

Documento Técnico: Análise dos Design Patterns no Projeto "Atividade Clínica"

Alunos: Gabrieli Bacca, Guilherme Guimarães, Luis Eduardo.

1. Introdução

Este documento tem como objetivo analisar e justificar a aplicação de quatro Design Patterns (Padrões de Projeto) essenciais no desenvolvimento do projeto "Atividade Clínica": Singleton, Observer, State e Builder. A utilização desses padrões visa promover a manutenibilidade, a flexibilidade e a clareza do código, separando responsabilidades e encapsulando comportamentos.

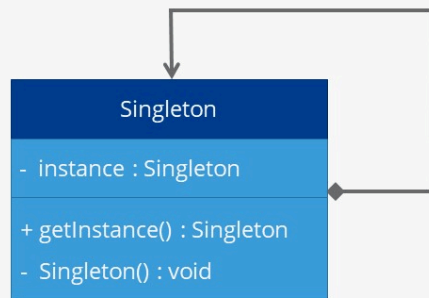
2. Padrões de Criação (Creational Patterns)

2.1. Singleton

Padrão	Aplicação no Projeto	Justificativa
Singleton	Implementado na classe FalsoBanco.	Garante que apenas uma única instância do "banco de dados" falso exista durante toda a execução da aplicação. Isso é crucial para simular um repositório de dados centralizado e evitar inconsistências que surgiriam se múltiplas instâncias do banco pudessem ser criadas.

O padrão Singleton restringe a instanciação de uma classe a um único objeto. A figura abaixo ilustra o conceito central do padrão, onde o acesso à instância é feito através de um método estático (getInstance), e o construtor é privado.

