**✅ Justificativas das Questões**

**1. Qual é o principal objetivo de definir "roles" (funções) em uma aplicação mobile?**

* **a) Reduzir o tempo de desenvolvimento da aplicação.**  
  ❌ *Errado*. Definir roles pode até **aumentar a complexidade**, pois exige planejamento de acesso.
* **b) Melhorar a interface gráfica do usuário.**  
  ❌ *Errado*. Roles não afetam diretamente a interface, mas sim **quem pode acessar o quê**.
* **c) Aumentar a quantidade de dados armazenados.**  
  ❌ *Errado*. Roles não influenciam a capacidade de armazenamento de dados.
* **d) Permitir que diferentes tipos de usuários tenham acessos distintos.**  
  ✅ *Correto*. Roles são **perfis de acesso** que permitem segmentar funcionalidades conforme o tipo de usuário.

**2. O que uma política de acesso define em uma aplicação?**

* **a) A aparência da tela inicial do aplicativo.**  
  ❌ *Errado*. Isso é parte do design UI, não da segurança.
* **b) A velocidade de carregamento das páginas.**  
  ❌ *Errado*. Esse fator está relacionado à performance, não à política de acesso.
* **c) Quem pode acessar determinadas funcionalidades ou dados.**  
  ✅ *Correto*. Políticas de acesso controlam **quais funções e dados cada perfil pode acessar**.
* **d) O tipo de banco de dados utilizado.**  
  ❌ *Errado*. O banco de dados é definido na arquitetura, não pela política de acesso.

**3. Qual é um exemplo de controle de acesso bem implementado?**

* **a) Permitir que todos os usuários acessem todas as funcionalidades.**  
  ❌ *Errado*. Isso **viola a segurança**, pois não há restrição de permissões.
* **b) Criar um único perfil de usuário para todos os acessos.**  
  ❌ *Errado*. Isso **centraliza demais o controle** e aumenta o risco de abuso.
* **c) Restringir funcionalidades administrativas apenas aos administradores.**  
  ✅ *Correto*. Isso é uma boa prática: **limitar funções críticas a perfis específicos**.
* **d) Fazer com que o usuário defina suas próprias permissões.**  
  ❌ *Errado*. Isso compromete a segurança e **foge do controle da aplicação**.

**4. Qual é a principal característica da arquitetura monolítica?**

* **a) Cada funcionalidade é um serviço independente.**  
  ❌ *Errado*. Isso é uma característica dos **microsserviços**.
* **b) Comunicação entre serviços via API.**  
  ❌ *Errado*. Essa é outra marca dos **microsserviços**.
* **c) Tudo está integrado em um único bloco de código.**  
  ✅ *Correto*. A arquitetura monolítica agrupa **todas as funções em um único projeto**.
* **d) Escalabilidade fácil e independente.**  
  ❌ *Errado*. Escalabilidade em sistemas monolíticos **é mais difícil**.

**5. Qual é um dos principais desafios ao migrar para uma arquitetura de microsserviços?**

* **a) Manter a aplicação em um único bloco de código.**  
  ❌ *Errado*. Isso **não é objetivo** dos microsserviços.
* **b) Reduzir a complexidade do sistema.**  
  ❌ *Errado*. Microsserviços **aumentam a complexidade**, principalmente em comunicação.
* **c) Garantir a comunicação eficiente entre serviços.**  
  ✅ *Correto*. A integração entre serviços é um desafio, por usar **chamadas de rede e APIs**.
* **d) Eliminar a necessidade de testes.**  
  ❌ *Errado*. Microsserviços **exigem ainda mais testes**.

**6. Qual padrão de design é comumente usado para garantir a resiliência em uma arquitetura de microsserviços?**

* **a) Singleton**  
  ❌ *Errado*. Usado para **garantir uma única instância**, não resiliência.
* **b) Factory Method**  
  ❌ *Errado*. Padrão de criação de objetos, sem relação direta com resiliência.
* **c) Circuit Breaker**  
  ✅ *Correto*. Evita falhas em cascata **interrompendo chamadas** para serviços com falhas.
* **d) Observer**  
  ❌ *Errado*. Relacionado a **eventos e notificações**, não à resiliência.

**7. Qual das opções abaixo é uma ferramenta comum para realizar testes de integração em APIs back-end?**

* **a) Notepad**  
  ❌ *Errado*. É apenas um editor de texto.
* **b) Google Analytics**  
  ❌ *Errado*. Serve para análise de tráfego web, **não para testes**.
* **c) Postman**  
  ✅ *Correto*. Ferramenta usada para **testar requisições HTTP em APIs**.
* **d) Visual Studio Code**  
  ❌ *Errado*. É um **editor de código**, não focado em testes de integração.

**8. Qual cenário de teste seria adequado para verificar a integração de uma API de pedidos com o banco de dados?**

* **a) Verificar se a API exibe um gráfico de vendas.**  
  ❌ *Errado*. Isso é visualização, não integração com banco.
* **b) Verificar se a API permite alterar a fonte da interface.**  
  ❌ *Errado*. Isso é questão de design, não de back-end.
* **c) Verificar se a API altera as permissões do usuário.**  
  ❌ *Errado*. Isso diz respeito à **autenticação/autorização**, não pedidos.
* **d) Verificar se a API retorna um pedido ao receber o ID correto.**  
  ✅ *Correto*. Isso valida **a comunicação entre API e banco**.

**9. Qual é um dos principais benefícios dos testes de integração em um sistema de e-commerce?**

* **a) Eliminar todos os bugs no sistema.**  
  ❌ *Errado*. Nenhum teste **elimina 100% dos erros**.
* **b) Substituir os testes unitários.**  
  ❌ *Errado*. São **testes complementares**, não substitutos.
* **c) Reduzir o tempo de desenvolvimento.**  
  ❌ *Errado*. Testes de integração geralmente **aumentam o tempo**, mas garantem qualidade.
* **d) Garantir que diferentes componentes funcionem bem juntos.**  
  ✅ *Correto*. Esse é o foco do **teste de integração**.

**10. Por que os padrões de design são importantes em arquiteturas de software?**

* **a) Para criar interfaces visuais mais bonitas.**  
  ❌ *Errado*. Isso é responsabilidade do front-end.
* **b) Para aumentar o uso de memória do sistema.**  
  ❌ *Errado*. Isso é um efeito colateral indesejado.
* **c) Para estruturar o código de forma mais eficiente e organizada.**  
  ✅ *Correto*. Design patterns promovem **clareza, modularidade e manutenção**.
* **d) Para eliminar a necessidade de testes de software.**  
  ❌ *Errado*. Testes **continuam necessários**, mesmo com bom design.