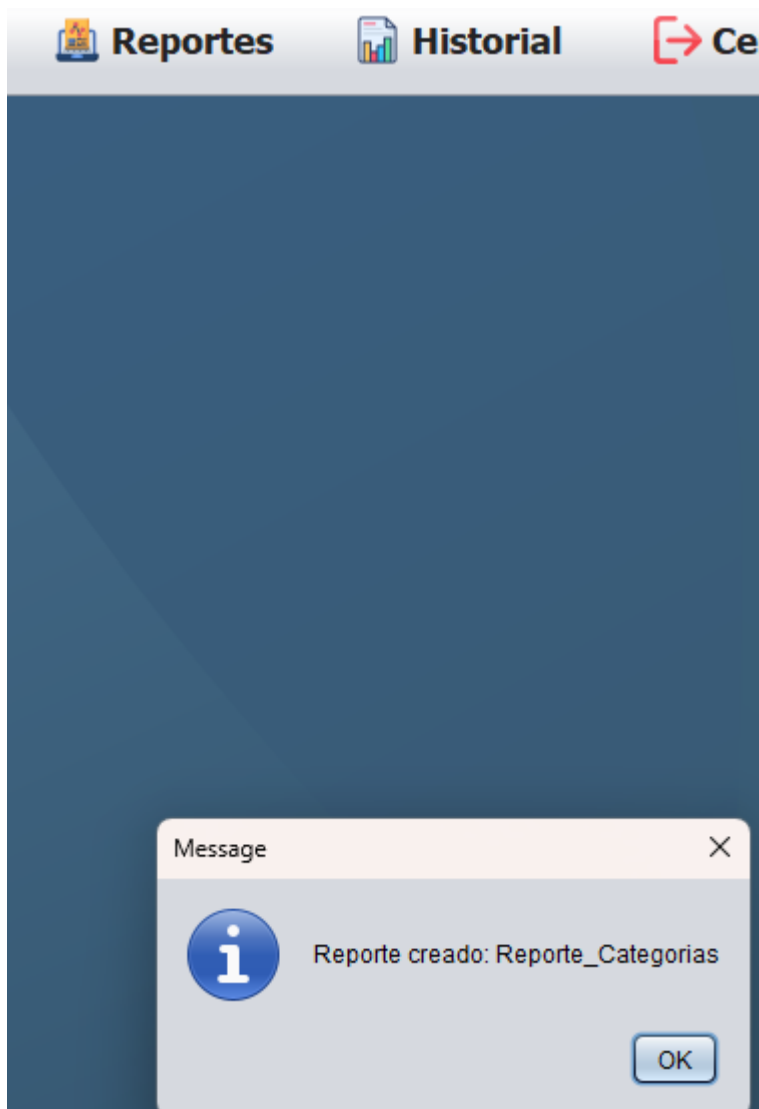
Uso desde el menú:

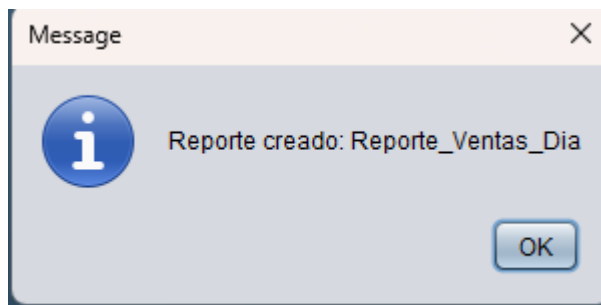
```
private void jMenuItem_reportes_clientesActionPerformed() {  
    new Reportes().generarReporteClientes();  
}
```

Beneficios:

- **Estandarización:** Todos los reportes siguen la misma estructura.
- **Mantenibilidad:** Cambios en el formato se aplican en un solo lugar (ReporteTemplate).
- **Extensibilidad:** Añadir nuevos reportes es rápido (ej: ReporteProveedores).
- **DRY:** Elimina código duplicado en la generación de PDFs.

Resultados





Reporte creado por
Miscelanea Calicanto

Reporte de Categorías

Codigo	Descripcion	Estado
1	Vinos y Licores	1
2	Lacteos	1
3	Frutas Y verduras	1
4	Sabritas	1
5	Tostadas	1
6	Bebidas Energetizantes	1
7	Refrescos	1

Reporte creado por
Miscelanea Calicanto

Reporte de Clientes

Codigo	Nombres	Cedula	Telefono	Direccion
1	Uri Hernandez	9514401725	9514401725	Av Montoya
2	Teresa Hernandez	9581732455	9581732455	Privada Texcoco
3	Alma Rosa Mendez	9512156554	9512156554	Privada Texcoco
6	abdiel Tomas Diaz	9514565656	9514565656	Av Tecnologico
7	Hector Vera	9512348721	9512348721	Tecnologico

Reporte creado por
Miscelanea Calicanto

Reporte de Productos

Codig o	Nombre	Cant.	Precio	Descripcion	Por. Iva	Categoria
1	Cerveza	143	25.00	Corona	0	Vinos y Licores
6	Yoghurt Oikos	0	19.00		0	Vinos y Licores
30	Vino tinto	32	12.00	Licor caro	12	Vinos y Licores
2	Leche Nutri Deslactosa	4	28.00	Leche Deslactosada	0	Lacteos
14	sabritas	2	56.00		0	Lacteos
15	dwed	34	23.40	adwda	0	Lacteos
18	awdawed	123	21.00	adwd	0	Lacteos

Reporte creado por
Miscelanea Calicanto

Reporte de Ventas

Codigo	Cliente	Tot. Pagar	Fecha Venta	Estado
1	Uri Hernandez	1254.00	2024-10-24	1
2	Uri Hernandez	50.16	2024-10-24	1
3	Uri Hernandez	50.16	2024-10-25	1
4	Uri Hernandez	125.40	2024-10-28	1
5	Uri Hernandez	125.40	2024-10-28	1
9	Uri Hernandez	168.00	2024-11-07	1
10	Uri Hernandez	194.00	2024-11-07	1
11	Uri Hernandez	72.00	2024-11-07	1
17	Uri Hernandez	76.00	2024-12-02	1
19	Uri Hernandez	294.00	2024-12-05	1
25	Uri Hernandez	56.00	2025-03-22	1

Reporte creado por
Miscelanea Calicanto

Reporte de Ventas del Día

Codigo	Cliente	Tot. Pagar	Fecha Venta	Estado
47	Uri Hernandez	36.00	2025-05-11	1
48	Alma Rosa Mendez	86.00	2025-05-11	1

Total de Ventas del Día: 122.0 MXN

Conclusión

La implementación del **Template Method** ha transformado el sistema de reportes en:

- **Un sistema modular**, donde cada tipo de reporte se desarrolla de forma independiente pero

Coherente

- **Una base escalable**, lista para incorporar nuevos formatos y funcionalidades.
- **Un código más limpio y mantenible**, alineado con buenas prácticas de diseño orientado a objetos.