APIs e Repository Pattern

# O que é uma API?

1. Um conjunto de regras que define como diferentes softwares se comunicam. OK
2. Um aplicativo que permite o acesso a um banco de dados.
3. Um sistema de armazenamento de dados.
4. Um protocolo de comunicação para a internet.

# Qual é a principal função de uma API?

1. Armazenar dados.
2. Criar interfaces gráficas.
3. Facilitar a comunicação entre sistemas diferentes. OK
4. Gerar relatórios.

# Qual é a diferença entre uma API pública e uma API privada?

1. Uma API pública é gratuita, enquanto uma API privada é paga. OK
2. Uma API pública é acessível a qualquer pessoa, enquanto uma API privada é acessível apenas a usuários autorizados.
3. Uma API pública é usada para aplicativos web, enquanto uma API privada é usada para aplicativos mobile.
4. Uma API pública é mais complexa do que uma API privada.

# Qual é o formato de dados mais comum usado em APIs?

1. XML
2. JSON OK
3. HTML
4. CSV

# O que é REST?

1. Uma linguagem de programação para desenvolvimento de APIs.
2. Um protocolo de comunicação para APIs.
3. Um padrão arquitetural para APIs.
4. Um sistema de gerenciamento de bancos de dados.

# Qual é o principal benefício de usar REST para desenvolver APIs?

1. Maior segurança.
2. Melhor desempenho.
3. Facilidade de uso.
4. Todas as alternativas acima.

# O que é um endpoint em uma API?

1. Um ponto de acesso específico a um recurso. OK
2. Um servidor que hospeda a API.
3. Um cliente que usa a API.
4. Um método de autenticação.

# Qual é a diferença entre um método GET e um método POST em uma API?

1. GET é usado para recuperar dados, enquanto POST é usado para enviar dados. OK
2. GET é usado para enviar dados, enquanto POST é usado para recuperar dados.
3. GET é mais seguro que POST.
4. POST é mais seguro que GET.

# O que é o padrão Repository Pattern?

1. Um padrão de projeto que define como armazenar dados em um banco de dados.
2. Um padrão de projeto que define como acessar dados de um banco de dados.
3. Um padrão de projeto que define como comunicar diferentes softwares.
4. Um padrão de projeto que define como gerenciar a segurança de uma aplicação.

# Qual é a principal vantagem de usar o Repository Pattern?

1. Simplificar o acesso a dados.
2. Aumentar a segurança do sistema.
3. Melhorar o desempenho da aplicação.
4. Todas as alternativas acima.

# O que é um repository?

1. Um objeto que representa uma fonte de dados. OK
2. Um objeto que encapsula as regras de negócio.
3. Um objeto que gerencia a interface gráfica.
4. Um objeto que controla o acesso a outros objetos.

# Qual é a relação entre o Repository Pattern e o conceito de abstração?

1. O Repository Pattern não se relaciona com a abstração.
2. O Repository Pattern usa a abstração para esconder os detalhes de implementação do banco de dados.
3. O Repository Pattern é uma forma de implementar a abstração.
4. O Repository Pattern é a única forma de implementar a abstração.

# Qual é a função do interface IRepository?

1. Definir os métodos de acesso aos dados.
2. Implementar os métodos de acesso aos dados.
3. Criar a conexão com o banco de dados.
4. Gerenciar o fluxo de dados.

# Qual é a diferença entre o repository e o DAO (Data Access Object)?

1. O repository é mais genérico que o DAO.
2. O DAO é mais genérico que o repository.
3. O repository é mais complexo que o DAO.
4. O DAO é mais complexo que o repository.

# Quando é recomendado usar o Repository Pattern?

1. Em qualquer aplicação que precise acessar dados.
2. Em aplicações que precisam de um acesso mais flexível aos dados.
3. Em aplicações que usam um banco de dados relacional.
4. Em aplicações que usam uma API REST.

# O que é a camada de persistência em uma aplicação?

1. A camada responsável por armazenar os dados.
2. A camada responsável por gerenciar a interface gráfica.
3. A camada responsável por processar os dados.
4. A camada responsável por se comunicar com outros sistemas.

