



Avaliação Técnica .Net

Teoria

1) Assinale as alternativas corretas sobre a linguagem C#:

- a. É uma linguagem fracamente tipada
- b. É uma linguagem fortemente tipada ✓
- c. É interpretada durante a execução
- d. É uma linguagem compilada e gerenciada ✓

2) Defina, com suas palavras, o que é Intermediate Language (IL)?

Também conhecida como CIL (Common Intermediate Language) é uma linguagem de programação de baixo nível é uma linguagem utilizada para fazer a compilação de linguagens de alto nível.

3) Comente sobre o que é o .Net Framework e o .Net Core.

.Net Framework é uma plataforma de desenvolvimento de aplicações que visa uma plataforma única para execução de sistemas e aplicações.

.Net Core é um framework livre, open-source e sucessor do famoso .Net Framework, ou seja, também é uma plataforma de desenvolvimento de aplicações, também é uma solução mais leve e modular que seu antecessor podendo ser utilizada em diferentes sistemas operacionais.

4) O que são classes e objetos? Relacione os dois.

Classes são como “molduras” de algo, como o modelo de um objeto a ser definido no mundo real, pense nela como a moldura de um quadro, a partir da moldura do meu quadro eu posso utilizar(instanciar) quantos pincéis e cores de tintas forem necessários (objetos) para fazer meu desenho, e eles são independentes da minha moldura(classe) e um do outro. Basicamente é minha moldura(classe) quem define as “regras” da minha pintura, os pincéis e as cores vão funcionar de acordo com o desenho e as cores que eu precisar.

5) O que é uma classe abstrata? (assinale as afirmações verdadeiras)

- a. É o mesmo que uma interface.
- b. Uma classe que, assim como uma interface, possui apenas assinaturas dos métodos.
- c. Uma classe que possui apenas atributos e propriedades, herdados pelas subclasses.

- d. Uma classe que possui implementação e assinatura de métodos e propriedades, mas não pode ser instanciada. ✓
- e. Uma classe que possui implementação e assinatura de métodos e propriedades. Pode ser instanciada com uso de singleton.
- f. Uma classe como outra qualquer, mas não pode implementar interfaces.

6) Qual a finalidade da palavra reservada virtual? Faça uma breve relação deste recurso com conceitos de Orientação à Objetos.

O virtual pode ser utilizado para métodos que têm o mesmo nome mas em classes diferentes, como uma classe que herda outra e tem exatamente o mesmo nome da sua classe base, mas executará uma ação diferente da outra, resumidamente é dizer que ele pode ser sobrescrito em uma classe derivada.

7) Descreva, resumidamente, o que significa cada letra da sigla SOLID.

S - Princípio da responsabilidade única (Single Responsibility Principle)

Uma classe deve ter um, e somente um, motivo para mudar.

O - Princípio Aberto-Fechado (Open-Closed Principle)

Objetos ou entidades devem estar abertos para extensão, mas fechados para modificação.

L - Princípio da substituição de Liskov (Liskov Substitution Principle)

Uma classe derivada deve ser substituível por sua classe base.

I - Princípio da Segregação da Interface (Interface Segregation Principle)

Uma classe não deve ser forçada a implementar interfaces e métodos que não irão utilizar.

D - Princípio da inversão da dependência (Dependency Inversion Principle)

Dependa de abstrações e não de implementações.

8) Descreva, resumidamente, o que é Injeção de Dependência.

É um tipo de Inversão de Controle e significa que uma classe não mais é responsável por criar ou buscar os objetos dos quais depende, serve para desacoplar as classes assim deixando uma independente da outra.

9) Comente sobre como uma API Rest deve ser padronizada.

Utilizar métodos HTTP corretamente, não mudar de estado com o GET, usar criptografia SSL, utilização de substantivos e não verbos e nomes no plural sempre que possível.

10) Considerando ORM, micro-ORM e ADO.Net puro, indique quando e porque utilizar cada um.

