

## Atividades da Primeira Aula

**Aluno: João Pedro Rodrigues Leite**

**RA: a2487055**

### Responda as questões a seguir utilizando esta cor no seu texto

1. O que é um código MACARRONICO?

R: É aquele tipo de código confuso e feio, que não é possível entender facilmente o que está acontecendo

2. O que é um sistema LEGADO?

R: Sistema legado é aquele tipo de sistema que está em funcionamento e vai precisar de uma manutenção, se adequar as novas tecnologias com foco na aplicação das boas práticas

3. O que diz a REGRA do ESCOTEIRO?

R: “Uma vez que você sai da área que está acampando, você deve deixá-la mais limpa do que quando você a encontrou”. Na programação isso significa que quando visualizarmos um código que está ruim sempre devemos deixá-lo melhor do que antes

4. Qual a HABILIDADE ESSENCIAL e NECESSÁRIA para ser um bom programador?

R: Basicamente temos dois caminhos, um que talvez vai gerar progresso, mas desconforto garantido e um que com certeza não vai gerar progresso, mas garante conforto. Com isso para que seja torne um bom programador devemos sempre escolher o caminho que gera desconforto, dessa forma sempre podemos evoluir.

5. Quais são os quatro hábitos que tornam um programador ineficiente? Cite e explique cada um deles.

R: 1 – Falar sim pra tudo. Faz com que sua produtividade seja reduzida porque você será interrompido o tempo todo, outro ponto é que as pessoas que pedem uma opinião sobre alguma coisa, um e-mail perigoso de se enviar ou algo assim, se você der a sua opinião sobre o e-mail, se der certo vocês irão dividir a conquista, e se der errado vocês iriam dividir a culpa, é muito melhor apenas a pessoa ter 100% do retorno;

2 – A palavra “pronto” nem sempre é pronto de verdade. Muitas vezes um programador fala que o código está pronto, mas o código que ele fez está ruim ao ponto de que outro desenvolvedor não consegue entender, então ele precisaria refatorar, ou seja, o código não estava pronto, ou até mesmo o código que foi feito e não há nenhuma documentação, então o código também não está pronto;

3 – Não testar o próprio código. Todo programador deveria escrever teste automatizados a fim de garantir um código funcional. Pois testar apenas o caminho feliz de um código é tão besta quanto eu dizer que concordo com a minha própria opinião;

4 – Fazer commits gigantescos. Não é uma boa prática, o ideal é que o commit não seja transformado em um diário, mas sim em uma unidade de alteração funcional;

6. Em um algoritmo, o que é o CAMINHO FELIZ?

R: O caminho feliz é quando o programador só considera os casos ideais de execução de um algoritmo, sem levar em conta possíveis exceções ou problemas

7. Porque testar o CAMINHO FELIZ não é uma boa prática?

R: É importante testar além do caminho feliz para garantir que o código seja robusto o suficiente para lidar com diferentes cenários de uso