ETEC PROF. HORÁCIO AUGUSTO DA SILVEIRA

PROGRAMAÇÃO DE APLICATIVOS MOBILE – II – 2024/1

Luiz Fernando Souza

Visual Studio para desenvolvimento Mobile

O desenvolvimento mobile para este semestre será realizado através do Visual Studio, com o .Net Maui (.NET Multi-Plataform App UI) evolução do Xamarin Forms, plataforma incorporada ao Visual Studio e que permite o desenvolvimento multiplataforma, com o ponto positivo da criação de uma única Solution podendo conter vários projetos de classes que podem ser executados no Android, IOS e Windows, utilizando-se de uma mesma estrutura, além de customizações de interface que cada sistema operacional oferece.

O que era o Xamarim.Forms

Xamarin.Forms é um kit de ferramentas de interface do usuário multiplataforma que permite que os desenvolvedores criem, com eficiência, layouts de interface de usuário nativos que possam ser compartilhados entre iOS, Android e aplicativos da Plataforma Universal do Windows.

Estrutura do Front-end .NET MAUI

Feita em XAML (Linguagem de marcação de aplicativo extensivo) – Linguagem declarativa baseada em XML desenvolvida pela Microsoft para criação de Layouts. É facilmente integrável com arquiteturas de aplicativo populares como o MVVM.

Controles XAML: BoxView, ListView, Button, Label, Maps, Image Button Search Bar, Progress Bar, Slidesr, etc.

Instalação do Visual Studio

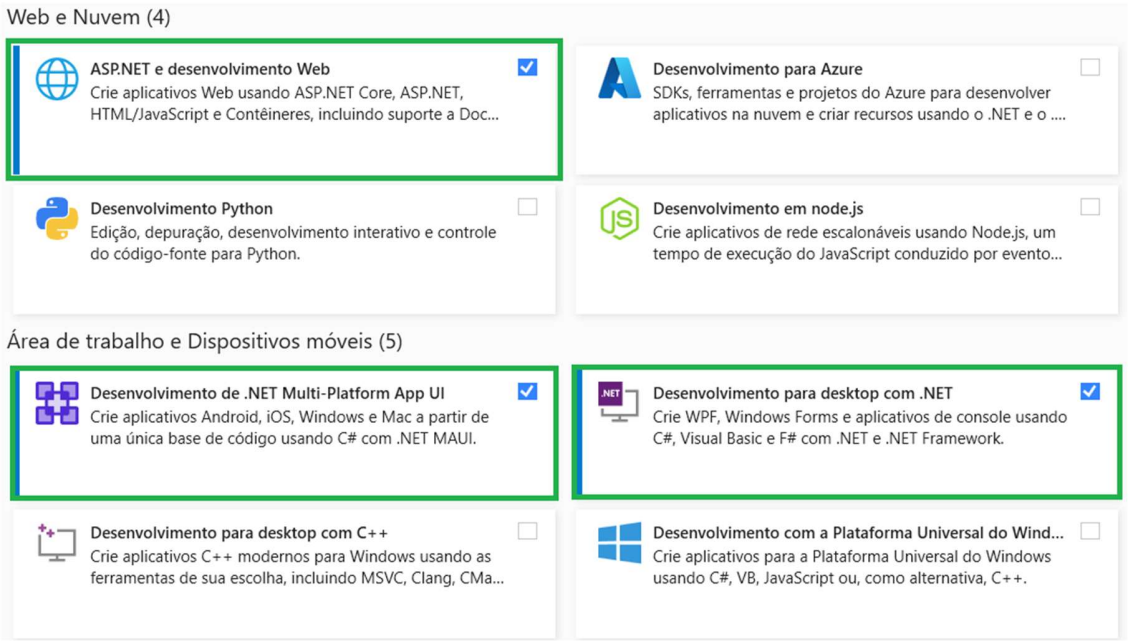
1. Acessar a página de Download do Visual Studio no link:

