
Curso	Técnico em Desenvolvimento de Sistemas
Aluno	Carina de Paula Mendes
Turma	EUV - Módulo 3
Agenda 6	Desenvolvimento de Sistemas III

UTILIZANDO MICROSOFT MAUI PARA DESENVOLVIMENTO MOBILE

1. Introdução

Neste projeto, realizei a finalização do aplicativo Minhas Compras, desenvolvido com .NET MAUI e SQLite, conforme proposto na Agenda 6.

O objetivo principal deste desafio foi implementar uma nova funcionalidade: o Relatório de Compras por Categoria, que faz parte do Desafio 1 do material didático.

Essa nova função permite organizar os produtos por tipo (como Alimentos, Higiene ou Limpeza) e gerar um relatório com o total gasto em cada categoria, tornando o app mais completo e útil para o usuário.

2. Etapas de Desenvolvimento

2.1. Alteração do Modelo de Dados — Produto.cs

A primeira modificação foi no modelo de dados, onde foi adicionado um novo campo chamado Categoria à classe Produto. Esse campo é essencial para classificar os itens e permitir o agrupamento posterior no relatório.

```
// AGENDA 6 – Campo Categoria (Nova funcionalidade do projeto)
// Adicionado na finalização do projeto para atender ao desafio "Relatório por Categoria".
// Este campo classifica o produto por tipo (Ex: Alimentos, Higiene, Limpeza).
private string _categoria;
7 referências
public string Categoria
{
    get => _categoria;
    set
    {
        if (string.IsNullOrWhiteSpace(value))
            throw new Exception("Por favor, preencher a categoria");
        _categoria = value;
    }
}

public string Categoria { get; set; }
```

Sem essa alteração, o sistema não teria como identificar o tipo de produto para calcular os totais por categoria.

2.2. Inclusão do Campo Categoria no Cadastro — NovoProduto.xaml e NovoProduto.xaml.cs

Em seguida, foi feita a atualização da tela de cadastro de produtos, incluindo um novo campo de entrada Categoria.

Essa adição permite ao usuário registrar, junto com nome, quantidade e preço, o tipo de produto correspondente a Categoria. Assim, ao salvar o item, o aplicativo passa a armazenar essa nova informação no banco de dados SQLite.



NovoProduto.xaml

```
<!-- AGENDA 6 -- Novo campo: Categoria do produto -->
<Label Text="Categoria do Produto" />
<Entry x:Name="txt_categoria" />
```

NovoProduto.xaml.cs

```
private async void ToolbarItem_Clicked(object sender, EventArgs e)
{
    try
    {
        // AGENDA 3 - Criação do objeto Produto com base nos campos do formulário
        // AGENDA 6 - Inclusão do campo Categoria (novo requisito do desafio)
        Produto p = new Produto
        {
            Descricao = txt_descricao.Text, // Agenda 3: campo original
            Categoria = txt_categoria.Text, // Agenda 6: campo novo adicionado
            Quantidade = Convert.ToDouble(txt_quantidade.Text), // Agenda 3
            Preco = Convert.ToDouble(txt_preco.Text) // Agenda 3
        };

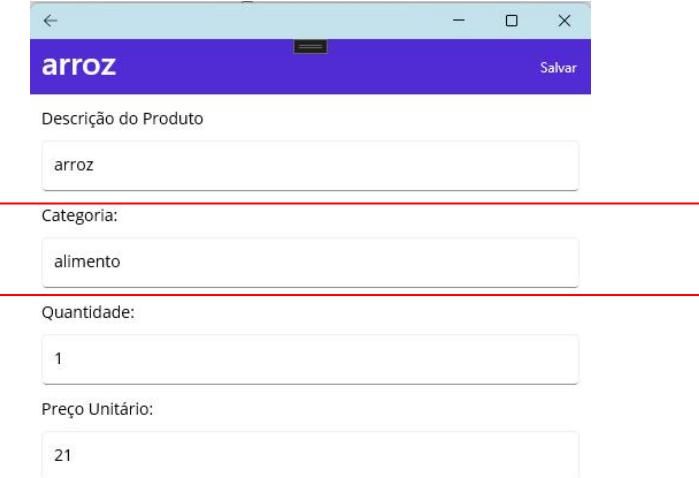
        // AGENDA 3 - Inserção no banco de dados (SQLite)
        await App.Db.Insert(p);

        // Exibe alerta confirmando que o registro foi salvo com sucesso
        await DisplayAlert("Sucesso!", "Registro Inserido", "OK");

        // Retorna para a tela de listagem após o cadastro
        await Navigation.PopAsync();
    }
    catch (Exception ex)
    {
        // Exibe mensagem de erro amigável caso algo dê errado
        await DisplayAlert("Ops", ex.Message, "OK");
    }
}
```

2.3. Edição de Produtos — EditarProduto.xaml e EditarProduto.xaml.cs

Foi implementada também a edição do campo Categoria. Agora, ao editar um produto, o usuário pode visualizar e alterar a categoria cadastrada anteriormente, corrigindo possíveis erros de digitação.



