

Documento sobre os padrões utilizados.

- Command

```
public interface Command {  
    public abstract Object execute(UsuariosDAO usuarios) throws Exception;  
    public abstract String getNome();  
}
```

```
public class ControlCommand implements Command {  
    private Usuario antigo, novo;  
    private String tipo;  
  
    public ControlCommand(Usuario antigo, Usuario novo, String tipo) {  
        this.antigo = antigo;  
        this.novo = novo;  
        this.tipo = tipo;  
    }  
  
    @Override  
    public Object execute(UsuariosDAO usuarios) throws Exception {  
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  
    }  
  
    @Override  
    public String getNome() {  
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  
    }  
}
```

- Memento

```
public class Memento {  
    private Command mem = null;  
  
    public void setEstado(Command est) {  
        mem = est;  
    }  
  
    public Command getEstado() {  
        return mem;  
    }  
}
```

```

public class MementoCareTaker {

    protected ArrayList<Memento> estados;

    public MementoCareTaker() {
        estados = new ArrayList<>();
    }

    public void adicionarMemento(Memento memento) {
        estados.add(memento);
    }

    public Memento getUltimoEstadoSalvo() {
        if (estados.size() <= 0) {
            return new Memento();
        }
        Memento estadoSalvo = estados.get(estados.size() - 1);
        estados.remove(estados.size() - 1);
        return estadoSalvo;
    }

}

```

- Memento/Fachada

```

public class Fachada {

    private final UsuariosDAO usuarios;
    private final MementoCareTaker mementoCareTaker;
    private Memento mem;

    //
    public Fachada(UsuariosDAO usuarios) throws errorMessageException {
        this.usuarios = usuarios;
        mementoCareTaker = new MementoCareTaker();
    }

    public Object service(Command c) throws Exception {
        Object o = c.execute(usuarios);
        mem.setEstado(c);
        mementoCareTaker.adicionarMemento(mem);
        return o;
    }

    public void undo(Memento mem) throws Exception {
        Command c = mementoCareTaker.getUltimoEstadoSalvo().getEstado();
        if (!(c == null)) {
            c.execute(usuarios);
            mem.setEstado(null);
        } else {
            throw new errorMessageException("Nenhuma operação para desfazer!");
        }
    }

}

```