   https://visualstudio.microsoft.com/pt-br/downloads

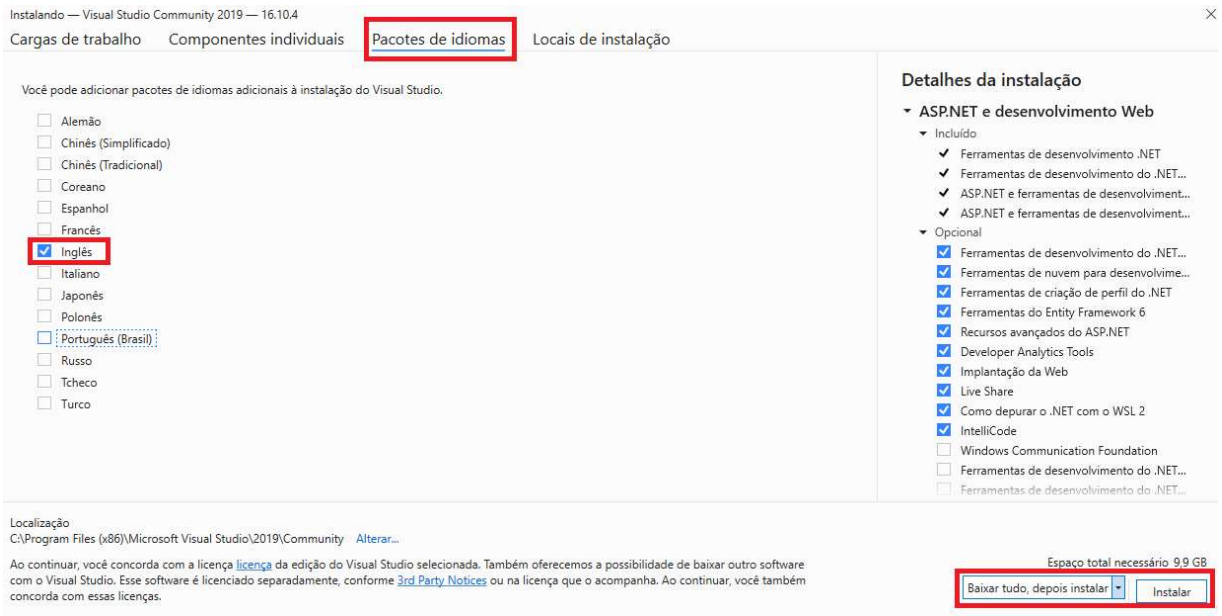
2. Desça um pouco a página até a parte de Downloads e faça a escolha para baixar a versão Comunidade (Community). Esta versão permite logar com o e-mail institucional. Se você já tem o Visual Studio 2019 instalado, procure no Windows por “Visual Studio Installer”, execute o programa e verifique se as configurações descritas na etapa 3 estão selecionadas, se alguma não estiver, basta selecionar e executar.

- O download se iniciará automaticamente. Verifique se o seu navegador não está bloqueando pop-ups.
- Esse arquivo se trata de um instalador online que gerencia o que será instalado ou atualizando e é onde escolhemos algumas configurações que falaremos adiante.

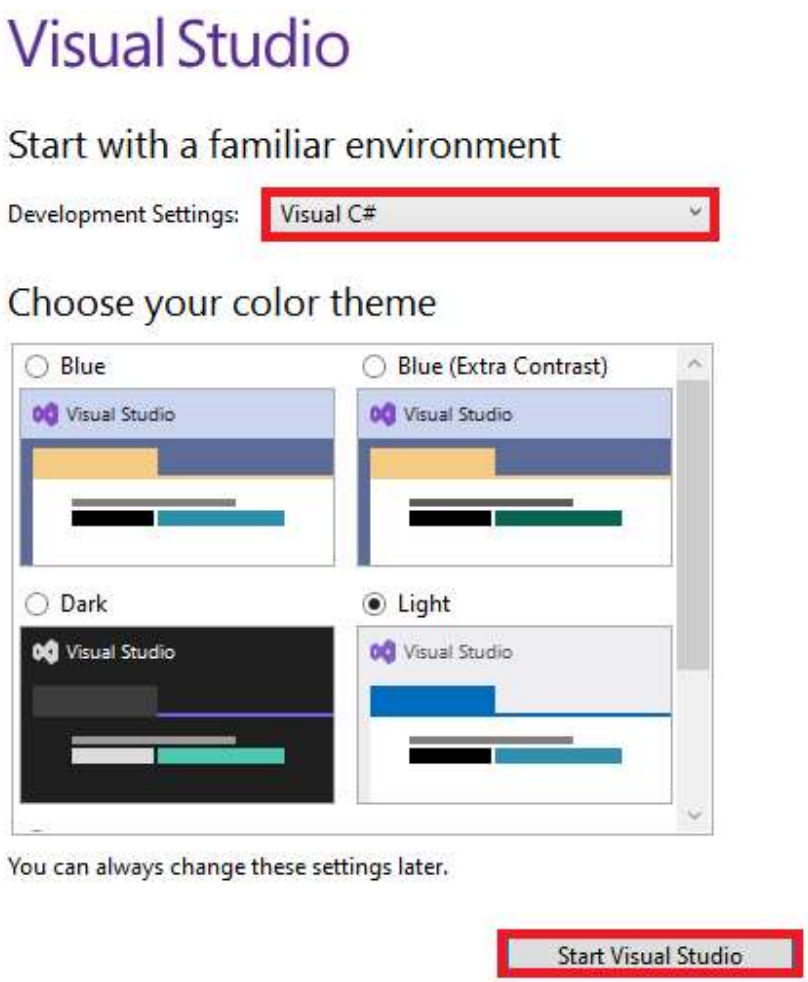
3. Ao Executar o arquivo baixado, a tela a seguir aparecerá e selecionaremos os seguintes itens no menu “Cargas de Trabalho”.



