



TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE OAXACA

Carrera:

Ingeniería en Sistemas Computacionales

Materia:

Diseño e Implementación de software con patrones

Documentación

Docente:

Espinoza Pérez Jacob

Equipo:

Ordaz Pacheco Ruudvan

Santos Manuel Julia Marlenny

Vera Acevedo Héctor Aramís

Grupo:

7SB

Fecha de entrega:

24/03/2025

Documentación del Patrón Composite en el Sistema de Ventas

Introducción

El patrón Composite es un patrón de diseño estructural que permite tratar objetos individuales y composiciones de objetos de manera uniforme. En el contexto de la clase Reportes, este patrón se utiliza para generar reportes de manera individual (clientes, productos, categorías, ventas) o en conjunto (todos los reportes), sin necesidad de modificar significativamente el código existente.

Estructura del Patrón Composite en Reportes

El patrón Composite se compone de los siguientes elementos:

1. **Componente (ReporteComponente)**: Es una interfaz que define el método común para generar reportes.
2. **Hoja (ReporteCliente, ReporteProducto, ReporteCategoria, ReporteVenta)**: Representa los reportes individuales.
3. **Compuesto (ReporteCompuesto)**: Representa una colección de reportes que pueden ser generados juntos.
4. **Cliente (Reportes)**: Es la clase que utiliza los componentes (hojas y compuestos) para generar los reportes.

Cambios Realizados

Interfaz ReporteComponente

Esta interfaz define el método común para generar reportes.

```
1  /**  
2  public interface ReporteComponente {  
3      void generarReporte ();  
4  }
```

Clases Hoja (ReporteCliente, ReporteProducto, ReporteCategoria, ReporteVenta)

Cada clase hoja implementa la interfaz ReporteComponente y contiene la lógica para generar un tipo específico de reporte.

```
public class ReporteCliente implements ReporteComponente {
    @Override
    public void generarReporte() {
        Document documento = new Document();
        try {
            String ruta = System.getProperty("user.home");
            PdfWriter.getInstance(documento, new FileOutputStream(ruta + "/Desktop/Reporte_Clientes.pdf"));
            Image header = Image.getInstance("src/img/header1.jpg");
            header.scaleToFit(650, 1000);
            header.setAlignment(Chunk.ALIGN_CENTER);

            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));
            parrafo.add("Reporte de Clientes \n\n");

            documento.open();
            documento.add(header);
            documento.add(parrafo);

            PdfPTable tabla = new PdfPTable(5);
            tabla.addCell("Codigo");
            tabla.addCell("Nombres");
            tabla.addCell("Cedula");
            tabla.addCell("Telefono");
            tabla.addCell("Direccion");

            try {
                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();
                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));
                    } while (rs.next());
                    documento.add(tabla);
                }
            }

            documento.close();
            JOptionPane.showMessageDialog(null, "Reporte de Clientes creado");

        } catch (DocumentException e) {
            System.out.println("Error 1 en: " + e);
        } catch (FileNotFoundException ex) {
            System.out.println("Error 2 en: " + ex);
        } catch (IOException ex) {
            System.out.println("Error 3 en: " + ex);
        }
    }
}
```

Clase Compuesto (ReporteCompuesto)

Esta clase representa una colección de reportes y también implementa la interfaz ReporteComponente.

```
public class ReporteCompuesto implements ReporteComponente {
    private List<ReporteComponente> reportes = new ArrayList<>();

    public void agregarReporte(ReporteComponente reporte) {
        reportes.add(reporte);
    }

    public void eliminarReporte(ReporteComponente reporte) {
        reportes.remove(reporte);
    }

    @Override
    public void generarReporte() {
        for (ReporteComponente reporte : reportes) {
            reporte.generarReporte();
        }
    }
}
```

Clase Reportes Modificada

La clase Reportes ahora utiliza el patrón Composite para generar reportes individuales o compuestos.

```
public class Reportes {

    /* *****
    * metodo para crear reportes de los clientes registrados en el sistema
    ***** */
    public void generarReporteCompuesto() {
        ReporteCompuesto reporteCompuesto = new ReporteCompuesto();

        // Agregar reportes individuales al compuesto
        reporteCompuesto.agregarReporte(new ReporteCliente());
        reporteCompuesto.agregarReporte(new ReporteProducto());
        reporteCompuesto.agregarReporte(new ReporteCategoria());
        reporteCompuesto.agregarReporte(new ReporteVentas());

        // Generar todos los reportes
        reporteCompuesto.generarReporte();
    }

    public void generarReporteIndividual(ReporteComponente reporte) {
        reporte.generarReporte();
    }
}
```