UML diagram: Singleton Pattern

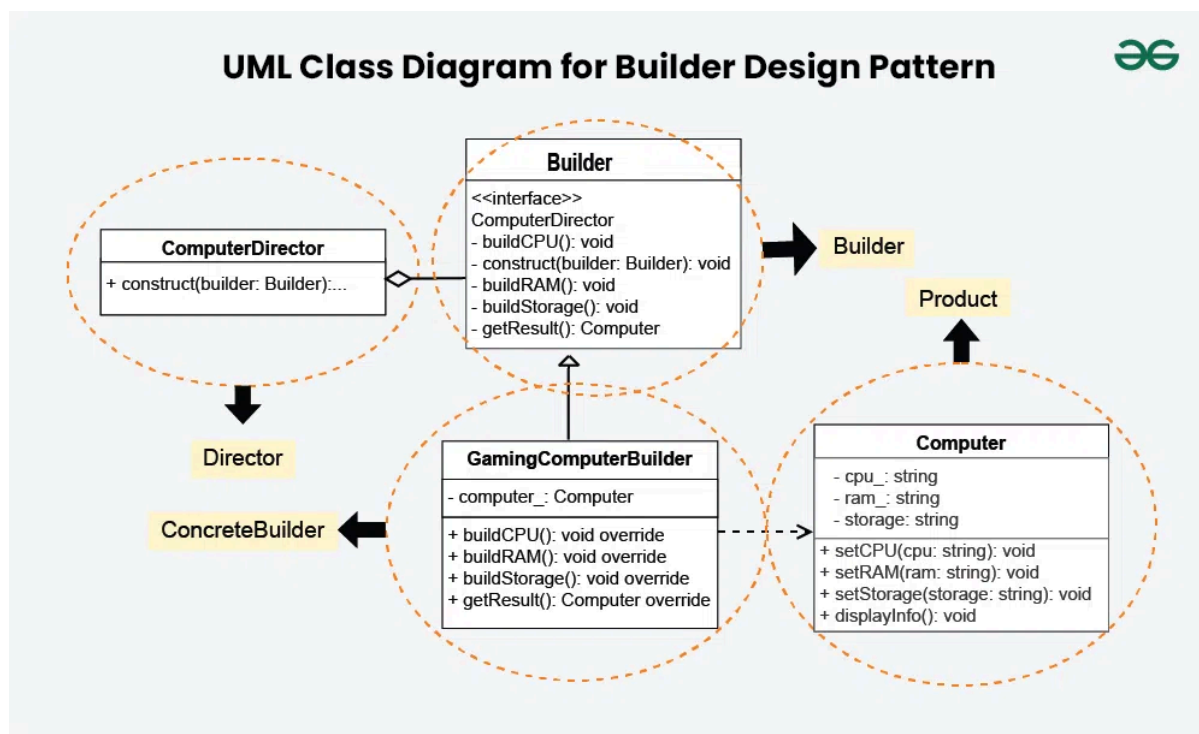


IONOS

2.2. Builder

Padrão	Aplicação no Projeto	Justificativa
Builder	Utilizado em todas as classes modelo (Paciente, Medico, Exame, etc.).	Permite a criação de objetos complexos passo a passo, sem a necessidade de múltiplos construtores com longas listas de parâmetros. No projeto, o Builder facilita a criação de objetos como Paciente e Medico, permitindo a definição opcional de atributos (como peso e altura) de forma legível e fluente.

O Builder separa a construção de um objeto de sua representação, permitindo que o mesmo processo de construção crie diferentes representações.

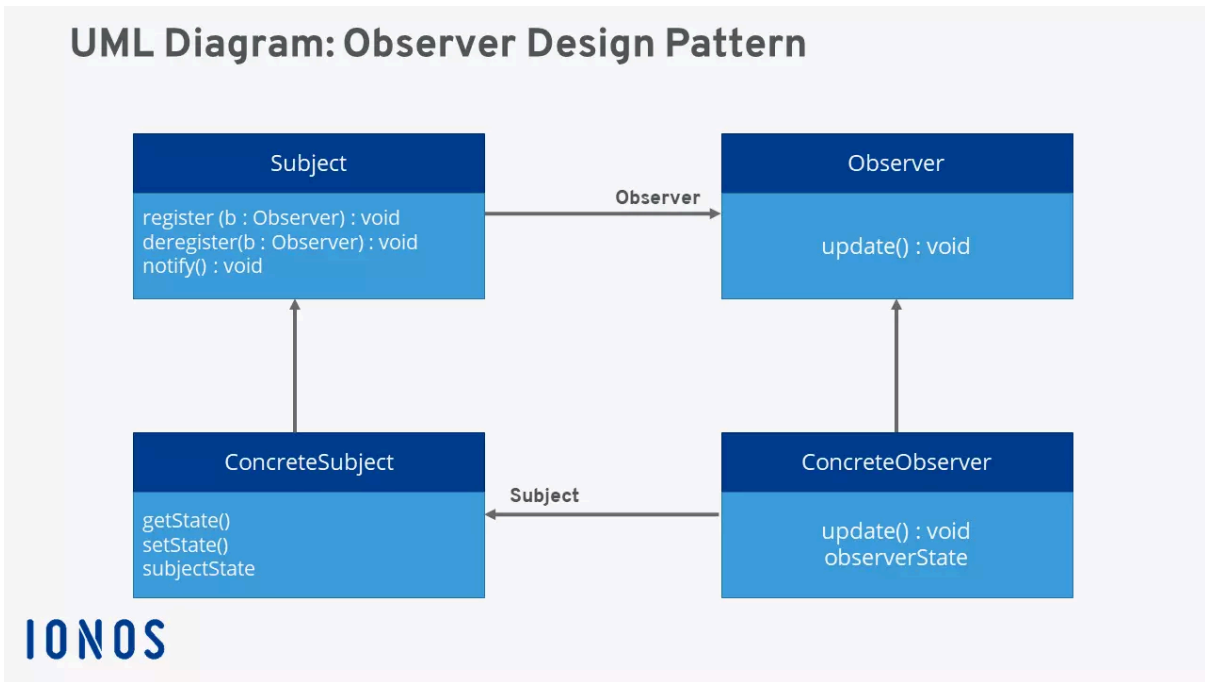


3. Padrões Comportamentais (Behavioral Patterns)

3.1. Observer

Padrão	Aplicação no Projeto	Justificativa
Observer	Utilizado para notificar médicos quando um exame de sua especialidade requer um profissional.	Define uma dependência um-para-muitos entre objetos. O Exame (ou a Especialidade associada) atua como o Subject (Sujeito), e os Medicos atuam como Observers (Observadores). Quando o estado do exame muda (ex: entra em EsperandoMedico), todos os médicos interessados são notificados, desacoplando a lógica de notificação da lógica de negócio principal.

