



### Aula 13 - Carregando imagem Salva do Azure BlobStorage no App

1. Crie uma pasta chamada **Converters** e dentro desta pasta, crie uma classe chamada **ByteArrayToImageSourceConverter**, implemente a interface sugerida e realize a programação no método *Convert* a seguir

```
public class ByteArrayToImageSourceConverter : IValueConverter
{
    0 references
    public object Convert(object value, Type targetType,
        object parameter, System.Globalization.CultureInfo culture)
    {
        ImageSource retSource = null;
        if (value != null)
        {
            byte[] imageAsBytes = (byte[])value;
            retSource = ImageSource.FromStream(() => new MemoryStream(imageAsBytes));
        }
        return retSource;
    }

    0 references
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {
        throw new NotImplementedException();
    }
}
```

- Exigirá o using *System.IO* e *System.Globalization*;
2. Abra a view *ImagemUsuarioView*, insirindo o namespace que vai referenciar a classe recém-criada e o recurso que será usado

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppRpgEtec.Views.Usuarios.ImagemUsuarioView"
    xmlns:conv="clr-namespace:AppRpgEtec.Converters"
    Title="ImagemUsuarioView">
    <ScrollView>
```

3. Adicione o recurso do conversor dentro da seção de recursos da content page antes deste primeiro *ScrollView* que aparece no print anterior.

```
<ContentPage.Resources>
    <ResourceDictionary>
        <conv:ByteArrayToImageSourceConverter x:Key="ByteArrayToImage" />
    </ResourceDictionary>
</ContentPage.Resources>
```



4. Troque ou altere o objeto Image que existia pelo sinalizado abaixo, pois existindo um conversor, podemos fazer o binding da viewModel diretamente da propriedade Foto, já que em tempo de execução as informações do array de bytes da imagem serão transformadas em uma imagem visualizável.

```
</ScrollView>
<Image Source="{Binding FonteImagem}" Margin="10"/>
<Image WidthRequest="400" HeightRequest="400" Margin="20"
    Source="{Binding Foto, Converter={StaticResource ByteArrayToImage}}" />
</VerticalStackLayout>
```

5. Abra a classe *ImagemUsuarioViewModel* e crie o método que vai buscar a foto do usuário através da classe de serviço.

```
public async void CarregarUsuarioAzure()
{
    try
    {
        int usuarioId = Preferences.Get("UsuarioId", 0);
        string filename = $"{usuarioId}.jpg";
        var blobClient = new BlobClient(conexaoAzureStorage, container, filename);

        if (blobClient.Exists())
        {
            Byte[] fileBytes;

            using (MemoryStream ms = new MemoryStream())
            {
                blobClient.OpenRead().CopyTo(ms);
                fileBytes = ms.ToArray();
            }
            Foto = fileBytes;
        }
    }
    catch (Exception ex)
    {
        await Application.Current.MainPage
            .DisplayAlert("Ops", ex.Message + " Detalhes: " + ex.InnerException, "Ok");
    }
}
```

6. Faça a chamada do método no construtor e execute o app para testar.

```
public ImagemUsuarioViewModel()
{
    string token = Preferences.Get("UsuarioToken", string.Empty);
    uService = new UsuarioService(token);

    FotografarCommand = new Command(Fotografar);
    SalvarImagemCommand = new Command(SalvarImagemAzure);
    AbrirGaleriaCommand = new Command(AbrirGaleria);
    CarregarUsuarioAzure();
}
```



- Resultado esperado: View carregando exibindo a imagem salva



7. Clique com o botão direito na pasta ViewModels e crie uma classe chamada **AppShellViewModel.cs**, herdando da classe BaseViewModel.

```
public class AppShellViewModel : BaseViewModel
```

8. Insira o atributo de serviço e o construtor conforme o código abaixo dentro da classe para que tenhamos todos os procedimentos para poder buscar os dados do usuário através da API e principalmente a foto cadastrada.

```
private UsuarioService uService;  
public AppShellViewModel()  
{  
    string token = Preferences.Get("UsuarioToken", string.Empty);  
    uService = new UsuarioService(token);  
  
    CarregarUsuarioAzure();  
}
```



9. Crie um atributo/propriedade para a foto e o método para trazer informações do usuário da API.

