



Validando a visualização de um botão – Conversores para cores

Na regra de negócios do app, não permitiremos que o botão de salvamento esteja ativo caso os pontos de vida sejam negativos. Definiremos uma propriedade para ser referenciada à view de vínculo através do binding

1. Crie uma propriedade chamada `CadastroHabilitado` na classe `CadastroPersonagemViewModel`

```
public bool CadastroHabilitado
{
    get
    {
        return (PontosVida > 0);
    }
}
```

- Perceba que esta propriedade só tem `get`, pois apenas retornará o `true` ou `false` não sendo possível atribuir valores, já que não temos o `set`.
2. Insira a notificação de mudança da propriedade `CadastroHabilitado` para os pontos de vida

```
public int PontosVida
{
    get => pontosVida;
    set
    {
        pontosVida = value;
        OnPropertyChanged();
        OnPropertyChanged(nameof(CadastroHabilitado));
    }
}
```

3. Abra a View de cadastro dos personagens e atribua `IsEnabled` ao binding do botão de Imagem.

```
<Button Text="Salvar" Command="{Binding SalvarCommand}"
        IsEnabled="{Binding CadastroHabilitado}"></Button>
```

- Realize o teste entrando no menu para cadastrar um novo personagem, como neste caso ele não terá `Id` ainda (será 0) o botão ficará invisível, pois a propriedade retornará `false`.



Validação de Campos

Podemos inserir validações baseadas nas propriedades de uma classe viewModel para habilitar ou desabilitar o botão de gravação.

4. Crie um método que verifica se o nome do personagem está preenchido e se os valores de força e defesa são diferentes de zero, além da propriedade que verifica os pontos de vida.

```
public bool ValidarCampos()
{
    return !string.IsNullOrEmpty(Nome)
        && CadastroHabilitado
        && Forca != 0
        && Defesa != 0;
}
```

5. Procure o construtor e altere o command do botão de salvar para examinar as condições, conforme abaixo

```
public CadastroPersonagemViewModel()
{
    string token = Preferences.Get("UsuarioToken", string.Empty);
    pService = new PersonagemService(token);
    _ = ObterClasses();

    SalvarCommand = new Command(async () => { await SalvarPersonagem(); }, () => ValidarCampos());
    CancelarCommand = new Command(async => CancelarCadastro());
}
```

- É necessário que na linha seguinte ao OnPropertyChanged de cada propriedade envolvida na validação, nós possamos inserir a codificação sinalizada abaixo, para que quando a viewModel identifique que a propriedade foi alterada, possa rodar a validação novamente. Repita o trecho sinalizado nas propriedades PontosVida, Forca e Defesa.

```
public string Nome
{
    get => nome;
    set
    {
        nome = value;
        OnPropertyChanged();
        ((Command)SalvarCommand).ChangeCanExecute();
    }
}
```

- Nesta validação está sendo verificado apenas o nome, quantidade de ponto de vida, força e defesa, ou seja, enquanto estes campos não estiverem preenchidos ou diferente de zero, o botão de salvar não será habilitado. Você pode incluir mais campos na validação observando o que foi feito nas etapas anteriores.



Destacando os personagens

6. Crie uma classe chamada PontosVidaConverter dentro da pasta Converters implementando a interface IValueConverter.

```
public class PontosVidaConverter : IValueConverter
```

16. Clique com CTRL + . (ponto) em IValueConverter e escolha implementar a interface.

7. Com a interface implementada, codifique o método *Convert* que definirá, de acordo com o valor dos pontos de vida, a cor que será retornada. Faça a notação "using Color = Microsoft.Maui.Graphics.Color" no topo do arquivo.

```
public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
{
    ColorTypeConverter converter = new ColorTypeConverter();//using Microsoft.Maui.Graphics.Converters;

    int pontosVida = (int)value;
    if (pontosVida == 100)
        return (Color)converter.ConvertFromInvariantString("SeaGreen");
    else if (pontosVida >= 75)
        return (Color)converter.ConvertFromInvariantString("YellowGreen");
    else if (pontosVida >= 25)
        return (Color)converter.ConvertFromInvariantString("Yellow");
    else if (pontosVida >= 1)
        return (Color)converter.ConvertFromInvariantString("OrangeRed");
    else
        return (Color)converter.ConvertFromInvariantString("Red");
}
```

8. Abra a view de listagem de personagens, faça referência a pasta dos conversores (A) e crie a área de recursos, fazendo a chamada a classe de Conversão de Pontos de Vida em cores (B).

A `xmlns:conv="clr-namespace:AppRpgEtec.Converters"`
`Title="ListagemView">`

B

```
<ContentPage.Resources>
    <ResourceDictionary>
        <conv:PontosVidaConverter x:Key="ColorConvert" />
    </ResourceDictionary>
</ContentPage.Resources>
```



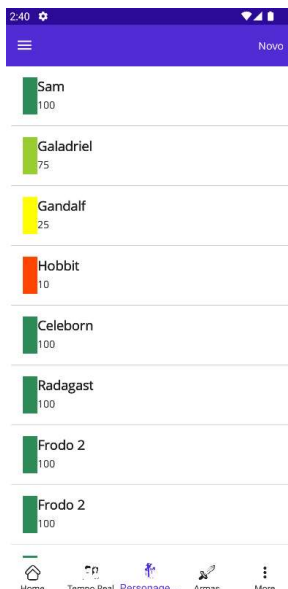
9. Ainda na Listagem de Personagens, procure as labels do nome e pontos de vida dos personagens, modificando o trecho para que tenhamos um Grid e um Box View para exibir a cor sinalizada pelos pontos de vida. Perceba que o Box View conterá a chamada para o Conversor

```
</ViewCell.ContextActions>
<Grid Padding="15">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto"/>
    <ColumnDefinition Width="Auto"/>
    <ColumnDefinition Width="Auto"/>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
  </Grid.RowDefinitions>

  <BoxView Grid.RowSpan="2" HeightRequest="50" WidthRequest="20"
    Color="{Binding PontosVida, Converter={StaticResource ColorConvert}}"/>

  <Label Grid.Row="0" Grid.Column="1" Text="{Binding Nome}" FontSize="18" FontAttributes="Bold"/>
  <Label Grid.Row="1" Grid.Column="1" Text="{Binding PontosVida}" FontSize="14"/>
</Grid>
</ViewCell>
```

- Execute o aplicativo e confirme se os personagens disponíveis estarão de acordo com os pontos de vida dos personagens. Outra possibilidade seria usar essa mesma conversão para a propriedade TextColor da Label



Referências

ListView com Grid: <https://learn.microsoft.com/pt-br/dotnet/maui/user-interface/controls/listview>

BoxView: <https://learn.microsoft.com/en-us/dotnet/maui/user-interface/controls/boxview>