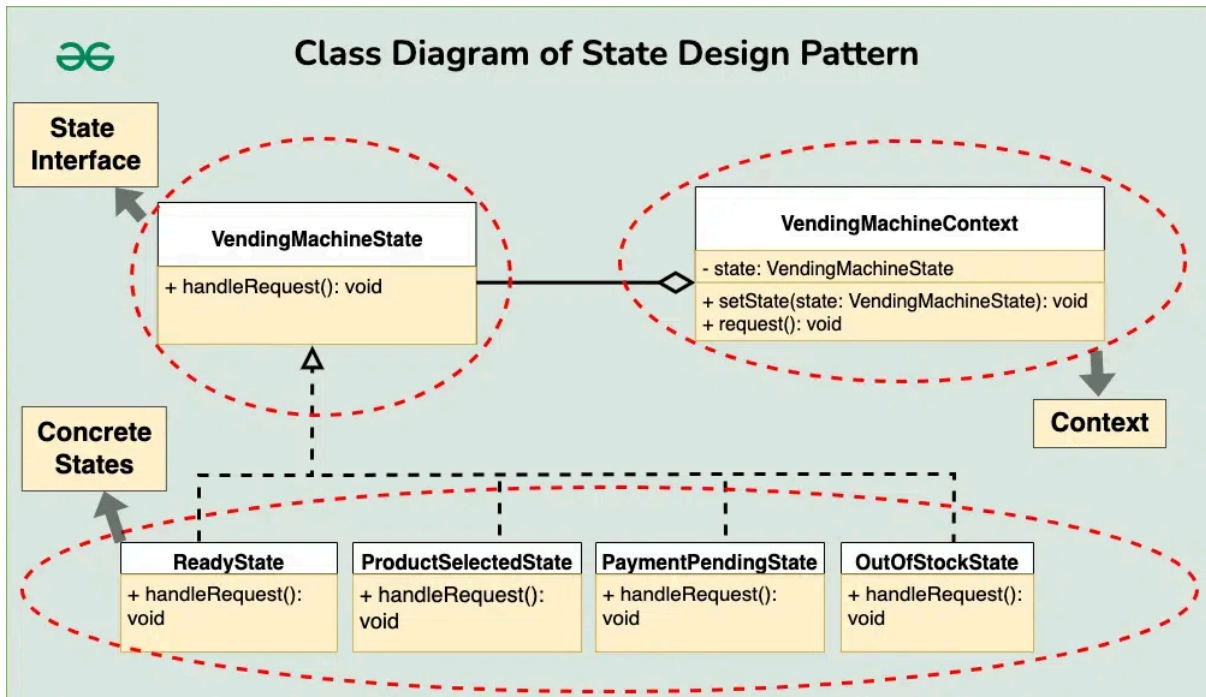
O padrão Observer é ideal para sistemas onde uma mudança de estado em um objeto deve ser comunicada a outros objetos sem que o objeto principal saiba quem são seus dependentes.



3.2. State

Padrão	Aplicação no Projeto	Justificativa
State	Utilizado para gerenciar a transição de estados do Exame (e.g., Inicio, EsperandoMedico, Agendado, Concluido, Cancelado).	Permite que um objeto mude seu comportamento quando seu estado interno muda. O objeto Exame muda seu comportamento (quais ações são permitidas, como agendar ou concluir) ao mudar seu estado. Isso evita grandes estruturas condicionais (if/else ou switch) e torna as transições de estado explícitas e fáceis de adicionar ou modificar.

O padrão State move a lógica de transição de estado para classes separadas que representam cada estado, tornando o código mais limpo e modular.



4. Conclusão

A aplicação dos padrões Singleton, Builder, Observer e State no projeto "Atividade Clínica" demonstra uma arquitetura de software robusta e bem pensada. Eles garantem que as responsabilidades sejam claramente separadas, que a criação de objetos seja controlada e que o comportamento do sistema seja flexível e fácil de estender, especialmente na gestão do ciclo de vida complexo do Exame.