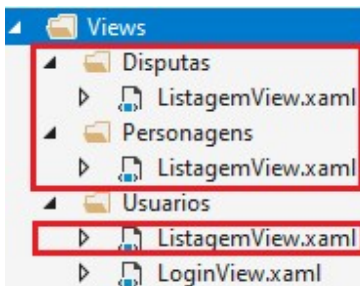




Aula 05 – Menu Flyout em páginas Shell – Layout, Estilos e Cores

1. Atualize a *Solution* com as seguintes pastas (clique com o botão direito na pasta Views → add → new Folder) e *ContentPages* (clique com o botão direito nas pastas → add → new item → content page).



2. Abra a View AppShell.Xaml, remova a tag FlyoutBehaviour (1) e adicione o um template que exibirá no topo da página shell uma imagem e informações do App (2). O Layout (2) encontra-se na sequência deste arquivo.

```
<Shell x:Class="AppRpgEtec.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:AppRpgEtec"
    FlyoutBehavior="Disabled"> 1
</Shell>

<Shell.FlyoutHeaderTemplate> 2
    <DataTemplate>
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="{OnPlatform Android=130, iOS=200}"></RowDefinition>
                <RowDefinition Height="*"></RowDefinition>
                <RowDefinition Height="40"></RowDefinition>
            </Grid.RowDefinitions>
            <FlexLayout Grid.Row="0" Direction="Row" AlignItems="Center" >
                <Frame Padding="0" CornerRadius="40"
                    HeightRequest="80" WidthRequest="80" >
                    <Image Source="has_logo_transparente.svg"/>
                </Frame>
                <VerticalStackLayout Padding="10,0,0,0" Spacing="0">
                    <Label Text="App RPG Etec" FontAttributes="Bold" FontSize="Medium" />
                    <Label Text="Disciplina PAM-II"/>
                </VerticalStackLayout>
            </FlexLayout>
        </Grid>
    </DataTemplate>
</Shell.FlyoutHeaderTemplate>

<ShellContent Title="Home" ContentTemplate="{DataTemplate local:MainPage}" Route="MainPage" />
```



```
<Shell.FlyoutHeaderTemplate>
  <DataTemplate>
    <Grid>
      <Grid.RowDefinitions>
        <RowDefinition Height="{OnPlatform Android=130, iOS=200}"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
        <RowDefinition Height="40"></RowDefinition>
      </Grid.RowDefinitions>
      <FlexLayout Grid.Row="0" Direction="Row" AlignItems="Center" >
        <Frame Padding="0" CornerRadius="40"
          HeightRequest="80" WidthRequest="80" >
          <Image Source="has_logo_transparente.svg" />
        </Frame>
        <VerticalStackLayout Padding="10,0,0,0" Spacing="0">
          <Label Text="App RPG Etec" FontAttributes="Bold" FontSize="Medium" />
          <Label Text="Disciplina PAM-II" />
        </VerticalStackLayout>
      </FlexLayout>
    </Grid>
  </DataTemplate>
</Shell.FlyoutHeaderTemplate>
```

3. Adicione mais referências para as pastas das views que usaremos na próxima etapa.

```
<Shell x:Class="AppRpgEtec.AppShell"
  xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:viewsUsuarios="clr-namespace:AppRpgEtec.Views.Usuarios"
  xmlns:viewsPersonagens="clr-namespace:AppRpgEtec.Views.Personagens"
  xmlns:viewsDisputas="clr-namespace:AppRpgEtec.Views.Disputas"
  xmlns:local="clr-namespace:AppRpgEtec">

  <Shell.FlyoutHeaderTemplate>
    <DataTemplate>
```

4. Adicione uma imagem para o item de menu existe e adicione mais itens de menu para as páginas de listagem

```
  </DataTemplate>
</Shell.FlyoutHeaderTemplate>

  <ShellContent Title="Home" Icon="home.svg"
    ContentTemplate="{DataTemplate local:MainPage}" Route="MainPage" />

  <ShellContent Title="Usuários" Icon="menuusuarios.svg"
    ContentTemplate="{DataTemplate viewsUsuarios:ListagemView}" />

  <ShellContent Title="Personagens" Icon="menupersonagens.svg"
    ContentTemplate="{DataTemplate viewsPersonagens:ListagemView}" />

  <ShellContent Title="Disputas" Icon="menudisputas.svg"
    ContentTemplate="{DataTemplate viewsDisputas:ListagemView}" />

</Shell>
```



