

# Princípios em Projeto de Software

## Atividades de Aprendizagem e Avaliação

Aluno: **Gabriel Junior Picagevicz RA: 2096943**

Use esta cor em seu texto

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

a) Comente a afirmação de John Ousterhout.

Ele não deixa claro porém deu para entender que quando falamos de projetos de software estamos procurando a solução de um problema em específico, e que esse problema surge na implementação de um sistema para atender os requisitos seja ele funcionais ou não, ele também sugere que devemos decompor o problema em pequenas partes, e que essas partes menores podem ser resolvida de forma independente.

Concordo com a afirmação do autor, visto que ao quebrar em partes menores o problema, podemos resolvemos de forma independente e assim diminuir a complexividade.

b) Comente o conceito de Integridade Conceitual.

Brooks propôs que um sistema não pode ser um amontoado de telas ou funcionalidades sem coerência e também sem coesão entre essas partes.

Eu concordo com afirmação, pois ao usar por exemplo um sistema de venda, fico confortável e seguro em procurar outros produtos pois a funcionalidade desse sistema sempre estão consistentes, acredito que se um usuário necessitasse de um aplicativo e tivesse divergência por exemplo dos preços dos produtos de real para dolar iria ficar desconfiado da qualidade do software, portanto, a integridade conceitual é fundamental para qualidade do software.

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

Padronização da equipe em usar por exemplo um modelo de nomeação de variáveis seja ele *camel case* ou *snake case* por exemplo, a padronização de utilizar um determinado framework para a manipulação de páginas web, cada classe manter apenas sua responsabilidade, e sem essas padronizações torna-se difícil do desenvolvedor aplicar evolução ou manutenção no sistema.

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

Qualquer classe deve ser coesa, sendo assim toda classe deve implementar uma única funcionalidade ou serviço, a coesão tem as seguintes vantagens:

Facilita a implementação de uma classe, bem como o seu entendimento e manutenção.

Facilita a alocação de um único responsável por manter uma classe.

Facilita o reuso e teste de uma classe, pois é mais simples reusar e testar uma classe coesa do que uma classe com várias responsabilidades.

Já a Separation of Concerns ou Separação de interesses é semelhante ao conceito de coesão porém ela defende que uma classe deve implementar apenas um concern (interesse), por exemplo: uma classe deve possuir apenas uma responsabilidade.

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

Acoplamento está relacionado com a força da conexão entre duas classes, ou seja, uma classe depende de outra pra funcionar.

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