

# Atividade Objetiva 03

Entrega Sem prazo

Pontos 10

Perguntas 10

Limite de tempo Nenhum

Tentativas permitidas Sem limite

[Fazer o teste novamente](#)

## Histórico de tentativas

	Tentativa	Tempo	Pontuação
MANTIDO	<a href="#">Tentativa 3</a>	Menos de 1 minuto	10 de 10
MAIS RECENTE	<a href="#">Tentativa 3</a>	Menos de 1 minuto	10 de 10
	<a href="#">Tentativa 2</a>	2 minutos	9 de 10
	<a href="#">Tentativa 1</a>	34 minutos	9 de 10

⚠ As respostas corretas estão ocultas.

Pontuação desta tentativa: **10** de 10

Enviado 28 ago em 1:15

Esta tentativa levou Menos de 1 minuto.

### Pergunta 1

1 / 1 pts

APIs e microserviços possuem uma relação complementar. Sobre esses estilos arquiteturais podemos dizer que:

☐ APIs implicam no uso de microserviços

☒ Microserviços suportam a disponibilização de APIs modulares, escaláveis e com tolerância a falhas

☐ Microserviços somente podem ser usados para APIs REST



APIs REST e RPC não possuem ainda suporte para microsserviços

## Pergunta 2

1 / 1 pts

Quando empresas adotam APIs em larga escala elas podem ser beneficiar de um acelerador arquitetural chamado de Plataforma de APIs.

Tipicamente, uma plataforma de API é composta de quais componentes centrais?



Ferramenta de documentação de APIs, Plataforma de Desenvolvimento de APIs e Plataforma de Engajamento para Desenvolvedores



Ferramentas de documentação de APIs, Plataformas de Execução de APIs (Runtime) e Plataforma de Engajamento



Plataformas de Execução de APIs, Plataforma de Engajamento e Plataformas de Monetização



Plataformas de Desenvolvimento de APIs, Plataformas de Execução de APIs (Runtime), e Plataformas de Engajamento para Desenvolvedores.

## Pergunta 3

1 / 1 pts

Sobre o padrão API Gateway, não podemos dizer que:



Ele isola clientes de APIs (ex. Web e Model) de componentes backend



Ele facilita a exposição de múltiplos microsserviços para clientes móveis



Ele não pode ser usado para a exposição de APIs de aplicações monolíticas



Ele facilita funções de segurança, escalabilidade e até monetização de APIs

#### Pergunta 4

1 / 1 pts

Um outro padrão arquitetural que tem sido muito usado na adoção de APIs e serviços Web é o BFF - Backend for Frontend. Sobre esse padrão, podemos dizer que:



Ele é usado para containerizar APIs em ambientes Docker e Kubernetes



Ele é uma derivação do padrão API Gateway. A diferença é granulosidade das APIs, que agora é centrada em clientes específicos como um App ou Site Web



Ele tem o mesmo significado que o padrão API Gateway (são sinônimos)



Ele é uma derivação do padrão SAGA e usado para descentralizar dados para clientes Web

### Pergunta 5

1 / 1 pts

Os padrões API Health Check e Disjuntor (Circuit Breaker) fornecem capacidades importantes para a operação de tempo de execução de APIs. Essas capacidades são, respectivamente:

- ☐ Segurança e Tolerância a Falhas
- ☒ Monitoração e Tolerância a Falhas
- ☐ Escalabilidade e Tolerância a Falhas
- ☐ Monitoração e Escalabilidade

### Pergunta 6

1 / 1 pts

APIs fornecem funcionalidades que executam comportamentos diversos e precisam retornar dados para os seus clientes. A respeito de APIs e os dados manipulados em servidores Web podemos dizer que:

- ☐ Uma API não pode acessar dados de múltiplas tabelas
- ☐ Uma API não pode acessar dados de múltiplos bancos de dados
- ☐ Uma API somente pode manipular bancos relacionais caso precise retornar dados em formato JSON



Uma API pode usar dados de múltiplos bancos de dados através da arquitetura de microsserviços

### Pergunta 7

1 / 1 pts

APIs precisam garantir a consistência dos dados manipulados nos servidores. Sobre APIs e controle transacional podemos dizer o seguinte:



APIs não possuem suporte a transações



APIs que fazem acesso a dois ou mais microsserviços não possuem suporte a controle transacional



APIs que fazem acesso a dois ou mais microsserviços possuem suporte a controle transacional executado nativamente por 2PC



APIs que fazem acesso a dois ou mais microsserviços que precisam de suporte a controle transacional normalmente fazem uso programático através de padrões como o SAGA

### Pergunta 8

1 / 1 pts

Quando você disponibiliza uma API, pode ser útil criar testes automatizados que verifiquem se os dados retornados estão em conformidade com as expectativas de seus clientes (serviço consumidor). Conforme a pirâmide de testes, esses testes são chamados de:

☐ Testes de integração (Gateway ou HTTP Client)

☒ Testes de Contratos

☐ Testes Exploratórios

☐ Testes de componentes

### Pergunta 9

1 / 1 pts

Quando uma API é criada ele precisa de um teste que verifique se o seu serviço está plenamente funcional, Isso não é sempre trivial pois APIs também podem chamar outros serviços externos. Para checar esses cenários complexos quando APIs chamam outras APIs/serviços usamos o seguinte tipo de testes:

☐ Testes E2E

☐ Testes de Contratos

☒ Testes de integração (Gateway ou HTTP Client)

☐ Testes de serviço

### Pergunta 10

1 / 1 pts

São aspectos arquiteturais importantes para a criação de APIs modernas:

☐ Uso de padrões arquiteturais como o MVVM



Uma boa arquitetura de estabilidade com o uso de automação de testes de contratos, serviços, componente e integração.



Uso obrigatório de microserviços



Uso mandatório do padrão REST para qualquer cenário

Pontuação do teste: **10** de 10