5. Insira as imagens da pasta da aula dentro da pasta Resources → Images.
6. Altere a classe app.xaml.cs para que a página de login seja a View inicial.
7. Altere a classe UsuarioViewModel dentro da pasta ViewModels/Usuarios para que a página após o login seja a página que contém o menu. Você terá que fazer using para identificar o caminho em que a página Shell se encontra.

```
public async Task AutenticarUsuario()//Método para autenticar um usuário
{
    try
    {
        Usuario u = new Usuario();
        u.Username = Login;
        u.PasswordString = Senha;

        Usuario uAutenticado = await uService.PostAutenticarUsuarioAsync(u);

        if (!string.IsNullOrEmpty(uAutenticado.Token))
        {
            string mensagem = $"Bem-vindo(a) {uAutenticado.Username}.";

            //Guardando dados do usuário para uso futuro
            Preferences.Set("UsuarioId", uAutenticado.Id);
            Preferences.Set("UsuarioUsername", uAutenticado.Username);
            Preferences.Set("UsuarioPerfil", uAutenticado.Perfil);
            Preferences.Set("UsuarioToken", uAutenticado.Token);

            await Application.Current.MainPage
                .DisplayAlert("Informação", mensagem, "Ok");

            Application.Current.MainPage = new AppShell();
        }
        else
        {
            await Application.Current.MainPage
                .DisplayAlert("Informação", "Dados incorretos :(", "Ok");
        }
    }
}
```

- Execute o aplicativo para realizar os testes necessários.



8. Clique com o direito no projeto (C#) e adicione duas *content pages*: **AboutView.xaml** e **DescriptionView.xaml**.
9. Faça as melhorias abaixo para que possamos criar ícones dos menus no rodapé da página Shell (FlyOutItem - 1), para criar uma página de menu que contenha duas abas (2 - Tab), para criar um item de menu para sair do app que aparecerá apenas no menu lateral (3) e um rodapé para o menu (4 - Shell.FlyOutFooter).

```
</Shell.FlyoutHeaderTemplate>
1 <FlyoutItem FlyoutDisplayOptions="AsMultipleItems">
    <ShellContent Title="Home" Icon="home.svg"
        ContentTemplate="{DataTemplate local:MainPage}" Route="MainPage" />
    <ShellContent Title="Usuários" Icon="menuusuarios.svg"
        ContentTemplate="{DataTemplate viewsUsuarios:ListagemView}" />
    <ShellContent Title="Personagens" Icon="menupersonagens.svg"
        ContentTemplate="{DataTemplate viewsPersonagens:ListagemView}" />
    <ShellContent Title="Disputas" Icon="menudisputas.svg"
        ContentTemplate="{DataTemplate viewsDisputas:ListagemView}" />
2 <Tab Title="Info" Route="info" Icon="info.svg">
    <ShellContent Title="Sobre" Route="sobre" ContentTemplate="{DataTemplate local>AboutView}" />
    <ShellContent Title="Descrição" Route="desc" ContentTemplate="{DataTemplate local:DescriptionView}" />
</Tab>
3 </FlyoutItem>
<ShellContent Title="Sair" Icon="Exit.png" ContentTemplate="{DataTemplate viewsUsuarios:LoginView}"
    Shell.FlyoutBehavior="Disabled"/>
4 <Shell.FlyoutFooter>
    <Label x:Name="lblLogin" TextColor="Black" FontAttributes="Bold" HorizontalOptions="Center" />
</Shell.FlyoutFooter>
</Shell>
```

- Observe que o item de menu “sair” possui uma propriedade Shell.FlyoutBehaviour com valor false, para que quando o usuário clique em sair o menu não mais fique disponível.



10. Entre na parte de código da página AppShell.xaml (F7) para recuperar dados do usuário da Preference e atribuir ao label que adicionamos no design

```
namespace AppRpgEtec;
```

```
5 references
```

```
public partial class AppShell : Shell
```

```
{
```

```
1 reference
```

```
public AppShell()
```

```
{
```

```
InitializeComponent();
```

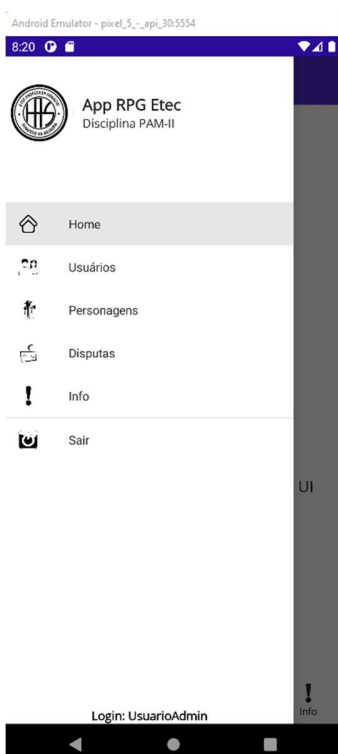
```
string login = Preferences.Get("UsuarioUsername", string.Empty);
```

```
lblLogin.Text = $"Login: {login}";
```

```
}
```

```
}
```

