

Implementación del Patrón Memento en el Sistema de Citas Médicas

1. INTRODUCCIÓN

El patrón Memento es útil en situaciones donde necesitas mantener un historial de estados de un objeto para poder revertir a un estado anterior. En este caso, se utiliza para permitir al usuario guardar y restaurar el estado de una cita médica, lo que puede ser especialmente útil en aplicaciones donde los usuarios pueden cometer errores al ingresar datos y necesitan una forma de deshacer cambios.

2. ¿QUÉ ES EL PATRÓN MEMENTO?

El patrón Memento es un patrón de diseño de comportamiento que permite capturar y externalizar el estado interno de un objeto sin violar la encapsulación, de modo que el objeto pueda ser restaurado a ese estado más tarde. Este patrón es especialmente útil cuando necesitas implementar funcionalidades como "deshacer" (undo), "rehacer" (redo) o "guardar y restaurar estados" en una aplicación.

Objetivo del Patrón Memento

El objetivo principal del patrón Memento es preservar el estado de un objeto en un momento dado, de manera que ese estado pueda ser restaurado posteriormente sin exponer los detalles internos del objeto. Esto es útil en situaciones donde necesitas:

Implementar funcionalidades de "deshacer" en una aplicación.

Guardar y restaurar el estado de un objeto en un sistema de persistencia.

Realizar copias de seguridad (backups) del estado de un objeto.

3. PARTICIPANTES DEL PATRÓN MEMENTO

En la implementación del patrón Memento para nuestro sistema de citas medicas los participantes son:

· **Originador (AgendarCita):**

Responsabilidad: Es el objeto cuyo estado se desea guardar y restaurar.

Atributos: Contiene los atributos que representan el estado de la cita, como id, idPaciente, idMedico, fecha, hora, consultorio, tipo, y disponible.

Métodos:

guardarEstado(): Crea y devuelve un objeto Memento que contiene el estado actual de la cita.

restaurarEstado(Memento memento): Restaura el estado de la cita a partir de un objeto Memento.

- **Memento (Memento):**

Responsabilidad: Almacena el estado interno del Originador (AgendarCita).

Atributos: Contiene los mismos atributos que el Originador (id, idPaciente, idMedico, fecha, hora, consultorio, tipo, disponible).

Características:

Es inmutable (una vez creado, su estado no puede modificarse).

Solo el Originador puede acceder a los detalles del Memento (encapsulación).

- **Caretaker (Cuidador)(Botones Guardar y Restaurar):**

Responsabilidad: Gestiona los Mementos (guardar, restaurar).

Atributos: Mantiene una referencia al Memento actual o una lista/pila de Mementos.

Métodos:

guardarEstado(Memento memento): Almacena un Memento.

restaurarEstado(): Devuelve el Memento almacenado para restaurar el estado del Originador.

4. ¿CÓMO FUNCIONA?

