Atividades de Aprendizagem e Avaliação

Separação de Conceitos

SoC – Separation of Concerns

Use esta cor no seu texto

Aluno:_JACSON SIQUEIRA CESAR_____RA:__2487012_____

1. Complete as sentenças a seguir

   a. Separar conceitos, é na verdade fazer com que uma aplicação seja modularizada, afim de focar resolver um único problema e caso precise realizar algo que não faz parte de sua tarefa, pede a outro módulo para colaborar quando se fizer necessário.

   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.

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

   d. A separação de conceitos promove a ter um design simplificado e a evoluir com facilidade na hora de desenvolver.

   e. A camada 'view' tem como responsabilidade a 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 abstrai toda a iteração com a fonte de dados, sendo de sua responsabilidade 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 a camada View do Modelo de Arquitetura visto nos slides da aula.

   h. *Domain Layer* é o armazena os objetos de negócios, e todos os relacionamentos entre estes objetos e está intimamente ligada com a abordagem de focar a modelagem nesta camada ou seja nas ideias, nos conceitos, na automatização do processo de negócios de acordo com as expectativas do usuário .

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

   j. O que torna um código fácil de fazer manutenção é os bons princípios de engenharia de software serem completamente ignorados.

k. No desenvolvimento de software, uma responsabilidade é uma tarefa a qual uma unidade se propõe a realizar: representar o conceito de "produto" na sua aplicação, receber requisições da rede, salvar um usuário no banco de dados, entre outros.

l. É comum que a camada de infraestrutura utilize o *repository pattern* para comunicação com o banco de dados (ou algum outro serviço externo de persistência, como uma API).

m. A Camada de Interface não deve possuir .nenhum conhecimento sobre as regras de negócio, casos de uso, tecnologias de persistência, nem nenhuma outra lógica 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 um princípio de projeto para subdividir um programa em seções distintas de tal forma que cada seção se fresponsabilize por um único interesse.

p. O pacote Awilix tem por objetivo permitir que você utilize a técnica sem acoplar seu código à ferramenta, e não sentir que está usando aquele ~~estranho~~ mecanismo de *dependency injection* do Angular 1, por exemplo.