

Convertendo tipos de valores no Binding

Luciana é desenvolvedora na empresa **Alura Car**. Ela precisa implementar uma página Xamarin Forms com um botão "Atualizar Versão".

Ela implementa a *view* e a *view model* para a página, e implementa botão em XAML (incluindo o *command* para o botão), conforme mostra a linha a seguir:

```
<Button Text="Atualizar Versão" Command="{Binding AtualizarVersaoCommand}"></Button>
```

Agora Luciana precisa modificar a aplicação para que o texto do botão "Atualizar Versão" mude de cor de acordo com o valor da propriedade `Urgente`, que existe na *view model* da página:

```
private bool urgente;
public bool Urgente
{
    get { return urgente; }
    private set
    {
        urgente = value;
        OnPropertyChanged();
    }
}
```

Se `Urgente` é verdadeiro, o texto do botão deve ser exibido em **vermelho**. Se for falso, deve ser exibido em **preto**.

O que Luciana deve fazer para implementar o *binding* para a propriedade da cor do texto do botão?

Selezione uma alternativa

- A** Modificar o *binding* na implementação XAML do botão para aplicar o operador ternário do C# (?) no *binding*:

```
<Button
    Text="Atualizar Versão"
    Command="{Binding AtualizarVersaoCommand}"
    TextColor="{Binding Urgente ? Red : Black}"></Button>
```

- B** Adicionar a referência do *conversor* no topo da página XAML, com uma expressão em C# para converter o valor booleano na cor adequada: um operador ternário:

```
<TabbedPage.Resources>
    <ResourceDictionary>
        <converters:UrgenteConverter
            x:Key="urgenteConverter"
            Value="Urgente ? Red : Black"/>
    </ResourceDictionary>
</TabbedPage.Resources>
```

Modificar o *binding* na implementação XAML do botão para aplicar o *conversor*:

```
<Button
    Text="Atualizar Versão"
    Command="{Binding AtualizarVersaoCommand}"
    TextColor="{Binding Urgente, Converter={StaticResource Key=urgenteConverter}}"></pre>
```

C

Criar um *conversor* (uma classe derivada de `IValueConverter`) para traduzir um valor booleano numa cor apropriada:

```
public class UrgenteConverter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
    {
        var urgente = (bool)value;

        if (urgente)
            return Color.Red;
        else
            return Color.Black;
    }
}
```

Adicionar a referência do *conversor* no topo da página XAML:

```
<TabbedPage.Resources>
    <ResourceDictionary>
        <converters:UrgenteConverter x:Key="urgenteConverter"/>
    </ResourceDictionary>
</TabbedPage.Resources>
```

Modificar o *binding* na implementação XAML do botão para aplicar o *conversor*:

```
<Button
    Text="Atualizar Versão"
    Command="{Binding AtualizarVersaoCommand}"
    TextColor="{Binding Urgente, Converter={StaticResource Key=urgenteConverter}}"></pre>
```

D

Criar um *conversor* (uma classe derivada de `IValueConverter`) para traduzir um valor booleano numa cor apropriada:

```
public class UrgenteConverter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
    {
        var urgente = (bool)value;

        if (urgente)
            return Color.Red;
        else
            return Color.Black;
    }
}
```

Anotar a propriedade `Urgente` com um atributo da classe `UrgenteConverter`.

```
private bool urgente;
[UrgenteConverter]
public bool Urgente
{
    get { return urgente; }
    private set
    {
        urgente = value;
        OnPropertyChanged();
    }
}
```

