

Autoavaliação

1. Cumprimento das regras

1.1. Relevância do Ciclo de Conhecimento

Vejo que os projetos, tanto no primeiro quanto no segundo ciclo, trazem relevância aos seus respectivos temas. Tanto a análise da aplicação de Inteligências Artificiais no refatoramento de código no 1º ciclo quanto a utilização dos módulos de terminal e versionamento de código do GitHub no 2º ciclo, apesar de seus resultados pouco satisfatórios, que serão discutidos na seção apropriada, trouxeram conteúdo relevante e alinhado com tanto com o tema quanto com os trabalhos das demais equipes.

Destaco que, para o cumprimento das regras, vejo o desempenho da equipe, tal como meu próprio, ao nível **Desenvolvedor**.

1.2. Fundamentação Profissional e Aplicabilidade

De certo, este foi um dos pontos objetivamente mais fracos que a equipe teve. Apesar de dizer que trouxemos conteúdo relevante para nossas apresentações em ambos ciclos, é inegável que a forma na qual este foi apresentado é insatisfatória. Nós trouxemos casos de uso reais, mas falhamos em executá-los por subestimar a dificuldade da tarefa – um erro de mensuração já reconhecido entre nosso grupo e que buscaremos melhorar nos próximos ciclos através de escopos mais realistas.

Para todos os efeitos, destaco que a aplicabilidade, ainda que teoricamente fundamentada, trouxe escopos impossíveis de entregar com os recursos disponíveis, e portanto, tanto eu quanto o restante da equipe desempenhamos ao nível **Estagiário**.

1.3. Originalidade, Inovação e Contribuição Técnica

Apesar do aprendizado adquirido durante o desenvolvimento de ambos ciclos, a originalidade é questionável no 1º ciclo, já que a tarefa de refatorar códigos antigos com a ajuda de Inteligências Artificiais não é mais tão revolucionária quanto costumava ser. Creio que, apesar da execução pouco satisfatória, o 2º ciclo trouxe uma ideia mais original, apresentando um estudo de caso interessante de módulos de um projeto de *software* aberto relevante para desenvolvedores: o Visual Studio

Code, e acredito que com melhor gerenciamento do tempo e uma limitação do escopo para um módulo em vez de dois, a equipe poderia ter empreendido melhor seus esforços, trazendo repertório teórico e prático com metodologias melhor fundamentadas e uma conclusão coerente.

Tendo isto em vista, vejo que o desempenho tanto da equipe quanto o meu próprio, neste quesito, estão no nível **Júnior**.

2. Resultados

Acredito que, em termos de conhecimento adquirido, ambos ciclos trouxeram bons resultados, tanto sobre os temas quanto sobre a gerência da equipe, que é uma configuração inédita. Contudo, é evidente que os resultados das apresentações não refletiram de forma suficiente os conhecimentos adquiridos, e acredito que isso se deve, novamente, pela mensuração incorreta que fizemos dos escopos escolhidos.

Diante disso, avalio que tanto eu quanto o grupo estamos no nível **Júnior**.

3. Dedicção

Minha dedicação em particular melhorou de um ciclo para o outro, devido a fatores externos do começo do ano que solucionei na 2ª metade do 1º trimestre, aos quais não vejo pertinência em detalhar. Creio que, com um esforço melhor alinhado com os demais membros da equipe, eu poderia ter feito muito mais, e uma falta de comunicação da minha parte certamente teve seu impacto nesse resultado negativo.

Diante disso, avalio que minha avaliação, no quesito dedicação, é do nível **Desenvolvedor**.

Relatórios do 2º Ciclo

1. É Nois

Participei da pesquisa sobre o módulo de terminal do Visual Studio Code no GitHub. Acredito que tanto o trabalho quanto a apresentação foram suficientemente bons, mas passíveis de melhorias. Creio que a ideia de abordar dois módulos, que veio de mim, não foi muito boa, pois subestimamos a complexidade da robustez dos módulos, e creio que teríamos um esforço melhor alocado com todos concentrados em um mesmo módulo. Eu gostei da metodologia que adotamos, embora os slides

pudessem ser melhor alinhados para ficar mais coerentes. No geral, acho que apresentamos exemplos práticos de como cada paradigma estava presente.

2. The rapazes

Eu gostei da apresentação deste grupo. Como sempre, os slides deles são muito bons em questão didática, o que contribui para a apresentação, que também foi muito boa. O conceito de código hidra é um que eu não tinha conhecimento, ao menos não com esse nome, mas a forma que utilizaram para explicá-lo foi muito boa e eu tive uma perspectiva nova sobre esse tipo de problema, que não são exatamente novos para mim, mas que eu não tinha nenhuma fundamentação teórica para explicar. Por fim, acho o experimento dos rapazes interessante, embora a explicação teórica tenha me chamado mais atenção que o projeto em si.

3. Los Bandoleiros

Acho que eles fizeram bem em trazer uma discussão mais teórica e comparativa acerca dos paradigmas imperativo e orientado a objetos. Certamente a comparação, baseada mais em teoria, pôde agregar aos trabalhos feitos anteriormente, já que os conceitos dos paradigmas estavam menos explicados neles para dar mais tempo aos seus respectivos estudos de caso. Acredito que a equipe utilizou muito de jargões durante a apresentação, mas nada que fosse um detrimento necessariamente.

4. CREV

Este grupo teve um foco mais teórico, assim como o anterior. Mas diferente do anterior, eles não buscam fazer uma comparação direta, ao menos a priori, mas sim trazer um apanhado histórico apresentando as origens de cada um dos 4 paradigmas, bem como as necessidades que eles deviam suprir. Por fim, eles trazem uma tabela apresentando a aplicação dos paradigmas ao fazer uma calculadora, utilizando linguagens de programação nas quais seus respectivos paradigmas representados predominam, apresentando como cada uma lida com a mesma tarefa, o que eu achei bastante interessante.

5. YET

O grupo focou em explicar o paradigma funcional em específico, trazendo referencial histórico e comparações diretas com o paradigma imperativo, apontando ambos como opostos um ao outro. Enquanto o paradigma funcional foca na imutabilidade das funções e na preservação dos dados nas rotinas por meio de

cópias, o paradigma imperativo foca na criação e manipulação direta dos dados, definindo um caminho preciso que deve ser seguido sem focar em pormenores que desviem dos caminhos previstos. Eles agregam nas apresentações anteriores ao expandir conceitos do paradigma funcional que foram citados em outras apresentações, além de mostrar em maiores detalhes a linguagem *Haskell*, da qual eu ouvi falar, mas não cheguei a pesquisar sobre como ela funciona.

6. CKS

A equipe foca em explicar paradigma funcional utilizando Java, o que é interessante considerando que Java, apesar de suportar vários paradigmas, ainda é uma linguagem mais conhecida pela programação orientada a objetos. É uma quebra de expectativa bastante interessante. Utilizando o Java 8 como base, eles apresentam, principalmente, métodos padrões que respeitam o paradigma funcional de imutabilidade, além do uso de padrões de projeto presentes na orientação a objetos que, combinados com a programação funcional, se tornam muito mais sucintos e, portanto, fáceis de prestar manutenção.