

# Princípios em Projeto de Software

## Atividades de Aprendizagem e Avaliação

Aluno: **Lucas Daniel Batista** RA: **2159171**

1. Considerando o conteúdo do texto do link “Princípios em Projeto de Software”

a) Comente a afirmação de John Ousterhout.

R: Dividir o problema em partes menores ajuda a criar soluções específicas e pontuais, que unidas resolverão o problema como um todo. Essas soluções menores podem ser testadas e validadas individualmente, fazendo com que sejam mais assertivas. Em outras palavras, pode ser entendido como o conceito de Separation of Concerns. A grande dificuldade é definir a melhor estratégia de divisão das tarefas da maneira que mais faça sentido e acabe não atrapalhando mais do que ajudando.

b) Comente o conceito de Integridade Conceitual.

R: Em resumo, pode-se descrever como a utilização de um padrão no visual, funcionalidades e/ou código fonte de um sistema, como se o sistema tivesse uma identidade que deve ser mantida entre todas as suas diversas aplicações. Esses padrões fazem com que o usuário final esteja habituado com o sistema e consiga intuitivamente utilizar novas funções por conta. No ponto de vista de código, é importante utilizar uma mesma identidade, ou política de boas práticas em todo o sistema, assim além de ser mais coeso, os programadores também terão mais agilidade em suas tarefas.

c) Cite e comente a(s) estratégia(s) para tornar sistemas de software mais flexíveis e fáceis de entender.

R: Desenvolvimento em paralelo: vários desenvolvedores implementando em paralelo as diversas classes, e elas contém ocultam suas principais informações.

Flexibilidade a mudanças: Quando alguma classe que está causando problemas ao software tem parte de seus detalhes de implementação ocultos do restante do sistema, fica mais fácil fazer sua substituição por alguma melhor.

Facilidade de entendimento: Quando um novo desenvolvedor tiver contato com o sistema, ele irá enxergar apenas parte da implementação, tornando seu entendimento mais fácil.

d) Comente a relação entre Coesão e Separação de Interesses.

R: Um sistema coeso, tem suas funcionalidades bem divididas, cada classe é responsável pelo seu próprio objetivo, assim como os métodos, que tem uma função específica e bem definida.

e) Conceitue Acoplamento no contexto do código fonte de um software.

R: é uma amarração, uma conexão entre duas classes ou camadas de um software, geralmente ocorre quando há uma dependência muito grande entre esses componentes ou eles conhecem demais sobre os detalhes de implementação uns dos outros.

Dica: Enriqueça seu aprendizado com os exercícios de fixação que estão ao final do capítulo.