

**Aluno:** Vinícius Margonar;

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

No projeto, trabalhei na pesquisa e análise do paradigma lógico, com foco na linguagem Prolog. Estudei sua origem histórica, princípios fundamentais, forças e limitações, além de identificar cenários de aplicação e casos reais de uso. Também desenvolvi exemplos de código explicativos, como uma calculadora simples e consultas baseadas em regras lógicas.

Esse trabalho foi importante para aprofundar meu entendimento sobre programação declarativa, raciocínio automático, e modelagem lógica de conhecimento. Aprendi como o paradigma lógico se diferencia dos paradigmas imperativos e orientado a objetos, e como ele é poderoso para representar e inferir conhecimento em áreas como inteligência artificial, sistemas especialistas e PLN.

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

**1. Apresentação do grupo “Enois”**

Análise de paradigmas de programação em módulos do VS Code

A apresentação demonstrou grande relevância ao mostrar que não é necessário restringir-se a apenas um paradigma de programação para desenvolver projetos complexos, como no caso do VS Code. A análise evidenciou que diferentes paradigmas podem coexistir e se complementar em ambientes multifuncionais. Isso reforça a importância de explorar novas abordagens e manter-se atualizado com diferentes estilos de programação, ampliando as possibilidades de solução e inovação.

**2. Apresentação do grupo “The Rapazes”**

Do paradigma ao resultado: Como diferentes abordagens resolvem os mesmos problemas de formas distintas

O grupo apresentou um conteúdo bastante interessante ao comparar diferentes paradigmas aplicados a um mesmo problema prático, utilizando um projeto real como base. Refatoraram o código original para outras abordagens e destacaram as diferenças em termos de legibilidade e estrutura. O ponto mais relevante foi a discussão sobre o esforço necessário para migrar de paradigma, mostrando que a mudança tem impacto direto na complexidade e no tempo de desenvolvimento.

**3. Apresentação do grupo “Los Bandoleiros”**

Comparação entre paradigmas: Orientado a Objetos e Imperativo

A equipe realizou uma comparação direta entre os paradigmas imperativo e orientado a objetos, utilizando exemplos de código para demonstrar as distinções

práticas entre os dois. A apresentação foi útil para entender as diferenças na organização e clareza dos códigos, além das implicações que cada paradigma traz para a resolução de problemas. A análise também destacou a complexidade envolvida na adaptação entre os modelos.

#### **4. Apresentação do grupo “YET”**

Estudo sobre paradigma de programação funcional

O grupo apresentou um estudo aprofundado sobre o Paradigma Funcional, destacando a linguagem Haskell, seus conceitos, características e comparando-o ao paradigma imperativo por meio de exemplos práticos de código. Também classificaram as linguagens funcionais em puramente, fortemente e parcialmente funcionais, mostrando sua aplicação em diferentes cenários. Ressaltaram que o paradigma funcional é indicado para contextos que exigem robustez, testabilidade e fácil manutenção, com aplicações em ferramentas como React.js e Redux, além de áreas como machine learning, NLP, roteamento de pacotes e sistemas tolerantes a falhas em telecomunicações.

#### **5. Apresentação do grupo “CKS”**

Estudo sobre paradigma de programação funcional (Java)

O grupo apresentou a base matemática do paradigma funcional, destacando sua natureza declarativa, e utilizou a linguagem Java como exemplo, por ser parcialmente funcional e permitir o uso de expressões lambda, operações com streams como map e filter, além de interfaces funcionais como `Function<T, R>`. Foram abordadas as vantagens do paradigma, como a previsibilidade das funções, que sempre retornam o mesmo resultado para os mesmos parâmetros. Também foram discutidas desvantagens, como a curva de aprendizado elevada para quem está habituado ao paradigma imperativo, as diferenças conceituais na resolução de problemas e a dificuldade de adoção do modelo funcional no ecossistema Java, que é majoritariamente imperativo e orientado a objetos.