Questões relacionadas a C#

1. Orientação a Objetos:

• Explique o conceito de herança múltipla e como C# aborda esse cenário.

*Conceito em que uma classe filha pode herdar comportamentos e características, de mais de uma classe pai. C# faz herança múltipla através de interfaces.*

• Explique o polimorfismo em C# e forneça um exemplo prático de como ele pode ser implementado.

*É um conceito de orientada a objetos, que permite que objetos de diferentes tipos pode se comportar de várias formas dependendo do contexto.*

*Exemplo: Classe Animal, Classe Cachorro, Classe Gato, tanto gato como cachorro emitem sons, mas os sons são diferentes, isso pode ser feito sobreescrevendo o método da Classe Animal Som.*

1. SOLID:

• Descreva o princípio da Responsabilidade Única (SRP) e como ele se aplica em um contexto de desenvolvimento C#.

*Princípio da Responsabilidade Única (SRP): Nos diz que uma classe deve ter apenas um motivo de mudança. Isso é importante para divisão de responsabilidades, coesão, facilidade de manutenção.*

• Como o princípio da inversão de dependência (DIP) pode ser aplicado em um projeto C# e como isso beneficia a manutenção do código?

*Através do uso de interfaces injetadas como dependência, evitando depender diretamente de classes concretas, evitando problemas como duplicação de código, gerando desacoplamento também.*

1. Entity Framework (EF):

• Como o Entity Framework gerencia o mapeamento de objetos para o banco de dados e vice-versa?

*Por meio dos mapeamentos, que ele cria baseado nas entidades definidas, e com isso cria o modelo que será gerado através das migrações. dessa forma faz o relacionamento entre objetos entidades do .Net e as entidades do banco.*

• Como otimizar consultas no Entity Framework para garantir um desempenho eficiente em grandes conjuntos de dados?

*Utilizando-se de várias abordagens, como asnotracking para desabilitar o rastreamento das entidades pelo contexto, include para relacionamento, definindo quais entidades serão carregadas com a entidade principal, filtros nos selects evitando consultas etc.*

1. WebSockets:

• Explique o papel dos WebSockets em uma aplicação C# e como eles se comparam às solicitações HTTP tradicionais.

*Websockets são bidirecionais diferente dos HTTP, websockets enquanto não for modificado seu status para closed, a conexão cliente e servidor fica aberta, diferente do HTTP que uma vez recebido a resposta fecha a conexão. Websockets ~e muito utilizado para comunicação direta, como chats, multiplayer etc.*

• Quais são as principais considerações de segurança ao implementar uma comunicação baseada em WebSockets em uma aplicação C#?

*Algumas coisas podem ser consideradas, rate limite para evitar ataque DDOS, validação dados entrada, obrigatoriamente deve ter autorização e autenticação dos clientes, patchs, dentre outras.*

1. Arquitetura:

• Descreva a diferença entre arquitetura monolítica e arquitetura de microserviços. Qual seria sua escolha ao projetar uma aplicação C#?

*Microserviços costumam ser mais desacopladas devido a ter um pouco menos de dependência, geralmente cada microserviço tem sua própria base, são mais resilientes, pois se um microserviço cair a aplicação ainda funciona, maior escalabilidade. Já o monolito é um inverso do microserviço, porém é mais fácil de dar manutenção, menos complexo na maioria dos casos.*

Minha escolha se basearia principalmente nos pilares de acoplamento e escalabilidade, se necessito de baixo acoplamento e maior escalabilidade escolheria microserviços.

• Como você escolheria entre a arquitetura de microserviços e a arquitetura monolítica ao projetar uma aplicação C# que precisa ser altamente escalável?

*Baseando-me no contexto da aplicação, se é mutável ao longo do tempo, se o planejamento desta aplicação já prevê crescimento, necessidade de múltiplas bases, caso contrário optaria por um monolito.*