```
private byte[] foto;
public byte[] Foto
{
    get => foto;
    set
    {
        foto = value;
        OnPropertyChanged();
    }
}
```

10. Use o mesmo método de carregar de carregamento através do armazenamento do Azure

```
public async void CarregarUsuarioAzure()
{
    try
    {
        int usuarioId = Preferences.Get("UsuarioId", 0);
        string filename = $"{usuarioId}.jpg";
        var blobClient = new BlobClient(conexaoAzureStorage, container, filename);

        if (blobClient.Exists())
        {
            Byte[] fileBytes;

            using (MemoryStream ms = new MemoryStream())
            {
                blobClient.OpenRead().CopyTo(ms);
                fileBytes = ms.ToArray();
            }
            Foto = fileBytes;
        }
    }
    catch (Exception ex)
    {
        await Application.Current.MainPage
            .DisplayAlert("Ops", ex.Message + " Detalhes: " + ex.InnerException, "Ok");
    }
}
```



11. Abra a parte de código da view AppShell (AppShell.xaml.cs), declarando a viewModel recém-criada (1) e atribuindo a viewModel como contexto da View

```
public partial class AppShell : Shell
{
    1 AppShellViewModel viewModel;
    1 reference
    public AppShell()
    {
        InitializeComponent();

        2 viewModel = new AppShellViewModel();
        BindingContext = viewModel;

        string login = Preferences.Get("UsuarioUsername", string.Empty);
        lblLogin.Text = $"Login: {login}";
    }
}
```

12. Abra o design da view AppShell e faça referência a pasta das classes de conversão (1), declare o método de conversão de array de bytes para imagem apelidando através da propriedade Key (2) e altere o objeto Image para que ele use a conversão para exibir a imagem através do Binding.

```
1 xmlns:conv="clr-namespace:AppRpgEtec.Converters"
  xmlns:local="clr-namespace:AppRpgEtec">

2 <Shell.Resources>
    <ResourceDictionary>
        <conv:ByteArrayToImageSourceConverter x:Key="ByteArrayToImage" />
    </ResourceDictionary>
</Shell.Resources>

<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="-20" CornerRadius="40" HeightRequest="80" WidthRequest="80" >
                    3 <Image Source="{Binding Foto, Converter={StaticResource ByteArrayToImage}}" />
                </Frame>
            </FlexLayout>
        </Grid>
    </DataTemplate>
</Shell.FlyoutHeaderTemplate>
```

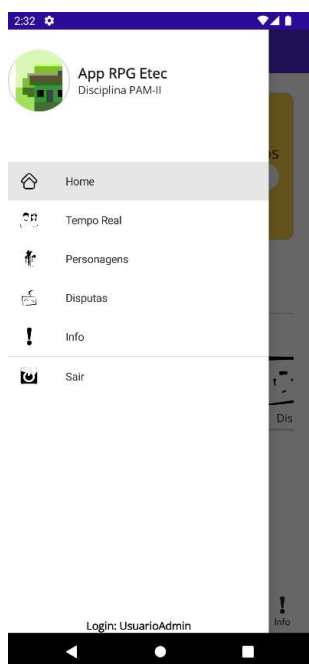


ETEC PROF. HORÁCIO  
AUGUSTO DA SILVEIRA

## PROGRAMAÇÃO DE APLICATIVOS MOBILE – II – 2025/2

Luiz Fernando Souza / Eliane Marion

13. Execute o app e confirme que a imagem que aparecerá no menu será a que foi enviada através do dispositivo



Referências: <https://learn.microsoft.com/pt-br/dotnet/maui/platform-integration/device-media/picker?view=net-maui-8.0&tabs=android>

Salvamento via API/MVC: <https://renicius-pagotto.medium.com/upload-de-arquivos-com-o-azure-blob-storage-76d86710ae3>





---

### **Apenas para ciência**

14. Para carregar a imagem através de uma API se ela estive guardada no banco de dados, poderíamos usar o método a seguir:

```
public async void CarregarUsuario()
{
    try
    {
        int usuarioId = Preferences.Get("UsuarioId", 0);
        Usuario u = await uService.GetUsuarioAsync(usuarioId);

        Foto = u.Foto;
    }
    catch (Exception ex)
    {
        await Application.Current.MainPage
            .DisplayAlert("Ops", ex.Message + " Detalhes: " + ex.InnerException, "Ok");
    }
}
```

15. Adaptando o construtor para chamada do método conforme sinalizado abaixo

```
public ImagemUsuarioViewModel()
{
    string token = Preferences.Get("UsuarioToken", string.Empty);
    uService = new UsuarioService(token);

    AbrirGaleriaCommand = new Command(AbrirGaleria);
    SalvarImagemCommand = new Command(SalvarImagem);
    FotografarCommand = new Command(Fotografar);

    CarregarUsuario();
}
```