Exercícios Teóricos

1. Explique o que é LINQ e como ele se integra ao C#.

R: LINQ é uma forma de consultar dados em C# usando uma sintaxe integrada e consistente com a linguagem.

2. Quais são as diferenças entre a sintaxe de consulta e a sintaxe de método em LINQ?

R: A sintaxe de consulta usa palavras-chave como from e select, enquanto a sintaxe de método usa métodos como Where() e Select().

3. O que faz o método Select em uma consulta LINQ? Dê um exemplo de quando você usaria o Select.

R: Select transforma cada item da sequência. Exemplo: Select(p => p.Nome) para pegar apenas os nomes.

4. Como você pode ordenar os resultados de uma consulta LINQ? Quais métodos são utilizados para ordenação?

R: Para ordenar os resultados de uma consulta LINQ, você pode usar os métodos OrderBy e OrderByDescending. O método OrderBy ordena os elementos em ordem crescente, enquanto OrderByDescending os ordena em ordem decrescente.

5. Explique o que são as operações de agregação em LINQ e dê exemplos de métodos de agregação.

R: As operações de agregação em LINQ são usadas para calcular um valor único a partir de uma coleção de valores. Exemplos de métodos de agregação incluem Sum, Average, Min, Max e Count.

6. Explique o conceito de “Deferred Execution” (Execução Adiada) em LINQ. Como ele afeta o comportamento das consultas?

R: Execução Adiada significa que a execução de uma consulta LINQ não ocorre até que seus resultados sejam iterados. Isso permite que as consultas sejam definidas sem serem executadas imediatamente, o que pode ser útil para otimizar o desempenho e evitar operações desnecessárias.

7. Qual é a diferença entre os métodos First e FirstOrDefault em LINQ? Quando você deve usar cada um?

R: O método First retorna o primeiro elemento de uma sequência e lança uma exceção se a sequência estiver vazia. Já o FirstOrDefault também retorna o primeiro elemento, mas retorna um valor padrão (como null para tipos de referência) se a sequência estiver vazia. Use FirstOrDefault quando houver a possibilidade de a sequência não conter elementos e você quiser evitar exceções.

8. O que são "Anonymous Types" (Tipos Anônimos) em LINQ e como você pode usá-los em consultas?

R: Tipos Anônimos são tipos que não têm um nome explícito e são definidos no momento da execução.

9. Como o operador GroupBy funciona em LINQ e como você pode usá-lo para agrupar dados?

R: O operador GroupBy em LINQ é usado para agrupar elementos de uma sequência com base em uma chave especificada.

10. Explique o conceito de &quot;Query Composition&quot; (Composição de Consultas) em LINQ. Como ele ajuda a criar consultas complexas?

R: Composição de Consultas refere-se à capacidade de construir consultas complexas em LINQ combinando várias operações de consulta.