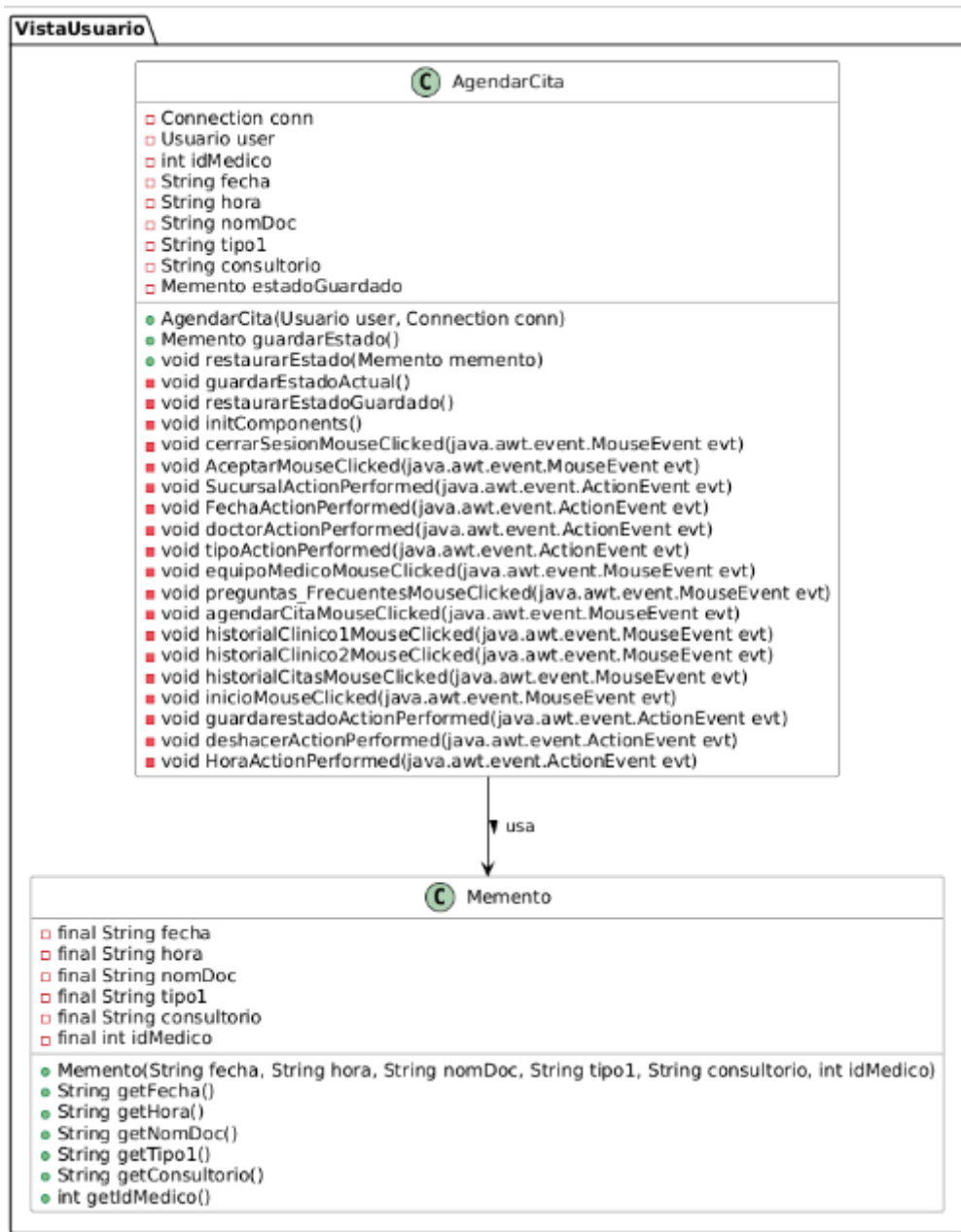
En nuestro proyecto de gestion de citas el patrón memento funciona así:

1. El usuario abre la interfaz de Agendar Cita.
2. El usuario configura los detalles de la cita:
3. El usuario hace clic en el botón "Guardar Estado":
Se ejecuta el método `guardarEstadoActual()`.
Este método llama a `guardarEstado()`, que crea un objeto Memento con el estado actual de la cita.
4. Se imprime: "Estado de la cita guardado."
5. El usuario hace clic en el botón "Deshacer":
6. Se ejecuta el método `restaurarEstadoGuardado()`. Este método verifica si hay un Memento guardado en `estadoGuardado`.
7. Si existe un Memento, se llama al método `restaurarEstado(Memento memento)`.
8. El método `restaurarEstado()` toma el Memento guardado y actualiza los atributos de la cita con los valores almacenados en el Memento.

5. IMPLEMENTACIÓN EN EL PROYECTO

En el sistema de gestion citas médicas, el patrón Memento es útil en situaciones donde necesitas mantener un historial de estados de un objeto para poder revertir a un estado anterior. En este caso, se utiliza para permitir al usuario guardar y restaurar el estado de una cita médica, lo que puede ser especialmente útil en aplicaciones donde los usuarios pueden cometer errores al ingresar datos y necesitan una forma de deshacer cambios.

6. DIAGRAMA UML



7.

