

1- Introdução

Transcrição

Para o Curso **LINQ: Crie queries poderosas em C# parte 2** é fortemente aconselhável cursar a primeira parte do curso, disponível [aqui \(https://cursos.alura.com.br/course/linq-c-sharp?preRequirementFrom=linq-c-sharp-parte-2\)](https://cursos.alura.com.br/course/linq-c-sharp?preRequirementFrom=linq-c-sharp-parte-2).

No presente curso serão abordados os seguintes pontos:

- **Relatório de Vendas Repaginado:**

Um Relatório de Vendas é repaginado quando é quebrado em páginas para melhorar a apresentação dos resultados. Tradicionalmente os resultados são mostrados de uma única vez, mas na forma repaginada eles são disponibilizados em diferentes páginas.

Nessa etapa também vamos verificar como criar um relatório utilizando diferentes métodos. Na sequência vamos aprender a calcular o número de registros que devem ser pulados para atingir uma determinada página e, por fim, utilizaremos uma função matemática da biblioteca C#, a `Ceiling`.

- **Subconsultas:**

Nesta parte abordaremos o conceito de consulta. Vamos simplificar consultas tornando-as menores e também vamos extrair de uma consulta maior, uma menor. Ainda, veremos sobre a aplicação de subconsultas em vários pontos da consulta LINQ, por exemplo, nas cláusulas `where`, `group by` e `select`.

- **Clientes que compram produtos mais vendidos:**

Neste trecho a proposta é verificar os produtos mais vendidos da loja, para isto, utilizaremos o conceito de projeção de dados e o método `Sum`. Na sequência vamos somar as vendas por produto por meio de um agrupamento. Caso um produto não possua nenhuma venda registrada, vamos nos deparar com um problema, pois chegaremos a um momento no qual ocorrerá uma exceção e aprenderemos a contornar essa situação!

Veremos que algumas expressões se repetem na consulta e acabaremos utilizando um operador que vai permitir a criação de variáveis internas a consulta. No final vamos construir uma consulta, executá-la, pegar seu resultado e utilizá-la em outra consulta.

- **Análise de afinidade**

Nesta parte aprenderemos sobre análise de afinidade ou análise de carrinho. A análise de carrinho mostra, quando fazemos uma compra na Internet, produtos similares aos que já adquirimos. Por exemplo, ao comprar uma impressora produtos como cartuchos são mostrados. Para gerar o relatório de análise de afinidade utilizaremos o `self join`.

- **Execução Adiada x Execução Imediata**

Vamos aprender sobre a diferença entre a execução adiada e a execução imediata. Ao final desta parte veremos como utilizar uma execução imediata trazendo os resultados de uma consulta LINQ para uma lista em memória. Analisaremos a possibilidade de utilizar um `cache` para guardar os dados e não precisar executá-los o tempo inteiro na consulta.

- **LINQ Paralelo**

Veremos uma situação em que a área de marketing solicita para nós a criação de códigos de tipo `QRCode` dentro de uma consulta `LINQ`. Assim, vamos perceber que a execução dessa consulta será um pouco mais demorada, pois transformar uma informação em imagem que representa `QRCode` demora um pouco! Vamos descobrir como medir o tempo de execução da consulta e também como diminuí-lo. Depois, trabalharemos com `stored procedures` dentro de consultas `LINQ` do `Entities`. No final, vamos realizar uma tarefa secundária que agrupará uma consulta em dois campos diferentes.

- **LINQPad**

Na parte final do curso vamos aprender sobre o `LINQPad` que é uma ferramenta já bastante madura e que facilita o trabalho de quem desenvolve consultas com `LINQ`. Veremos que o `LINQPad` permite a execução de códigos `C#` em geral e não apenas códigos que possuam uma consulta `LINQ`. Iniciaremos trabalhando com expressões simples e veremos que é possível abrir arquivos do `C#` dentro do `LINQPad` modificando a extensão para um formato que ele conheça. Aprenderemos a pegar uma consulta com sintaxe de consulta e gerar uma sintaxe de método (ou sintaxe `lambda`). Por fim, vamos gerar uma consulta a partir do `xml` e configuraremos também uma consulta `LINQ` do `Entities`.