# Qual é a relação entre a camada de persistência e o Repository Pattern?

1. O Repository Pattern é uma forma de implementar a camada de persistência.
2. A camada de persistência é uma forma de implementar o Repository Pattern.
3. O Repository Pattern e a camada de persistência são conceitos distintos.
4. O Repository Pattern e a camada de persistência são sinônimos.

# Qual é a vantagem de usar o Repository Pattern em conjunto com o Dependency Injection?

1. Permite a troca de implementações do repository sem alterar o código da aplicação.
2. Aumenta a segurança da aplicação.
3. Melhora o desempenho da aplicação.
4. Facilita o teste da aplicação.

# O que é um Unit Test?

1. Um teste que verifica a funcionalidade completa da aplicação.
2. Um teste que verifica a funcionalidade de um módulo específico da aplicação.
3. Um teste que verifica a segurança da aplicação.
4. Um teste que verifica o desempenho da aplicação.

# Qual é a importância de realizar testes unitários na camada de persistência?

1. Assegurar a qualidade e a integridade dos dados.
2. Identificar erros de código e corrigir bugs.
3. Facilitar a manutenção do sistema.
4. Todas as alternativas acima.

# Qual é a diferença entre uma API e um Web Service?

1. API é um conjunto de regras para comunicação entre softwares, enquanto Web Service é uma aplicação que fornece serviços através de uma API.
2. Web Service é um conjunto de regras para comunicação entre softwares, enquanto API é uma aplicação que fornece serviços através de uma API.
3. Não existe diferença, ambos são sinônimos.
4. API é um protocolo de comunicação, enquanto Web Service é um sistema de gerenciamento de dados.

# O que é um middleware em uma API?

1. Um software que se conecta à API e fornece funcionalidades adicionais.
2. Um servidor que hospeda a API.
3. Um cliente que usa a API.
4. Um método de autenticação.

# Qual é a função de um gateway de API?

1. Redirecionar as requisições de API para o servidor correto.
2. Controlar o acesso à API.
3. Gerenciar o cache da API.
4. Todas as alternativas acima.

# O que é a autenticação em uma API?

1. O processo de verificar a identidade do usuário.
2. O processo de verificar as permissões do usuário.
3. O processo de criptografar os dados da API.
4. O processo de gerenciar o acesso à API.

# Qual é a função da autorização em uma API?

1. Verificar a identidade do usuário.
2. Controlar as ações que o usuário pode realizar.
3. Criptografar os dados da API.
4. Gerenciar o acesso à API.

# Qual é a diferença entre o padrão Repository Pattern e o padrão DAO?

1. O padrão DAO é mais genérico e abrangente, enquanto o padrão Repository Pattern é mais específico para acesso a dados.
2. O padrão Repository Pattern é mais genérico e abrangente, enquanto o padrão DAO é mais específico para acesso a dados.
3. Não existe diferença significativa entre os dois padrões.
4. O padrão DAO é mais focado em testes, enquanto o padrão Repository Pattern é mais focado em desempenho.

# Qual é a relação entre o padrão Repository Pattern e o conceito de SOLID?

1. O padrão Repository Pattern é uma aplicação direta do princípio de Open/Closed Principle.
2. O padrão Repository Pattern é uma aplicação direta do princípio de Interface Segregation Principle.
3. O padrão Repository Pattern é uma aplicação direta do princípio de Dependency Inversion Principle.
4. O padrão Repository Pattern é uma aplicação direta de todos os princípios SOLID.

# Qual é a importância do padrão Repository Pattern em um sistema de testes unitários?

1. Permite a criação de mocks para testar o código da aplicação sem depender de um banco de dados real.
2. Facilita a integração de diferentes bancos de dados no sistema de testes.
3. Permite a simulação de erros e falhas na camada de persistência.
4. Todas as alternativas acima.

# Qual é a vantagem de usar uma API RESTful?

1. Maior flexibilidade e facilidade de uso.
2. Maior segurança e desempenho.
3. Maior compatibilidade com diferentes plataformas.
4. Todas as alternativas acima.

# Qual é a importância de documentar uma API?

1. Facilitar o entendimento e o uso da API por desenvolvedores.
2. Garantir a qualidade e a consistência da API.
3. Facilitar a manutenção e a evolução da API.
4. Todas as alternativas acima.