**NOME DA IES**

**CAMPUS**

**Calculadora em Java**

**Heitor Vilaça Santana**

**Paulo Epifanio**

**2024**

**moreno/pernanbuco**

* DIAGNÓSTICO E TEORIZAÇÃO
* Identificação das partes interessadas e parceiros

As partes interessadas são alunos de cursos de tecnologia ou áreas afins, de ensino médio ou superior, com interesse em aprender programação orientada a objetos (POO) e desenvolvimento de interfaces gráficas em Java. O perfil socioeconômico é variado, e a faixa etária média é de 16 a 25 anos.

* Problemática e/ou problemas identificados

tive dificuldades com o efetuamento do projeto, por estar sozinho, tive que resolver problemas de maneira individual, erros de codigo juntamente com falta de prática e conhecimento, foi um trabalho complicado que teve um fim satisfatório

* Justificativa

A principal razão para isso está relacionada à carga horária limitada das aulas. Uma disciplina dessa natureza deveria contar com uma carga horária mais extensa, além de uma cobrança mais rigorosa em relação ao envolvimento dos alunos. Muitas vezes, os estudantes formam grupos com o intuito de se apoiar nos colegas, sem contribuir de maneira efetiva para o desenvolvimento das atividades.

* Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos)

Demonstrar os conceitos fundamentais de POO em Java, como encapsulamento, herança e polimorfismo.

Criar uma interface gráfica simples e funcional usando Swing.

Fornecer um projeto prático que facilite o aprendizado de programação orientada a objetos.

* Referencial teórico (subsídio teórico para propositura de ações da extensão)

Grady Booch - Em seu livro "Object-Oriented Analysis and Design with Applications" (2007), Booch explora os pilares da POO, como encapsulamento, herança e polimorfismo. A aplicação desses conceitos de forma estruturada facilita a compreensão de como organizar código, o que justifica sua inclusão no projeto. A obra serve como base para ensinar como esses conceitos podem ser usados para melhorar a modularidade e reusabilidade no desenvolvimento de software.

Robert Eckstein - Autor de "Swing" (2000), Eckstein é uma referência importante para a compreensão do uso de bibliotecas gráficas em Java, especificamente a Swing. Seu trabalho fornece as bases para a criação de interfaces gráficas eficientes e interativas. A utilização do Swing no desenvolvimento da calculadora de Java visa demonstrar como um aspecto fundamental do design de software, como a interface gráfica, pode ser construído e melhorado, de forma a gerar uma experiência mais rica para o usuário.

Barbara Liskov - A Liskov Substitution Principle, proposta por Barbara Liskov em 1987, é um princípio essencial da POO que sugere que os objetos de uma classe devem poder ser substituídos por objetos de uma subclasse sem alterar as propriedades desejáveis do programa. Este conceito, além de se alinhar com o projeto em questão, é diretamente aplicável ao uso de herança e polimorfismo para que operações matemáticas, por exemplo, possam ser implementadas de forma genérica e, ao mesmo tempo, específica nas subclasses de operação.

* PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO
* Plano de trabalho (usando ferramenta acordada com o docente)

O plano inclui a criação de uma base teórica sobre POO, desenvolvimento de classes de cálculo, criação da interface gráfica, e testes práticos.

* Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los.

não houve pois o projeto foi efetuado de maneira individual.

* Grupo de trabalho (descrição da responsabilidade de cada membro)

O aluno Heitor ficou responsável por tudo.

* Metas, critérios ou indicadores de avaliação do projeto

Avaliação de entendimento dos conceitos de POO e habilidades em desenvolvimento gráfico, observando a clareza no código e a funcionalidade da calculadora.

* Recursos previstos

Computadores com ambiente de desenvolvimento Java, software de edição de código (VSCode/NetBeans), e material de apoio didático.

* Detalhamento técnico do projeto

Uso de classes para operações matemáticas, uma estrutura de herança para diferentes operações e encapsulamento de dados.

* ENCERRAMENTO DO PROJETO
* Relato Coletivo:

Trabalho efetuado individualmente .

* Avaliação de reação da parte interessada

Realizar uma pesquisa de opinião com os alunos para avaliar a eficácia do projeto como ferramenta de ensino.

* Relato de Experiência Individual (Pontuação específica para o relato individual)

Heitor:A experiência foi um bom aprendizado, mas também uma lição de paciência e perseverança. A teoria parecia simples, mas na prática, as dificuldades encontradas, principalmente na parte gráfica, foram frustrantes. Para futuros projetos, eu recomendaria mais atenção ao planejamento da interface e uma melhor organização do código desde o início.