4. Selecione a aba “pacote de idiomas”, mantenha apenas a opção “Inglês” selecionada. Para prosseguir a instalação selecione a opção “Baixar tudo e instalar” e mantenha seu computador conectado a internet.



5. Quando executar o Visual Studio pela primeira vez, aparecerá uma caixa de seleção para escolher a linguagem de programação padrão. Você deve escolher o C#.



ETEC PROF. HORÁCIO AUGUSTO DA SILVEIRA

PROGRAMAÇÃO DE APLICATIVOS MOBILE – II – 2024/1

Luiz Fernando Souza

Atividade Prática

- Configuração para execução no próprio aparelho ou no emulador

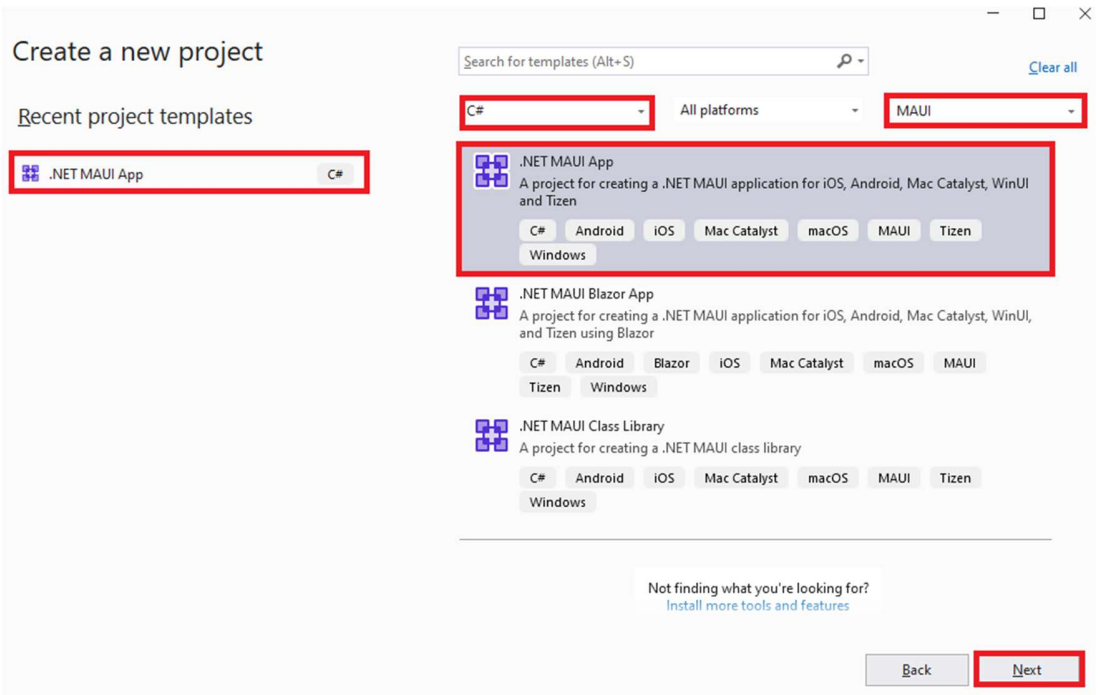
- Criação do primeiro projeto .NET MAUI

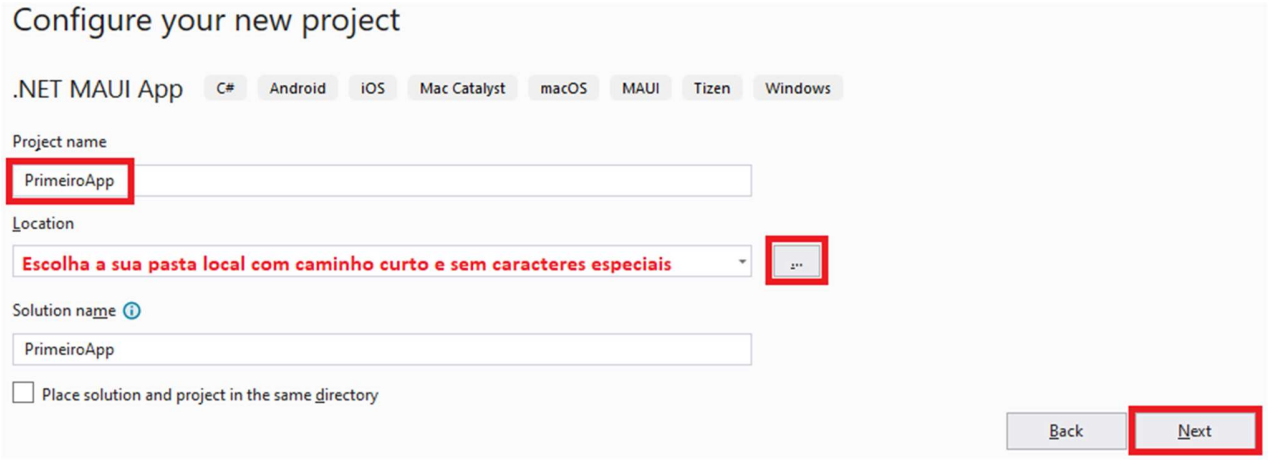
- Entendimento dos elementos XAML para a interface do Aplicativo.

- Aplicativo básico para sala de Aula:

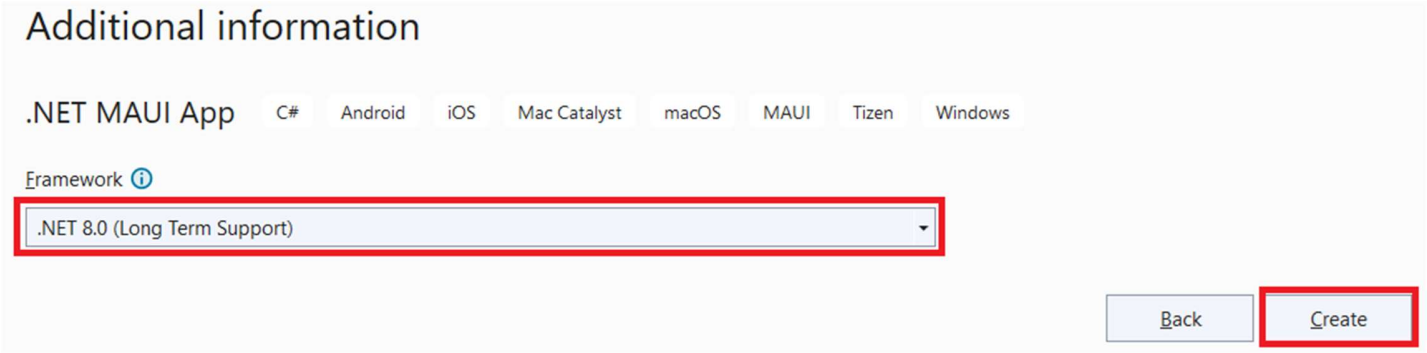
Primeiro Aplicativo:

Programação de um Aplicativo que exiba uma Label e Botão e exibindo o número de vezes que o usuário clicou no botão e demais acréscimos.



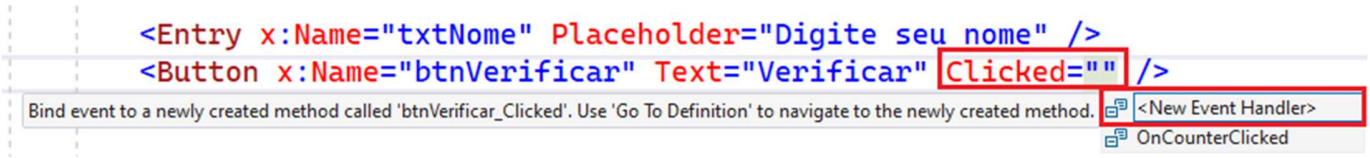


Escolher a versão do .NET e clique em criar.