Integración con la Interfaz Gráfica

La interfaz gráfica contiene botones para generar reportes individuales o compuestos. Cada botón está asociado a un evento que llama al método correspondiente en la clase Reportes.

```
public class FrmMenu extends javax.swing.JFrame {  
  
    private Reportes reportes;  
    public static JDesktopPane jDesktopPane_menu;
```

```
    private void jMenuItem_reportes_clientesActionPerformed(java.awt.event.ActionEvent evt) {  
        Reportes reporte = new Reportes();  
        reportes.generarReporteIndividual(new ReporteCliente());  
    }
```

```
    private void jMenuItem_reportes_productosActionPerformed(java.awt.event.ActionEvent evt) {  
        Reportes reporte = new Reportes();  
        reportes.generarReporteIndividual(new ReporteProducto());  
    }
```

```
    private void jMenuItem_reportes_categoriasActionPerformed(java.awt.event.ActionEvent evt) {  
        Reportes reporte = new Reportes();  
        reportes.generarReporteIndividual(new ReporteCategoria());  
    }
```

```
    private void jMenuItem_ver_historialActionPerformed(java.awt.event.ActionEvent evt) {  
        InterGraficas interGraficas = new InterGraficas();  
        jDesktopPane_menu.add(interGraficas);  
        interGraficas.setVisible(true);  
    }
```

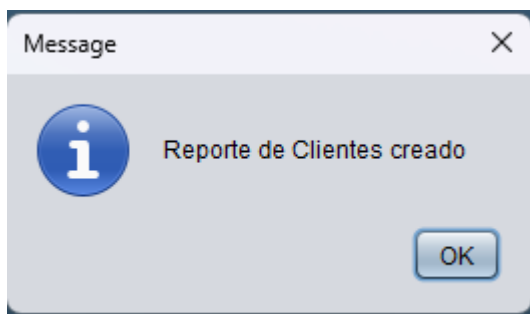
```
    private void jMenuItem_reportes_ventas_diaActionPerformed(java.awt.event.ActionEvent evt) {  
        Reportes reporte = new Reportes();  
        reportes.generarReporteIndividual(new ReporteVentasDias());  
    }
```

Ventajas del Patrón Composite en este Contexto

1. Tratamiento Uniforme: Permite tratar reportes individuales y compuestos de la misma manera.
2. Flexibilidad: Facilita la adición de nuevos tipos de reportes sin modificar el código existente.
3. Reutilización: El código para generar reportes individuales y compuestos se reutiliza fácilmente.

Resultados de la implementación del patrón Composite.

Reporte Clientes



Escritorio			
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div>Ordenar</div> <div>Ver</div> <div></div> </div> </div>			
Nombre	Fecha de modificación	Tipo	Tamaño
Acceso directo a Escritorio (OneDrive - P...	16/03/2025 07:11 p. m.	Acceso directo	2 KB
CurseForge	09/03/2025 09:18 p. m.	Acceso directo	3 KB
Discord	11/03/2025 10:36 a. m.	Acceso directo	3 KB
Genshin Impact	17/03/2025 12:11 a. m.	Acceso directo a l...	1 KB
Reporte_Clientes	23/03/2025 03:23 a. m.	Brave HTML Docu...	31 KB

Archivo C:/Users/hecto/Desktop/Reporte_Clientes.pdf

de marcadores: [Importar marcadores ahora...](#)

1 / 1 | 100%

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
5	Alberto Mendez	9511111212	9511111212	Texcoco 6
6	abdiel Diaz	9514565656	9514565656	Av montealban



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
8	Radiadores	1

Reporte creado por
Miscelanea Calicanto

Reporte de Productos

Codig o	Nombre	Cant.	Precio	Descripcion	Por. Iva	Categoria
1	Cerveza	80	25.00	Corona	0	Vinos y Licores
6	Yoghurt Oikos	0	19.00		0	Vinos y Licores
2	Leche Nutri Deslactos a	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
23	peokd	431	74.00	adwdaw	12	Lacteos
16	dwad	12	31.10	awdawdsd	0	Frutas Y verduras
17	dawdaw	123	32.00	adwdwa	0	Frutas Y verduras
22	dwda	234	34.00	awdawd	0	Frutas Y verduras
3	Sabritas Clasicas	16	18.00		0	Sabritas
4	Paquetax o Azul Ch	0	18.00		0	Sabritas
5	Cheetos Torciditos	0	16.00		0	Sabritas

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
20	Uri Hernandez	116.00	2024-12-05	1
25	Uri Hernandez	56.00	2025-03-22	1
30	Uri Hernandez	18.00	2025-03-22	1
35	Uri Hernandez	287.00	2025-03-22	1
6	Teresa Hernandez	390.96	2024-11-02	1
8	Teresa Hernandez	84.00	2024-11-07	1

Reporte creado por
Miscelanea Calicanto

Reporte de Ventas del Día

No se han realizado ventas hoy.

Conclusión

El patrón Composite es ideal para manejar estructuras jerárquicas donde los objetos individuales y las composiciones de objetos deben tratarse de manera uniforme. En este caso, se aplicó para generar reportes de manera individual o en conjunto, lo que simplifica el código y lo hace más flexible y mantenible. La integración con la interfaz gráfica permite que el usuario genere reportes de manera intuitiva, ya sea individualmente o todos juntos.