* CONTEXTUALIZAÇÃO
* O projeto em questão foi voltado para o desenvolvimento de uma calculadora simples utilizando a Programação Orientada a Objetos (POO) em Java. O objetivo principal foi aplicar conceitos de POO — como encapsulamento, herança e polimorfismo — no desenvolvimento de software, ao mesmo tempo em que se praticava a criação de interfaces gráficas utilizando a biblioteca Swing,ambiente de desenvolvimento como o VS Code, além de criar uma interface gráfica interativa para a aplicação. A proposta envolveu o uso de uma metodologia ativa, onde a prática de programação foi central para a construção da solução.
* METODOLOGIA

A metodologia foi voltada para a realização do projeto de forma individual, em que todas as etapas foram realizadas no meu ambiente de desenvolvimento (VS Code), com foco na implementação prática e constante aprimoramento do código. As etapas do processo foram as seguintes:

Planejamento da Lógica de Programação: Definição das operações e estruturação do código utilizando os conceitos de POO, como a criação de uma classe base e subclasses para diferentes operações matemáticas.

Desenvolvimento da Interface Gráfica: Implementação da interface usando a biblioteca Swing para permitir a interação do usuário com os cálculos.

Integração da Lógica e Interface: Ajuste entre a lógica de cálculos e a interface gráfica para garantir o correto funcionamento do programa.

Testes e Ajustes: Realização de testes para garantir que o programa funcionava como esperado e ajustes baseados nos erros encontrados.

Apesar de ser um trabalho individual, não houve interação direta com outros membros ou feedback constante, o que gerou um processo de aprendizado mais lento e com algumas dificuldades, especialmente na fase de integração entre a lógica e a interface gráfica.

* RESULTADOS E DISCUSSÃO:

A expectativa inicial era de que o projeto fosse uma oportunidade de aplicar os conceitos de POO de forma prática e aprender a usar uma biblioteca gráfica simples, como o Swing. O que foi vivenciado foi um processo de aprendizado mais complexo do que o esperado. A construção da interface gráfica, em particular, foi desafiadora, pois exigia uma comunicação eficiente entre a lógica e os elementos visuais da aplicação. A experiência foi repleta de tentativas e erros, com várias revisões no código até que a interação entre os componentes fosse bem-sucedida.

O resultado final foi uma calculadora funcional, mas a construção da interface levou mais tempo do que o planejado. As dificuldades foram principalmente no aspecto visual, tentando criar uma interface agradável e funcional, ao mesmo tempo em que integrava as operações de cálculo.

Eu me senti, ao mesmo tempo, desafiado e realizado ao ver o progresso do projeto. As dificuldades me ensinaram a importância de uma boa organização e planejamento, além de reforçarem a necessidade de se aprofundar nas ferramentas utilizadas. Uma recomendação seria focar mais em um planejamento detalhado da interface gráfica antes de começar a codificação, para evitar retrabalho.

* REFLEXÃO APROFUNDADA

A experiência prática me permitiu compreender melhor como a teoria de POO pode ser aplicada em projetos reais. O uso de herança e polimorfismo foi eficiente, pois permitiu a extensão do código sem modificações substanciais nas classes principais. No entanto, a integração entre a interface gráfica e a lógica de programação demonstrou a importância de se ter um planejamento mais claro, como a separação das responsabilidades do modelo, da visualização e da interação com o usuário.

O conceito de encapsulamento foi útil para proteger dados dentro das classes e evitar modificações externas indevidas, mas a integração da interface com a lógica revelou um desafio maior do que inicialmente previsto. A teoria de modularização se refletiu na prática ao conseguir dividir o código em diferentes partes responsáveis, como a classe de operações e a interface, o que facilitou a manutenção e expansão da funcionalidade do projeto.

* CONSIDERAÇÕES FINAIS

Embora o projeto tenha sido realizado de forma individual, o aprendizado adquirido foi significativo. A experiência proporcionou uma compreensão mais profunda de como os conceitos de POO são aplicados na prática, especialmente no contexto de desenvolvimento de interfaces gráficas. As dificuldades enfrentadas, como a comunicação entre a lógica e a interface, podem ser mitigadas no futuro com um melhor planejamento.

Para trabalhos futuros, sugiro que a interface gráfica seja trabalhada de maneira mais planejada, talvez até utilizando outras ferramentas como JavaFX, que oferecem recursos mais avançados para a criação de interfaces visuais mais interativas e modernas. Além disso, implementar testes automatizados poderia tornar o desenvolvimento mais eficiente e aumentar a confiabilidade do software.

Em termos de pesquisa, a utilização de padrões de projeto, como MVC (Model-View-Controller), poderia ser uma alternativa para organizar melhor a estrutura do código, facilitando a manutenção e escalabilidade da aplicação. Isso também abriria portas para o estudo de boas práticas no desenvolvimento de sistemas mais complexos, aproveitando ao máximo os conceitos de POO.

**OBSERVAÇÃO: Exige-se que todo o processo de desenvolvimento do projeto de extensão seja documentado e registrado através de evidências fotográficas ou por vídeos, tendo em vista que o conjunto de evidências não apenas irá compor a comprovação da realização das atividades, para fins regulatórios, como também poderão ser usadas para exposição do projeto em mostras acadêmico-científicas e seminários de extensão a serem realizados pelas IES.**