Perguntas pré-entrevista técnica

O que é melhor, entidades de modelo/domínio com validações e comportamentos ou sem? Pode dar um exemplo real?

Para projetos com a necessidade extensiva de validações na entrada de dados é recomendável o uso de modelo/domínio, estes permitem a validação dos comportamentos validando assim as regras de entrada de dados de uma melhor forma. Um bom exemplo é a entrada de dados pessoais de uma pessoa para geração de um certificado digital, existem regras e campos obrigatórios a serem preenchidos, evitando assim fraudes ou que uma pessoa se passe por outro.

Que vantagens/desvantagens você vê no uso de TDD (desenvolvimento orientado a testes)?

Trabalho com TDD, pois a grandevantagem dele é que você define o comportamento esperado que o código venha a fazer, diminuindo assim o tempo de desenvolvimento, já que a ideia do comportamento, validações, campos e métodos já estão bem definidas, podendo haver durante o processo a inclusão de alguma funcionalidade que passou desapercebido.

Quais práticas você usa para garantir que seu código não é um emaranhado/espaguete? Já passou por essa situação? Pode dar um exemplo real?

Os códigos são escritos em classes essas, tem os seus objetivos bem definidos e responsabilidades, evitando assim duplicidade de código, quando se faz necessário, estende ou importa-se uma classe para utilizar a função previamente feita, esta está bem detalhada e descrita.

Tem conhecimento em orientação a objetos e SOLID? Pode explicar o que entende sobre as letras “S” ou “O”?

S - Single responsiblity principle: significa dizer que uma classe só pode ser especialista em um único assunto, esta só pode executar uma única tarefa a qual foi designada de projetada.

O – Princípio Open Closse: Objetos ou entidades, devem estar fechados para modificações externas, mas podendo serem estanciados e adicionados novos comportamentos aos que já estão definidos previamente na classe herdada.

Design Patterns (padrões de projeto), chegou a usar algum? Pode dar um exemplo real? (Qual problema este padrão resolveu/ajudou?)

Sim, e o mais usual que faço nos projetos por uma questão de necessidade é o Singleton, este garante que uma seja uma única instância no caso em que utilizo, para controle de acessos a recursos compartilhados. Outro que utilizo é o Observer, tilizando para notificações quando muda-se o estados e notifica-se os demais para as mudanças ocorridas.

O que você entende por DDD (design orientado a domínio)?

É o desenvolvimento voltado a domínios, onde uma classe possui um construtor privado e um métodos estático para atribuir os valores passados, assim garantindo a integridade do objeto instanciado, este não possui métodos públicos além do estático de criação da classe, seu construtor interno é quem passa os valores as propriedades, esta técnica permite realizar algumas validações e métodos no seu interior para validar os valores passados, se atendem ou não o objetivo ao qual foi designado.