Resultados esperados:





Cores e Estilos

1. Abra o arquivo Resources/Styles/Colors.xaml e crie as tags abaixo. Definiremos neste arquivo cores nomeadas por nomenclaturas que poderemos usar em outras partes do projeto.

```
<ResourceDictionary
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml">

    <Color x:Key="MainColor">#f8cc45</Color>
    <Color x:Key="DarkColor">Black</Color>
    <Color x:Key="DarkTextColor">Black</Color>

    <Color x:Key="Primary">#512BD4</Color>
    <Color x:Key="Secondary">#DFD8F7</Color>
    <Color x:Key="Tertiary">#2B0B98</Color>
```

2. Abra o arquivo Resources/Styles/Styles.xaml e crie o estilo abaixo

```
<ResourceDictionary
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml">

    <Style x:Key="WhiteButtonSmall" TargetType="Button">
        <Setter Property="BackgroundColor" Value="White" />
        <Setter Property="TextColor" Value="Black" />
        <Setter Property="HeightRequest" Value="36" />
        <Setter Property="CornerRadius" Value="18" />
        <Setter Property="FontSize" Value="10" />
        <Setter Property="WidthRequest" Value="80" />
    </Style>
```

3. Abra a view MainPage.Xaml e delete os elementos dentro da tag ContentPage. Insira o layout abaixo

```
<ContentPage.Content>
    <VerticalStackLayout Padding="25" Spacing="30">
        <Frame
            BackgroundColor="{StaticResource MainColor}"
            HeightRequest="{OnPlatform Android=190, iOS=220}"
            Padding="0" HasShadow="false" CornerRadius="10">
            <Grid>
                <Image Source="dotnet_bot.svg" />
                <FlexLayout Direction="Row" Margin="20" AlignItems="Stretch">
                    <Label Text="RPG App" FlexLayout.Grow="1" FontAttributes="Bold"
                        FontSize="20" VerticalTextAlignment="Center"/>
                    <FlexLayout FlexLayout.Grow="2" Direction="Column"
                        AlignItems="End" JustifyContent="Center">
                        <Label Text="Usuários" FontSize="20"/>
                        <Button Style="{StaticResource WhiteButtonSmall}" Text="VER"/>
                    </FlexLayout>
                </FlexLayout>
            </Grid>
        </Frame>
    </VerticalStackLayout>
</ContentPage.Content>
```



A montagem do layout ficará conforme abaixo. Perceba que temos a configuração da cor *MainColor*, inserida no arquivo de cores sendo mencionada no *BackgroundColor* através do *StaticResource*. Perceba também que o botão presente no layout está recebendo o estilo *WhiteButtonSmall* configurado no arquivo de estilos. Execute o aplicativo para testar as alterações.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppRpgEtec.MainPage">
    ...
    <ContentPage.Content>
        <VerticalStackLayout Padding="25" Spacing="30">
            <Frame
                BackgroundColor="{StaticResource MainColor}"
                HeightRequest="{OnPlatform Android=190, iOS=220}"
                Padding="0" HasShadow="false" CornerRadius="10">
                <Grid>
                    <Image Source="dotnet_bot.svg"/>
                    <FlexLayout Direction="Row" Margin="20" AlignItems="Stretch">
                        <Label Text="RPG App" FlexLayout.Grow="1" FontAttributes="Bold"
                            FontSize="20" VerticalTextAlignment="Center"/>
                        <FlexLayout FlexLayout.Grow="2" Direction="Column"
                            AlignItems="End" JustifyContent="Center">
                            <Label Text="Usuários" FontSize="20"/>
                            <Button Style="{StaticResource WhiteButtonSmall}" Text="VER"/>
                        </FlexLayout>
                    </FlexLayout>
                </Grid>
            </Frame>
        </VerticalStackLayout>
    </ContentPage.Content>
</ContentPage>
```

4. Insira as tags abaixo antes de "*ContentPage.Content*" para testarmos mais uma forma declarar recursos de layout.

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppRpgEtec.MainPage">
    ...
    <ContentPage.Resources>
        <ResourceDictionary>
            <Style x:Key="FrameOptions" TargetType="Frame">
                <Setter Property="Margin" Value="10" />
                <Setter Property="HasShadow" Value="True" />
            </Style>
        </ResourceDictionary>
    </ContentPage.Resources>
    <ContentPage.Content>
```



