**Atividades de Aprendizagem e Avaliação**

**Separação de Conceitos**

**SoC – Separation of Concerns**

**Use esta cor no seu texto**

Aluno: André Luis Quiosi RA: 2369958

1. Complete as sentenças a seguir
   1. Separar conceitos, é, na verdade, fazer com que uma aplicação seja modularizada.
   2. 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.
   3. A interação entre camadas deve ocorrer de modo a não invadir a privacidade ou responsabilidade da outra.
   4. A separação de conceitos promove a produtividade e a agilidade na hora de desenvolver.
   5. A camada ‘view’ tem como responsabilidade apresentar para o usuário a interface com a qual ele interage.
   6. A camada 'Application Layer’ tem por finalidade obter requisições da view e passar para a camada de persistência que realizará alguma operação e retornar ou não um resultado.
   7. A camada 'Application Layer’ se relaciona com a camada View do Modelo de Arquitetura visto nos slides da aula.
   8. *Domain Layer* é o coração do software de negócios e está intimamente ligada com a abordagem DDD
   9. A *Database Layer* é responsável por realizar a persistência e armazenar as informações do banco de dados.
   10. O que torna um código fácil de fazer manutenção é a separação de conceitos.
   11. No desenvolvimento de software, a responsabilidade é uma tarefa a qual uma unidade se propõe a realizar.
   12. É comum que a camada de infraestrutura utilize o repository pattern para comunicação com o banco de dados, serviços de e-mail e sistemas de fila.
   13. A Camada de Interface não deve possuir conhecimento sobre as regras de negócio ou casos de uso, sua responsabilidade é apenas 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.
   14. Se migrar a camada Web API de Express para Hapi, causa mudanças em outras camadas da sua aplicação é fortemente acoplada
   15. Acoplamento é quando as camadas da aplicação dependem das funções umas das outras, dificultando a mudança.
   16. O pacote Awilix tem por objetivo permitir que você utilize a técnica dependency injection sem acoplar seu código à ferramenta.