

**Nome: VINICIUS BARBOSA DA SILVA**

1) **Padrões de projeto** é possível identificar os pontos comuns entre duas soluções diferentes para um mesmo problema. Conhecer esses pontos comuns nos permite desenvolver soluções cada vez melhores e mais eficientes que podem ser reutilizadas, permitindo, assim, o avanço do conhecimento humano. padrões de projeto propicia a construção de aplicações e ou estruturas de código de forma flexível e a documentação de soluções reaproveitáveis.

2) **Um nome:**

-Descreve o problema do projeto, suas soluções e consequências em poucas palavras.

-Permite projetar num nível mais alto de abstração.

**O Problema:**

-Descreve quando aplicar o padrão.

-Descreve o problema e o contexto.

-Pode descrever estruturas de objetos ou de classes que são sintomas de um projeto inflexível.

-Às vezes, o padrão lista condições que devem se aplicar para usar o padrão.

**A Solução:**

-Descreve os elementos constituintes do projeto, seus relacionamentos, responsabilidades e colaborações.

-A solução não descreve um projeto ou implementação concretos porque um padrão é um *gabarito* de solução para várias situações.

**A consequência**

- Os resultados e trade-offs da aplicação do padrão.
- Diz respeito a trade-offs de espaço, tempo, flexibilidade, extensibilidade, portabilidade.

3) Os frameworks constituem uma das mais promissoras correntes tecnológicas para suportar a reutilização de software em grande escala, modelando tanto a parte genérica – frozen spots – como as partes variáveis – hot spots – de um sistema. E que eles devem ser construídos fazendo uso dos padrões de projetos – design patterns – para melhor documentar a sua arquitetura e proporcionar um vocabulário uníssono entre os desenvolvedores.

- 4) No desenvolvimento de software é possível identificar que a procura por uma solução de um problema específico possui características idênticas, senão igual a encontrada em um projeto anteriormente desenvolvido. Neste sentido, pode-se dizer que padrões de projetos criam um vocabulário comum de projeto, pois proporcionam elementos que conduzem ao **reaproveitamento de soluções**, e **não apenas a reutilização de código**.
  - 5) De certa forma os padrões ajudam a documentar um projeto, pois, ao utilizá-los, temos um “mapa” para a solução do problema, mas não uma estratégia. Desta forma, a utilização deste “mapa” pode nos ajudar a encontrar as similaridades e como resolver o problema, facilitando sua documentação.
  - 6) Não faz sentido pois o Padrão Strategy sugere que se produza uma família de classes para cada variação do algoritmo e que se forneça para a classe hospedeira uma instância de Strategy para a qual ela delegará em tempo de execução. Desta forma, se eu tenho apenas uma única variação, não faz sentido eu usar o padrão Strategy.
- 9)
- O padrão **Chain of Responsibility** não é um padrão frequente em um programa **Java**, pois é relevante apenas quando o código opera com cadeias de objetos. Um dos casos de uso mais populares para o padrão é a interferência de eventos nos componentes pai nas classes da GUI.