- Para inicialização do emulador, navegue até o menu Tools → Android → Android Device Manager, clicando no botão Iniciar no dispositivo existente.
- ATENÇÃO: O emulador será iniciado uma vez durante a aula para que possamos executar o projeto diversas vezes. Não feche o emulador.
- Para executar o projeto selecione a triângulo ao lado botão Play e selecione o emulador iniciado. Agora sim, você poderá clicar no botão play e conferir a execução do aplicativo.

1. Abra a ToolBox ao lado esquerdo, e adicione mais controles na View chamada MainPage.Xaml. Ao digitar a propriedade Clicked, dentro das aspas duplas, selecione a opção sinalizada.



2. Toda view (arquivo .xaml) possuirá um arquivo de código C#, com extensão .xaml.cs. Abra (F7) o arquivo de código MainPage.Xaml.cs e perceba que o evento clicando na opção anterior na View, foi criado na parte de código. Realize a programação dentro do evento conforme a seguir

```
private void btnVerificar_Clicked(object sender, EventArgs e)
{
    string texto = $"O nome tem {txtNome.Text.Length} caracteres";

    DisplayAlert("Mensagem", texto, "Ok");
}
```

3. Arraste mais um botão e crie o evento Cliecked

```
<Button x:Name="btnLimpar" Text="Limpar" />
```

4. Realize a programação do evento, na parte de código da View

```
private async void btnLimpar_Clicked(object sender, EventArgs e)
{
    if(await DisplayAlert("Pergunta", "Deseja realmente limpar a tela", "Yes", "No"))
    {
        txtNome.Text = string.Empty;
    }
}
```

5. Insira os controles abaixo para verificar os dias vividos.

```
<Label Text="Selecione sua data de nascimento" />
<DatePicker x:Name="txtDtNascimento"  />
<Button x:Name="btnVerificarDiasVividos"
        Text="Verificar Dias Vividos" Clicked="btnVerificarData_Clicked"/>
```

6. Crie o evento para o botão e realize a programação para exibição dos dias vividos

```
private async void btnVerificarData_Clicked(object sender, EventArgs e)
{
    int diasVividos = DateTime.Now.Subtract(txtDtNascimento.Date).Days;

    await Application.Current.MainPage
        .DisplayAlert("Mensagem", $"Você já viveu {diasVividos} dias", "Ok");
}
```

7. Crie mais um botão com o nome a seguir, criando um evento para o mesmo

```
<Button x:Name="btnCalcular" Text="Calcular" />
```

8. Programe a primeira parte do evento coletando os números digitados pelo usuário

```
private async void btnCalcular_Clicked(object sender, EventArgs e)
{
    string n1 = await Application.Current.MainPage
        .DisplayPromptAsync("Mensagem", "Digite o primeiro número", "Ok");

    string n2 = await Application.Current.MainPage
        .DisplayPromptAsync("Mensagem", "Digite o segundo número", "Ok");

    string operacao = await Application.Current.MainPage
        .DisplayActionSheet("Mensagem", "Selecione uma opção",
        "Cencelar", "Somar", "Subtrair", "Multiplicar", "Dividir");
```

9. Realize a segunda parte do evento, coletando qual a operação escolhida pelo usuário e realizando o cálculo.

```
string operacao = await Application.Current.MainPage
    .DisplayActionSheet("Mensagem", "Selecione uma opção",
    "Cencelar", "Somar", "Subtrair", "Multiplicar", "Dividir");
```

```
if(operacao != null)
{
    if (operacao == "Somar")
    {
        int resultado = int.Parse(n1) + int.Parse(n2);

        await Application.Current.MainPage
            .DisplayAlert("Mensagem", $"O resultado é {resultado}","Ok");
    }
    else if(operacao == "Subtrair")
    {
        int resultado = int.Parse(n1) - int.Parse(n2);

        await Application.Current.MainPage
            .DisplayAlert("Mensagem", $"O resultado é {resultado}", "Ok");
    }
    //Realize as demais operações.
}
```