5. Insira as tags abaixo do Frame mais externo

```
<Label Text="Este App oferece recursos para" FontAttributes="Bold" FontSize="Medium"/>
  <ScrollView Orientation="Horizontal">
    <HorizontalStackLayout>
      <Frame Style="{StaticResource FrameOptions}" Padding="0"
HeightRequest="{OnPlatform Android=150, iOS=140}"
      WidthRequest="{OnPlatform Android=120, iOS=140}">
        <VerticalStackLayout Spacing="0" VerticalOptions="Center">
          <Image Source="menupersonagens.svg"/>
          <Label Text="Personagens" FontSize="Small"
HorizontalTextAlignment="Center" Margin="10,0"/>
        </VerticalStackLayout>
      </Frame>
      <Frame Style="{StaticResource FrameOptions}" Padding="0"
HeightRequest="{OnPlatform Android=150, iOS=140}"
      WidthRequest="{OnPlatform Android=120, iOS=140}">
        <VerticalStackLayout Spacing="30" VerticalOptions="Center">
          <Image Source="menuarmas.png"/>
          <Label Text="Armas" FontSize="Small" HorizontalTextAlignment="Center"
Margin="10,0"/>
        </VerticalStackLayout>
      </Frame>
      <Frame Style="{StaticResource FrameOptions}" Padding="0"
HeightRequest="{OnPlatform Android=150, iOS=140}"
      WidthRequest="{OnPlatform Android=120, iOS=140}">
        <VerticalStackLayout Spacing="30" VerticalOptions="Center">
          <Image Source="menudisputas.png"/>
          <Label Text="Disputas" FontSize="Small"
HorizontalTextAlignment="Center" Margin="10,0"/>
        </VerticalStackLayout>
      </Frame>
    </HorizontalStackLayout>
  </ScrollView>

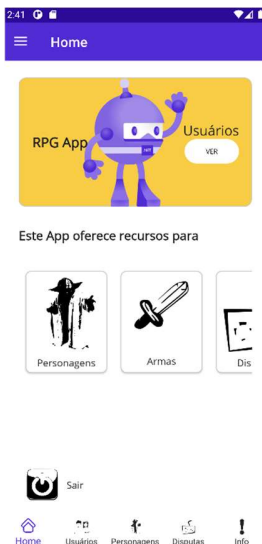
  <HorizontalStackLayout Margin="10,100">
    <Image Source="exit.svg" WidthRequest="50"/>
    <Label Text="Sair" Margin="10,0,0,0" VerticalTextAlignment="Center" />
  </HorizontalStackLayout>
```




O layout deverá ficar conforme abaixo. Execute o aplicativo para testar

```
</FlexLayout>
</Grid>
</Frame>

<Label Text="Este App oferece recursos para" FontAttributes="Bold" FontSize="Medium"/>
<ScrollView Orientation="Horizontal">
  <HorizontalStackLayout>
    <Frame Style="{StaticResource FrameOptions}" Padding="0" HeightRequest="{OnPlatform Android=150, iOS=140}"
      WidthRequest="{OnPlatform Android=120, iOS=140}">
      <VerticalStackLayout Spacing="0" VerticalOptions="Center">
        <Image Source="menupersonagens.svg"/>
        <Label Text="Personagens" FontSize="Small" HorizontalTextAlignment="Center" Margin="10,0"/>
      </VerticalStackLayout>
    </Frame>
    <Frame Style="{StaticResource FrameOptions}" Padding="0" HeightRequest="{OnPlatform Android=150, iOS=140}"
      WidthRequest="{OnPlatform Android=120, iOS=140}">
      <VerticalStackLayout Spacing="30" VerticalOptions="Center">
        <Image Source="menuarmas.png"/>
        <Label Text="Armas" FontSize="Small" HorizontalTextAlignment="Center" Margin="10,0"/>
      </VerticalStackLayout>
    </Frame>
    <Frame Style="{StaticResource FrameOptions}" Padding="0" HeightRequest="{OnPlatform Android=150, iOS=140}"
      WidthRequest="{OnPlatform Android=120, iOS=140}">
      <VerticalStackLayout Spacing="0" VerticalOptions="Center">
        <Image Source="menudisputas.png"/>
        <Label Text="Disputas" FontSize="Small" HorizontalTextAlignment="Center" Margin="10,0"/>
      </VerticalStackLayout>
    </Frame>
  </HorizontalStackLayout>
</ScrollView>
<HorizontalStackLayout Margin="10,100">
  <Image Source="exit.svg" WidthRequest="50"/>
  <Label Text="Sair" Margin="10,0,0,0" VerticalTextAlignment="Center" />
</HorizontalStackLayout>
</VerticalStackLayout>
</ContentPage.Content>
</ContentPage>
```



Recomendações de estudos para Criação de layout, estilos e cores – Canal dotNET

Documentação: <https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/?view=net-maui-7.0>

Parte I https://www.youtube.com/watch?v=x_sNPEwS3kA

Parte II <https://www.youtube.com/watch?v=k3oediCzZvs>