

# Busca Dinâmica com .NET MAUI

## 1. Você conhece o .NET MAUI?

Sim, o .NET MAUI (Multi-platform App UI) é uma framework de desenvolvimento de aplicativos que permite criar aplicativos nativos para várias plataformas, como Android, iOS, Windows e macOS, utilizando um único código-base. Ele é uma evolução do Xamarin.Forms, mas com maior foco em desempenho e recursos modernos. O .NET MAUI permite que você crie interfaces gráficas (UIs) utilizando XAML (para descrever a interface) e C# (para a lógica do aplicativo), oferecendo uma forma eficiente de compartilhar código entre diferentes plataformas.

## 2. Conhece sobre mecanismos de buscas?

Sim, mecanismos de busca são ferramentas ou funções que permitem procurar por um termo específico dentro de uma coleção de dados. Esse tipo de mecanismo é amplamente utilizado em aplicativos para filtrar informações com base no que o usuário digita. Por exemplo, ao procurar por um produto em um e-commerce, o mecanismo de busca filtra os produtos que contêm o termo inserido na pesquisa. Esses mecanismos podem ser implementados de diferentes formas. Em muitos aplicativos, a busca é feita em tempo real, ou seja, à medida que o usuário digita o termo, os resultados são atualizados instantaneamente, sem a necessidade de pressionar 'Enter' ou realizar outra ação.

## 3. Como implementar o evento TextChanged em um SearchBar?

O evento TextChanged no SearchBar é um dos eventos mais comuns para implementar buscas dinâmicas. Esse evento é disparado toda vez que o texto no campo de busca (SearchBar) muda, ou seja, quando o usuário digita ou apaga caracteres. A partir desse evento, você pode iniciar a filtragem dos dados em tempo real. A ideia é que, sempre que o texto muda, a aplicação reaja imediatamente e filtre os dados de acordo com o que foi digitado. Isso pode ser feito utilizando listas de dados, onde o mecanismo de busca vai comparar o termo inserido com as informações da lista e exibir os itens que atendem ao critério.

## 4. Como filtrar dados em uma ObservableCollection em tempo real?

Para filtrar dados em uma ObservableCollection (que é uma coleção dinâmica que notifica a interface de usuário sempre que seus dados mudam), você pode criar uma lógica de filtragem que verifica se os itens da coleção atendem ao critério de pesquisa (ex. se o nome do produto contém o termo digitado). A principal vantagem de usar ObservableCollection é que ela notifica automaticamente a interface gráfica sempre que um item é adicionado, removido ou alterado. Quando você filtra os dados, a interface será atualizada de forma dinâmica para refletir os itens filtrados.

## 5. Como implementar um exemplo de utilização do SearchBar para permitir a busca dinâmica de produtos?

A utilização de um SearchBar para busca dinâmica de produtos funciona basicamente da seguinte forma: 1. Captura do texto inserido: Você cria um campo SearchBar onde o usuário digita o termo de busca. 2. Filtragem dos dados: À medida que o texto vai sendo digitado, o evento TextChanged é acionado, permitindo que você execute a filtragem da lista de produtos. O critério de busca pode ser simples, como verificar se o nome do produto contém o termo digitado. 3. Atualização da UI:

Uma vez que os dados são filtrados, a interface gráfica (UI) é automaticamente atualizada, mostrando apenas os produtos que atendem ao critério de busca. Isso é possível porque a `ObservableCollection` mantém a interface sincronizada com as mudanças nos dados. 4. Resultado final: O usuário consegue ver os resultados da pesquisa em tempo real à medida que digita, sem precisar pressionar nenhum botão.