

# Atividades de Aprendizagem e Avaliação

## Revisão – Princípios em Programação

**Aluno: João Pedro Rodrigues Leite RA: a2487055**

1. Considerando o texto “Capítulo 2 – Código Limpo” pg 7 a 18
  - a) Na avaliação da qualidade de um código podemos considerar aspectos tais como:
    - i. Testabilidade
    - ii. Eficiência
    - iii. Facilidade de modificação
    - iv. Processo pelo qual foi desenvolvido
  - b) Um código limpo apresenta pelo menos as três características a seguir
    - i. Elegância
    - ii. Facilidade de alteração
    - iii. Simplicidade
  - c) Um código limpo está inserido em um estilo de programação que busca proximidade a três valores:
    - i. Expressividade
    - ii. Simplicidade
    - iii. Flexibilidade
  - d) A medida para a expressividade está na facilidade para um desenvolvedor que não o próprio autor ....
  - e) A flexibilidade reflete a facilidade de estender a aplicação sem fazer grandes alterações na estrutura já implementada

- f) As versões iniciais de um método, classe e outras estruturas, nunca **são exatamente uma boa solução**
- g) A presença de testes automatizados são **a fundação para o desenvolvimento de um código limpo**
- h) Se as variáveis estiverem devidamente nomeadas, não precisaremos **criar um comentário para explicá-las**
- i) Se os métodos estiverem bem nomeados e possuírem uma única tarefa, não será necessário **documentar o que são parâmetros e o valor de retorno**
- j) Os nomes dos métodos devem ser escolhidos de modo que descrevem **muito bem a tarefa que realizam**
- k) A economia de palavras deve ser descartada em favor **de uma boa expressividade**
- l) Cada método deve ser pequeno o suficiente para facilitar **sua leitura e compreensão.**
- m) Sabemos que estamos trabalhando em um código limpo quando **cada rotina que temos faz o que esperávamos (Martin, 2008).**
- n) A **Composição de métodos** é a base para a criação de um código limpo.
- o) Um método explicativo deve ser criado quando **ficamos tentados a comentar uma única linha de código.**
- p) Se um método tem uma longa estrutura de *ifs* e *elses* encadeados, o leitor **terá dificuldades para compreender todos os casos e fluxos possíveis**

- q) Se um método recebe muitos argumentos, provavelmente **os utiliza para um conjunto de operações e não uma somente.**
- r) Devemos preferir o uso de exceções em lugar **de códigos de erro e valores nulos.**

2. Considerando o Texto “Capítulo 2” páginas 18 até 30

- a) Programar buscando minimizar o tamanho das classes nos auxilia a **criar unidades coesas e a evitar duplicações**
- b) Uma classe Controlador com os métodos capturaEntrada(), criaCorpoHtml() e criaCabecalhoHtml() possui dois tipos de tarefas diferentes: **lidar com a criação de HTML e processar a entrada do usuário.**
- c) A coesão da classe está intimamente relacionada com **as responsabilidades que assume.**
- d) A medida do acoplamento está atrelada a quanto às classes do sistema **dependem uma das outras.**
- e) Só devemos investir em flexibilidade em partes do sistema que realmente **precisam de flexibilidade**