8. CÓDIGO DE IMPLEMENTACIÓN

Clase AgendarCita

```

public class AgendarCita extends javax.swing.JFrame {
    private Connection conn;
    private Usuario user;
    private int idMedico;
    private String fecha;
    private String hora;
    private String nomDoc;
    private String tipo1;
    private String consultorio;

```

```

private Memento estadoGuardado; // Para almacenar el estado guardado

public AgendarCita(Usuario user, Connection conn) {
    initComponents();
    this.user = user;
    doctor.setEnabled(false);
    Fecha.setEnabled(false);
    Hora.setEnabled(false);
    this.conn=conn;
    Formato_Citas.obtenerDatosSucursales(conn,Sucursal);
}

/**
 * Método para guardar el estado actual en un Memento.
 * @return Un objeto Memento que contiene el estado actual.
 */
public Memento guardarEstado() {
    return new Memento(fecha, hora, nomDoc, tipol,
consultorio, idMedico);
}

/**
 * Método para restaurar el estado desde un Memento.
 * @param memento El objeto Memento que contiene el estado a
restaurar.
 */
public void restaurarEstado(Memento memento) {
    this.fecha = memento.getFecha();
    this.hora = memento.getHora();
    this.nomDoc = memento.getNomDoc();
    this.tipol = memento.getTipol();
    this.consultorio = memento.getConsultorio();
    this.idMedico = memento.getIdMedico();

    // Actualizar la interfaz gráfica con los valores
restaurados
    Fecha.setSelectedItem(fecha);
    Hora.setSelectedItem(hora);
    doctor.setSelectedItem(nomDoc);
    tipo.setSelectedItem(tipol);
    Sucursal.setSelectedItem(consultorio);

```

```

    }
    /**
     * Método para guardar el estado actual.
     */
    private void guardarEstadoActual() {
        estadoGuardado = guardarEstado();
        System.out.println("Estado de la cita guardado.");
    } private void restaurarEstadoGuardado() {
        if (estadoGuardado != null) {
            restaurarEstado(estadoGuardado);
            System.out.println("Estado de la cita restaurado.");
        } else {
            System.out.println("No hay estado guardado para restaurar.");
        }
    }
}

```

Clase Memento

```

public class Memento {
    private final String fecha;
    private final String hora;
    private final String nomDoc;
    private final String tipo1;
    private final String consultorio;
    private final int idMedico;

    /**
     * Constructor de la clase Memento.
     * @param fecha La fecha de la cita.
     * @param hora La hora de la cita.
     * @param nomDoc El nombre del doctor.
     * @param tipo1 El tipo de cita.
     * @param consultorio El consultorio.
     * @param idMedico El ID del médico.
     */
    public Memento(String fecha, String hora, String nomDoc, String tipo1,

```

```
String consultorio, int idMedico) {  
    this.fecha = fecha;  
    this.hora = hora;  
    this.nomDoc = nomDoc;  
    this.tipo1 = tipo1;  
    this.consultorio = consultorio;  
    this.idMedico = idMedico;  
}
```

// Getters para acceder a los atributos del estado guardado

```
public String getFecha() {  
    return fecha;  
}
```

```
public String getHora() {  
    return hora;  
}
```

```
public String getNomDoc() {  
    return nomDoc;  
}
```

```
public String getTipo1() {  
    return tipo1;  
}
```

```
public String getConsultorio() {  
    return consultorio;  
}
```

```
public int getIdMedico() {  
    return idMedico;  
}
```

```

    }


    } Caretaker(botón guardar y restaurar estado)
    private void guardarestadoActionPerformed(java.awt.event.ActionEvent evt)
    {
        guardarEstadoActual(); // TODO add your handling code here:
    }

    private void deshacerActionPerformed(java.awt.event.ActionEvent evt) {
        restaurarEstadoGuardado(); // TODO add your handling code here:}

```

9. SÁLIDA DEL CÓDIGO

Estado de la cita guardado.

 Output

```

run:
ROAA731111MOSRYRV2
Oaxaca
No hay estado guardado para restaurar.

```

Restaurando estado: 2024-06-04, 07:00:00, Dr. Roberto S nchez, Dentista, Odontologia 1

10. CONCLUSIÓN

concluimos que la implementación del patr n Memento en nuestro sistema de citas m dicas ha sido una decisi n acertada y beneficiosa. Este patr n nos permiti  manejar de manera eficiente el historial de estados de las citas sin comprometer la encapsulaci n de la l gica interna de los objetos. Gracias a esto, pudimos brindar a los usuarios la capacidad de guardar y restaurar estados de las citas m dicas, lo cual es especialmente  til para revertir cambios err neos o no deseados de forma sencilla.