ORM's são Mapeadores poderosos de objetos-relacionais e funcionam consultando, criando objetos em tempo de execução entre o código-fonte e o banco de dados, fornece muitos recursos, como armazenamento em cache de objetos, linguagens de consulta orientada a objetos, controle de concorrência etc.. ótimo para aplicações de médio/grande porte ou até mesmo pequeno mas que precisem de algo mais robusto, sem levar tanto performance em consideração.

micro-ORM basicamente são mapeadores mais básicos que faz a criação de objetos com base em consulta no banco, e abstrai muito dos recursos de um ORM normal, ótimo para aplicações de pequeno/médio porte (até grande as vezes) em que necessitem de grande performance.

ADO.Net é um conjunto de classes que vem por padrão no .NET e dispõem serviços de acesso a dados para desenvolvedores do .Net framework, possui uma infra mínima e até alguma pequena funcionalidade próxima ao de um repositório, mas de forma bastante limitada. Ele é sempre uma camada em cima do provedor de dados e por isso é mais lento. Se utilizado de forma errônea pode ser triste, por isso seu uso é recomendado apenas em aplicações de pequeno porte que não necessitem trafegar grande volume de dados e performance.

11) Quanto tempo você levaria para fazer uma funcionalidade com as operações CRUD (criar, recuperar/consultar, editar e excluir)?

Depende da complexidade e do nível de exigência do projeto, hoje existem diversas formas de se fazer um Crud, e existem até mesmo frameworks como o Entity que fazem isso por você apenas em alguns cliques, mas supondo que queiramos um crud do 0 dentro de uma aplicação mais escalável e mais bem feita, uma média de 4 horas.

12) O que é uma sealed class?

O conceito de sealed class é que quando uma classe é definida como uma sealed ela não pode ser herdada.



Prática

Criar uma API Rest que permita realizar as operações CRUD (criar, recuperar/consultar, editar e excluir) de Cursos para turmas de formação do Cast group.

Critérios de Aceite Obrigatórios

1) Banco de Dados

Pode-se utilizar qualquer gerenciador de banco de dados, desde que contenha as informações abaixo.

- Curso
 - Descrição do assunto (obrigatório)
 - Data de início (obrigatório)
 - Data de término (obrigatório)
 - Quantidade de alunos por turma (opcional)
 - Categoria (obrigatório)



- Categoria
 - Código
 - Descrição



Observações:

- a. Apesar de estar estruturada de forma relacional, pode-se utilizar bancos não relacionais.
- b. As categorias são: Comportamental, Programação, Qualidade e Processos.



2) Regras de Negócio

- a. Não será permitida a inclusão de cursos dentro do mesmo período. O sistema deve identificar tal situação e retornar um código de erro e a mensagem:
“Existe(m) curso(s) planejados(s) dentro do período informado.”



- b. Não será permitida a inclusão de cursos com a data de início menor que a data atual.



3) API Rest

Deve-se disponibilizar um endpoint com a interface do Swagger para fácil visualização das operações disponíveis.



Critérios de Aceite Opcionais

1) Testes de Unidade

Crie alguns testes de unidade para garantir o correto funcionamento das unidades da aplicação.

2) Design Patterns

Aplique e explique, através de comentários ou documentação no formato Markdown, algum(uns) padrão(ões) de projeto.



3) Outras Funcionalidades

Crie mais funcionalidades na API, à sua escolha, de forma a demonstrar outros conceitos/técnicas que deseje, lembrando de explicar o que foi feito em documentação no formato Markdown no repositório do projeto.



4) Integração Contínua

Caso o repositório escolhido permita, configure uma integração contínua simples que apenas compile o projeto a cada nova revisão do controlador de versão.



Entrega

O código-fonte, documentos em Markdown e demais artefatos gerados deverão ser disponibilizados em um repositório público.

O endereço do repositório e deverá ser enviado, junto com as questões respondidas deste documento, para os e-mails:

drielli.costa@castgroup.com.br

gustavo.bergamim@castgroup.com.br