Desafio Técnico C#

Objetivo

Resolver pequenos desafios de programação em C# e responder a questões conceituais sobre esses desafios.

Instruções

- O código deve ser armazenado em um repositório público no GitHub e o link deve ser compartilhado através do e-mail: **gente.gestao@portergroup.com.br**; Caso não se sinta confortável em deixar o código aberto pedimos que compartilhe com os seguintes perfis: joaopedrobc e fmpinheiro

- Resolva cada um dos desafios e faça commits frequentes.

- Responda às questões em um arquivo chamado “QUESTIONS.md”.

Desafios

1. Implemente uma função que recebe um número inteiro e retorne uma string com a representação por extenso desse número. Exemplo: 123 -> "cento e vinte e três".

2. Implemente uma função que recebe um array de inteiros e retorne a soma desses números. O array pode ter até 1 milhão de números.

3. Implemente uma função que recebe uma string contendo uma expressão matemática simples (sem parênteses) e retorne o resultado dessa expressão.
   Exemplo: "2 + 3 * 5" -> 17.

4. Implemente uma função que recebe uma lista de objetos e retorne uma nova lista apenas com os objetos únicos, ou seja, sem repetições.

Questões

1. Como você implementou a função que retorna a representação por extenso do número no desafio 1? Quais foram os principais desafios encontrados?

2. Como você lidou com a performance na implementação do desafio 2, considerando que o array pode ter até 1 milhão de números?

3. Como você lidou com os possíveis erros de entrada na implementação do desafio 3, como uma divisão por zero ou uma expressão inválida?

4. Como você implementou a função que remove objetos repetidos na implementação do desafio 4? Quais foram os principais desafios encontrados?

Dicas

- Gostamos de códigos limpos e bem estruturados, fique a vontade para demonstrar seus conhecimentos de padrões de projeto e os princípios do SOLID.

- Um código com testes unitários garantem a qualidade e manutenibilidade, aproveite para fazer uma boa cobertura de testes.

- Códigos simples e funcionais são melhores que códigos complexos e com falhas.

Bom desafio!