

Tema: FERRAMENTA DE MODELAGEM DE SOFTWARE PARA ENSINO DE PROGRAMAÇÃO COM DESIGN PATTERNS

Introdução

Em desenvolvimento de software é comum a utilização de Design Patterns durante as fases de programação e refatoração. Alguns desafios associados ao aprendizado e ensino desses conceitos, são a elevada abstração e dificuldade na escolha do padrão adequado para um contexto específico.¹ Uma maneira de melhorar este problema é pela introdução gradual aos estudantes de exemplos e exercícios práticos, desde aplicações mais simples até as mais complexas.²

Objetivo: Desenvolver uma ferramenta que utiliza diagramas UML para gerar de códigos de design patterns, focando em projetos educativos e de pequeno porte para sistemas web e aproximando o estudo teórico da prática.

Metodologia

A base teórica do projeto é a metodologia da Ciência do Design. Esta abordagem é frequentemente adotada em disciplinas onde a ação prática e a criação de soluções concretas são centrais para o processo de pesquisa como projetos de engenharia, design e ciência da computação.

Arquitetura

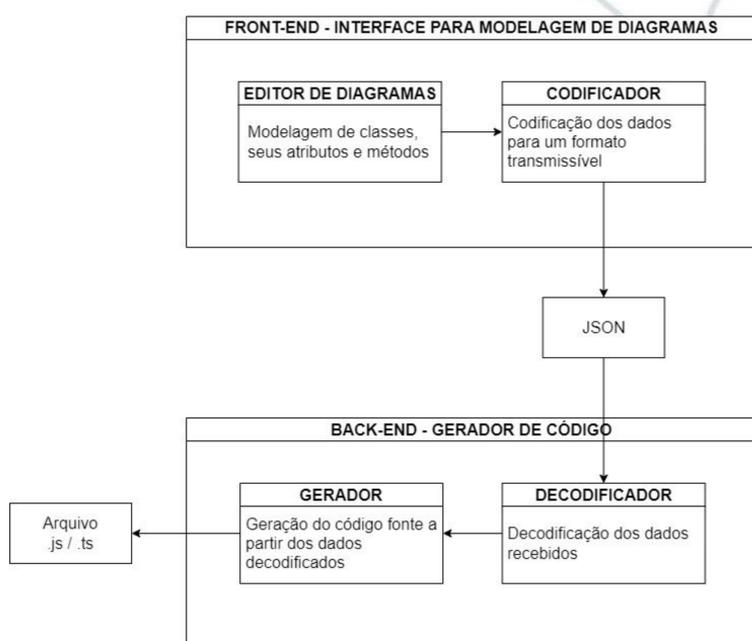


Figura 1 - Arquitetura cliente-servidor do sistema

A arquitetura elaborada para o projeto segue como base o modelo cliente-servidor. O cliente consiste de um editor de diagramas que possibilita a modelagem em

UML e que codifica a informação em JSON para transmissão via Web. Recebendo o dado, o servidor o decodifica para gerar os arquivos de códigos a partir de um gerador de código fonte.

Resultados

O software foi desenvolvido com sucesso e abrange os principais requisitos levantados, incluindo: modelagem de diagramas UML, associação a conceitos de design patterns e geração de templates de código.

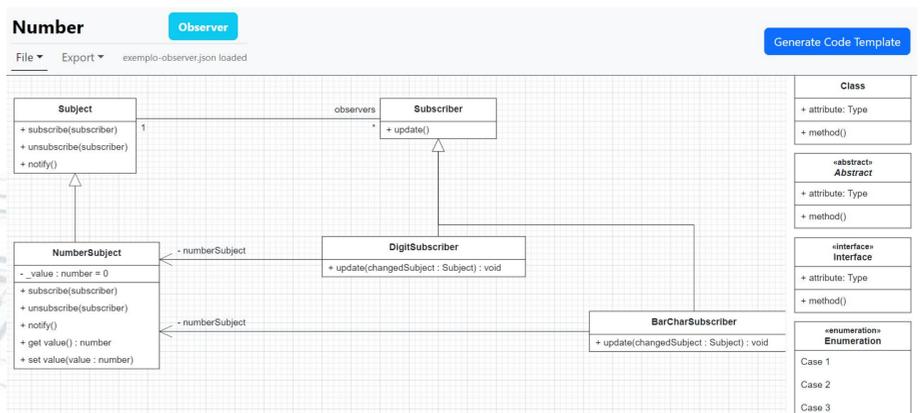


Figura 2 - Tela desenvolvida para o editor de diagramas

```

export class Subject {
  //region Attributes
  //endregion

  constructor() {
  }

  //region Methods
  public subscribe(subscriber){
    # Implement This Method
  };
  public unsubscribe(subscriber){
    # Implement This Method
  };
  public notify(){
    # Implement This Method
  };
  //endregion
}

export class DigitSubscriber {
  //region Attributes
  //endregion

  constructor() {
  }

  //region Methods
  public update(changedSubject : Subject) : void{
    # Implement This Method
  };
  //endregion
}

export class Subscriber {
  //region Attributes
  //endregion

  constructor() {
  }

  //region Methods
  public update(){
    # Implement This Method
  };
  //endregion
}
    
```

Figura 3 - Exemplos de códigos gerados a partir do diagrama

Conclusões

Como prova de conceito, foi desenvolvido uma pequena aplicação utilizando os códigos gerados pelo sistema através do padrão Strategy, consistindo em um jogo de pedra, papel e tesoura com duas possíveis estratégias. Com isso, pode-se validar o uso desta ferramenta para implementar projetos de software simples a partir dos modelos elaborados com design patterns. Assim, o usuário poderá relacionar mais facilmente a motivação e o uso desses padrões com a implementação prática.

Referências

[1] HUANG, H.; YANG, D. Teaching design patterns: A modified pbl approach. IEEE, 2008. Disponível em: <<https://ieeexplore.ieee.org/document/4709353>>.
 [2] PILLAY, N. Teaching design patterns. Universidade de KwaZulu-Natal, 2010.

Integrantes: André Hideki Gashu Nishimura
 William Abe Fukushima

Professor(a) Orientador(a): Prof. Dr. Jorge Luis Risco