Aluno: Eduardo Tormena Cavazin;

Em qual parte do projeto trabalhou? Qual relevância e aprendizado?

Em qual parte do projeto trabalhei?

- Fui responsável pela **seção do Paradigma Funcional** no trabalho de comparação entre imperativo, OO, funcional e lógico.
    - Elaborei o **embasamento teórico**, com histórico (λ-cálculo, Lisp, ML, Haskell) e princípios-chave (imutabilidade, funções puras, lazy evaluation, HOFs).
    - Desenvolvi o **exemplo prático** em Haskell (lista infinita, `filter` + `take`, parser de calculadora funcional) para ilustrar como se escreve código sem efeitos colaterais.
    - Preparei a **análise comparativa**, evidenciando diferenças de modularidade, testabilidade e tratamento de estado frente aos demais paradigmas.

Qual a relevância e o que aprendi?

- **Relevância dessa parte**
    1. Demonstra na prática como o funcional **modulariza** lógica via composição de funções puras, sem misturar I/O.
    2. Serve de **contraponto** às seções imperativa (estado mutável) e OO (objetos e encapsulamento), enriquecendo a discussão sobre trade-offs de design.
- **Aprendizados principais**
    1. **Separação total de I/O e lógica**: toda a computação pura ficou isolada, o que facilita testes unitários e property-based.
    2. **Uso de tipos opcionais (`Maybe`)** para tratamento de erro sem exceções, reforçando a robustez do fluxo de dados.

3. **Lazy evaluation** e listas infinitas como ferramentas poderosas para gerar e processar dados sob demanda.

4. **Composição de funções de ordem superior** (`map`, `filter`, `fold`) para criar pipelines claros e declarativos.

5. **Comparação estruturada**: ao ver o código funcional lado a lado com o OO e o imperativo, ficou evidente como cada paradigma lida com estado, reutilização e paralelismo.

Para cada apresentação descreva:
Houve relevância e/ou algo interessante? cite e justifique.

Apresentação Grupo “Enois”:
Análise de paradigmas de programação em módulos do VS Code;

Sim, o projeto teve relevância pois o mesmo exemplifica que não se faz necessário se manter em apenas um paradigma para executar um projeto, ainda mais sendo o projeto de IDE complexa e multi-funcional como o VS Code, portanto, aprendemos que é importante estudar e pesquisar sobre outros paradigmas de programação, sempre se mantendo atualizado com novas técnicas e conhecimento.

Apresentação do grupo “The Rapazes”:
Do paradigma ao resultado: Como diferentes abordagens resolvem os mesmos problemas de formas distintas

A equipe apresenta sobre diferentes paradigmas e mostra a refatoração de código em um projeto de um dos membros, apresentando a diferença diferentes soluções, analisando a complexidade de fazer o mesmo e as diferenças na leitura e complexidade do código. A relevância se dá em poder apresentar qual o custo temporal que se faz necessário para refatoração de código para um diferente paradigma.

Apresentação do grupo “Los Bandoleiros”:
Comparação entre paradigmas: Orientado a Objetos e Imperativo

A equipe apresenta códigos escritos em ambos paradigmas, fazendo uma comparação tanto em seu aspecto visual para leitura, quanto na complexidade do

mesmo. Apontam as principais diferenças entre ambos paradigmas e apresenta a complexidade dos mesmos.

Apresentação do grupo “YET”
Estudo sobre paradigma de programação Funcional

A equipe realizou um estudo aprofundado no Paradigma Funcional, assim como na linguagem Haskell, apresentando sua história e conceitos fundamentais. Realizaram também uma com o paradigma imperativo, apresentando suas diferenças e as vantagens do Paradigma Funcional contra o Paradigma Imperativo, também foi apresentado códigos em ambos paradigmas.

Foi também apresentado exemplos de linguagens de programação que conseguem aplicar o paradigma, separadas em linguagens puramente funcionais, fortemente funcionais e parcialmente funcionais.

O paradigma funcional precisa de situações mais específicas. Amplamente aplicada em áreas que exigem robustez, testabilidade e manutenibilidade. Frameworks como React.Js e Reduz utilizam imutabilidade e funçẽos puras. Linguagens como Haskell na manipulação de dados em tarefas e de machine learning e nlp. Linguagens funcionais são aplicadas em roteamento de pacotes e sistemas tolerantes a falhas na área de telecomunicações.

Apresentação do grupo “CKS”:
Estudo do Paradigma de Programação Orientado a Objetos

O grupo traz a base matemática do paradigma, que é a função declarativa. A linguagem utilizada foi o Java, uma linguagem parcialmente funcional. Pode ser usada através de expressões lambda, stream: map, filter, Interfaces funções como Function<T,R>.
O grupo também traz as vantagens do paradigma como o fato de que ela assegura que as respostas das funçẽos do programa sempre irão responder de acordo com seus parâmetros. Também trouxeram desvantagens, como a curva de aprendizado acentuada, especialmente para programadores acostumados com o paradigma imperativo/tradicional. Diferenças significativas na forma de resolver problemas, exigindo mudança de mentalidade. Ecossistema java predominantemente imperativo e orientado a objetos, dificultando a integração natural do paradigma funcional.