

## Populando coleções após a carga da entidade principal

### Transcrição

Queremos filtrar as compras acima de **R\$ 10,00**. Não podemos usar o `Where()` para filtrar o preço da compra, justamente por estarmos acessando a partir de `Produtos`. O `Include()` também não será útil nessa situação. O que faremos é um segundo `SELECT` para filtrar as compras do produto que já buscamos.

Com o objeto `contexto`, chamaremos o método `Entry()` passando a referência de `produto`. Após, pegaremos a coleção da propriedade `Compra` com o método `Collection(p => p.Compras)`. Em seguida faremos uma `Query()`, que filtrará com a condição `Where(c => c.Preco > 10)`. Por último, carregaremos com `Load()` na referência passada no `Entry()`. O código ficará da seguinte maneira:

```
static void Main(string[] args)
{
    using (var contexto = new LojaContext())
    {
        var serviceProvider = contexto.GetInfrastructure<IServiceProvider>();
        var loggerFactory = serviceProvider.GetService<ILoggerFactory>();
        loggerFactory.AddProvider(SqlLoggerProvider.Create());

        var cliente = contexto
            .Clientes
            .Include(c => c.EnderecoDeEntrega)
            .FirstOrDefault();

        Console.WriteLine($"Endereço de entrega: {cliente.EnderecoDeEntrega.Logradouro}");

        var produto = contexto
            .Produtos
            .Where(p => p.Id == 9004)
            .FirstOrDefault();

        contexto.Entry(produto)
            .Collection(p => p.Compras)
            .Query()
            .Where(c => c.Preco > 10)
            .Load();

        Console.WriteLine($"Mostrando as compras do produto {produto.Nome}");
        foreach (var item in produto.Compras)
        {
            Console.WriteLine($"{item}");
        }
    }
}
```

Executaremos a aplicação com "Ctrl + F5". Poderemos ver que foi executado um `SELECT` em `Produtos` no banco de dados, buscando o produto com o `Id` passado no filtro `Where()`. Em seguida foi feito um segundo `SELECT` em `Compras`, filtrando

pelo `Id` do produto passado no `Entry()` e pelo preço. Por fim, serão apresentados os produtos com preço acima de **R\$ 10,00**.

Esse cenário é bastante utilizado quando queremos aplicar filtros em objetos relacionados da aplicação. Esta estratégia é conhecida como **Carregamento Explícito**, onde trazemos só o que nos interessa.