



## **Parte 1 - Exercícios de Fixação**

1. Dada a sua complexidade, sistemas de bancos de dados são componentes relevantes na arquitetura de qualquer tipo de sistema. Verdadeiro ou falso? Justifique a sua resposta.
2. Descreva três vantagens de arquiteturas MVC.
3. Qual a diferença entre classes Controladoras em uma Arquitetura MVC tradicional e classes Controladoras de um sistema Web implementado usando um framework MVC como Ruby on Rails?
4. Descreva resumidamente quatro vantagens de microserviços.
5. Por que microserviços não são uma bala de prata? Isto é, descreva pelo menos três desvantagens do uso de microserviços.
6. Explique a relação entre a Lei de Conway e microserviços.
7. Explique o que significa desacoplamento no espaço e desacoplamento no tempo. Por que arquiteturas baseadas em filas de mensagens e arquiteturas Publish/Subscribe oferecem essas formas de desacoplamento?
8. Quando uma empresa deve considerar o uso de uma arquitetura baseada em filas de mensagens ou uma arquitetura publish/subscribe?
9. Explique o objetivo do conceito de tópicos em uma arquitetura publish/subscribe.
10. (POSCOMP, 2019, adaptado) Marque V ou F.  
( ) O padrão MVC é uma adaptação do padrão arquitetural Camadas. A Camada Visão lida com a apresentação e a manipulação da interface, a Camada Modelo organiza os objetos específicos da aplicação, e a Camada Controle posiciona-se entre estas duas com as regras do negócio.

( ) O padrão Broker é voltado a problemas de ambientes distribuídos. Sugere uma arquitetura na qual um componente (broker) estabelece uma mediação que permite um desacoplamento entre clientes e servidores.

( ) Mesmo que um dado padrão arquitetural ofereça uma solução para o problema sendo resolvido, nem sempre ele é adequado. Fatores como contexto e o sistema de forças que afeta a solução fazem também parte do processo de avaliação e da escolha de padrões adequados.

## **Parte 2 - Atividades práticas**

- 1) Exemplo (didático) de um sistema MVC Web:

<https://replit.com/@engsoftmoderna/ExemploArquiteturaMVC>

Navegue no link **Show files** para visualizar o fonte do projeto.

Indique quais arquivos devem ser alterados para que o sistema de exemplo possibilite a consulta pelo “isbn text” do livro.

- 2) Roteiro prático sobre microserviços

<https://github.com/aserg-ufmg/micro-livraria>

### **2.1 Tarefa Prática #1: Implementando uma Nova Operação**

### **2.2 Tarefa Prática #2: Criando um Container Docker**

- 3) Roteiro prático sobre Publish/Subscribe

<https://github.com/aserg-ufmg/pub-sub-store>

### **3.1 Passo 1: Instalando, Executando e Inicializando o RabbitMQ**

## **3.2 Passo 2: Subindo os Serviços**

## **3.3 Passo 3: Colocando a Mão na Massa**

**Atenção à entrega:** Poste os arquivos que você trabalhou nestas atividades na sua conta do GitHub e coloque o link no google Classroom do seu projeto no GitHub.