EditarProduto.xaml

```
<!-- AGENDA 6 -- Novo campo Categoria -->
<Label Text="Categoria:" />
<Entry x:Name="txt_categoria" Text="{Binding Categoria}" />
```

EditarProduto.xaml.cs

```
// AGENDA 5 - Evento disparado quando o usuário clica no botão "Salvar" da Toolbar
0 referências
private async void ToolbarItem_Clicked(object sender, EventArgs e)
{
    try
    {
        // AGENDA 5 - Recupera o produto atual (vindo via BindingContext)
        Produto produto_anexado = BindingContext as Produto;

        // AGENDA 5 - Cria um novo objeto Produto com os dados atualizados
        // AGENDA 6 - Inclui a propriedade Categoria (nova no modelo)
        Produto p = new Produto
        {
            Id = produto_anexado.Id,
            Descricao = txt_descricao.Text, // Agenda 5: campo original
            Categoria = txt_categoria.Text, // Agenda 6: novo campo adicionado
            Quantidade = Convert.ToDouble(txt_quantidade.Text),
            Preco = Convert.ToDouble(txt_preco.Text)
        };
    }
}
```

2.4. Exibição dos Produtos com Categorias e Novo Botão Relatório

Na tela de listagem, foram feitas duas melhorias principais: adição de uma nova coluna “Categoria”, exibindo o tipo de cada produto e inserção de um botão “Relatório”, responsável por gerar o resumo dos gastos por categoria.

Essas alterações tornam a interface mais informativa e funcional, permitindo acesso rápido ao relatório diretamente da tela principal.

Minhas Compras

Somar Adicionar Relatorio

Busca de Produtos

ID	Descrição	Categoria	Preço	Qnt	Total
1	arroz	alimento	R\$ 21,00	1	R\$ 21,00
4	feijao	alimento	R\$ 6,00	2	R\$ 12,00
5	sabão de coco	limpeza	R\$ 20,00	1	R\$ 20,00
6	desodorante	higiene	R\$ 8,00	1	R\$ 8,00
7	papel higienico	higiene	R\$ 14,00	1	R\$ 14,00
8	detergente	limpeza	R\$ 2,00	1	R\$ 2,00
9	desinfetante	limpeza	R\$ 20,00	1	R\$ 20,00
10	pao de forma	alimento	R\$ 11,00	1	R\$ 11,00

2.5. Implementação do Relatório — ListaProduto.xaml.cs

No código por trás da listagem, foi implementada a lógica principal do relatório. Foram realizadas as seguintes etapas:

Atualização do método de busca instantânea, permitindo filtrar produtos tanto por nome quanto por categoria.

Criação do método ToolbarItem_Clicked_Relatorio, que: agrupa os produtos por categoria usando GroupBy, calcula o total gasto em cada categoria usando Sum e exibe o resultado em uma janela de mensagem (DisplayAlert) com valores formatados em R\$.

Esse recurso permite que o usuário veja rapidamente quanto gastou em cada tipo de produto.

Minhas Compras

Somar Adicionar Relatorio

Busca de Produtos

alimento

ID	Descrição	Categoria	Preço	Qnt	Total
1	arroz	alimento	R\$ 21,00	1	R\$ 21,00
4	feijao	alimento	R\$ 6,00	2	R\$ 12,00
10	pao de forma	alimento	R\$ 11,00	1	R\$ 11,00

Total dos Produtos
O total é R\$ 44,00

Relatório por Categoria

alimento: R\$ 44,00
limpeza: R\$ 42,00
higiene: R\$ 22,00

3. Resultados Obtidos

Com as modificações, o aplicativo Minhas Compras passou a oferecer:

- Cadastro e edição de produtos com campo de categoria;
- Filtro de busca por nome e categoria;
- Geração de relatório com total de gastos por categoria;
- Exibição organizada e intuitiva dos dados na interface.



ID	Descrição	Categoria	Preço	Qnt	Total
1	arroz	alimento	R\$ 21,00	1	R\$ 21,00
4	feijao	alimento	R\$ 6,00	2	R\$ 12,00
5	sabão de coco	limpeza	R\$ 20,00	1	R\$ 20,00
6	desodorante	higiene	R\$ 8,00	1	R\$ 8,00
7	papel higienico	higiene	R\$ 14,00	1	R\$ 14,00
8	detergente	limpeza	R\$ 2,00	1	R\$ 2,00
9	desinfetante	limpeza	R\$ 20,00	1	R\$ 20,00
10	pao de forma	alimento	R\$ 11,00	1	R\$ 11,00

4. Conclusão

A implementação do Relatório de Compras por Categoria aprimorou significativamente o aplicativo, tornando-o mais completo e útil para o controle de despesas pessoais.

Essa etapa consolidou conhecimentos sobre modelagem de dados, interface em XAML e manipulação de banco de dados SQLite, dentro do ambiente de desenvolvimento .NET MAUI.