Atividades de Aprendizagem e Avaliação

Separação de Conceitos

SoC – Separation of Concerns

Use esta cor no seu texto

Aluno: Gustavo Matheus Pauvels RA: 2371278

1. Complete as sentenças a seguir

   a. Separar conceitos, é na verdade um princípio de projeto para subdividir um programa em seções distintas de tal forma que cada seção se fresponsabilize por um único interesse.

   b. Uma das maneiras de se construir um sistema contemplando a separação de conceitos, é separar a sua aplicação em camadas e fazer com que cada uma delas foque em resolver tarefas apenas de sua responsabilidade. As camadas, separam conceitos de acordo com a sua função dentro de uma arquitetura.

   c. A interação entre camadas deve ocorrer de modo a não invadir a privacidade ou responsabilidade da outra.

   d. A separação de conceitos promove a Single Responsability Principle (SOLID/SRP) e a Separation Of Concerns (SOC) na hora de desenvolver.

   e. A camada ‘view’ tem como responsabilidade apresentar para o usuário a interface com a qual ele irá iteragir, ou seja, a única responsabilidade desta camada é a iteração com o usuário.

   f. A camada ‘Aplication Layer’ tem por finalidade obter requisições da view e passar para a camada de persistência que irá realizar alguma operação e retornar ou não um resultado.

   g. A camada ‘Aplication Layer’ se relaciona com Camada de Interface do Modelo de Arquitetura visto nos slides da aula.

   h. *Domain Layer* é o DDD e está intimamente ligada com a abordagem de armazenar os objetos de negócios, e todos os relacionamentos entre estes objetos.

   i. A *Database Layer* é responsável por realizar a persistência e armazenar as informações no banco de dados.

j. O que torna um código fácil de fazer manutenção é junte as unidades que mudam pelo mesmo motivo, separe as que mudam por motivos diferentes.

k. No desenvolvimento de software, uma responsabilidade é separar conceitos.

l. É comum que a camada de infraestrutura utilize o *repository pattern* para comunicação com o banco de dados

m. A Camada de Interface não deve possuir regras de negócio ou casos de uso sua responsabilidade é a passagem dos dados de entrada (como parâmetros de uma URL) para um caso de uso da *camada de aplicação* e a devolução da resposta da mesma para o usuário.

n. Se migrar a camada Web API de Express para Hapi, causa mudanças em outras camadas da sua aplicação é sinal de que há acoplamento entre elas.

o. Acoplamento é a medida de quão forte é a ligação entre os elementos que compõe o módulo.

p. O pacote Awilix tem por objetivo permitir que você utilize a técnica de fazer que toda dependência de uma classe seja recebida no seu construtor em vez de criada pela própria instância sem acoplar seu código à ferramenta.