

Macoratti.net Xamarin Android - Criando sua primeira Aplicação

Android : Activity (Conceitos) - II



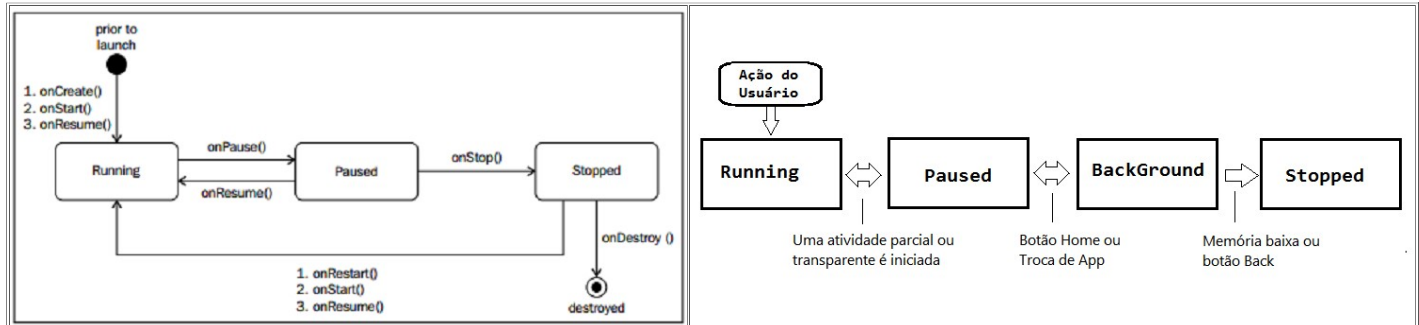
Neste artigo vou mostrar como criar uma aplicação **Android** usando os recursos do **Xamarin** no Visual Studio 2015 abordando os principais conceitos relacionados e o conceito de Activity: **Passando informações entre atividades**.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

Na [primeira parte do artigo](#) criamos nossa primeira aplicação Android usando o Xamarin e o Visual Studio e apresentamos sua estrutura e o conceito de **Activity**.

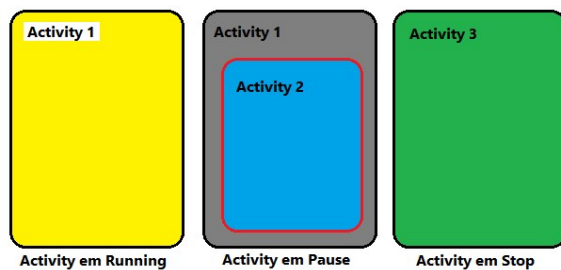
Para recordar temos abaixo dois diagramas do ciclo de vida de uma Atividade ou **Activity** representando os estados que ocorrem durante o ciclo de vida:



Enfatizando as ocorrências relacionadas com cada estado, podemos fazer um pequeno resumo:

Estados	Ocorrências
Running	As atividades que estão em primeiro plano.(<i>foreground</i>) Atividades neste estado tem maior prioridade e só são mortas pelo SO em circunstâncias extremas.
Paused	Atividades pode estar neste estado quando : - O dispositivo esta em repouso - Outra atividade esconde parcialmente a atividade - Uma atividade transparente obstrui a atividade Atividades nestes estado mantêm o seu estado e permanecem anexadas ao gerenciador de janelas. Possuem a segunda prioridade no sistema
Stopped (BackGround)	Uma atividade em <i>segundo plano ou parada</i> esta completamente oculta por outra atividade. Elas tentam manter o estado, mas possuem menor prioridade e assim tem mais probabilidade de serem mortas pelo SO para liberar recursos. Até estar morta, uma atividade poderá ser retomada a qualquer momento sem perda de informações de estado. Se a atividade for morta, e, em seguida, o usuário navegar de volta para ela, ela deve ser reiniciada e restaurada ao seu estado anterior (<i>que esta salvo</i>). Isso é responsabilidade do desenvolvedor.

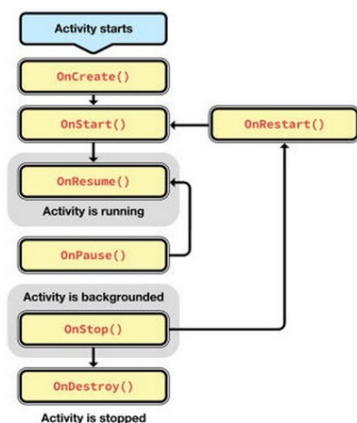
A seguir temos a figura que mostra os estados de uma Activity durante o ciclo de vida da aplicação :



- Ao iniciar a atividade a **Activity 1** esta pronta com o foco e em estado **Running**;
- A seguir a **Activity 2** foi invocada (recebeu o foco) e entra na frente da Activity 1(perdeu o foco) que ainda fica visível. Neste momento a Activity 1 esta no estado **Paused**;
- Depois a **Activity 3** é chamada e oculta totalmente a Activity 1. Neste momento a Activity 1 esta no estado **Stopped**.
- A **Activity 2** também foi encoberta e já terminou o seu trabalho esta no estado **Stopped** e será destruída.

se o usuário retornar para a Activity 1 ela será reiniciada (*OnRestart*) e voltará ao estado **Running**.

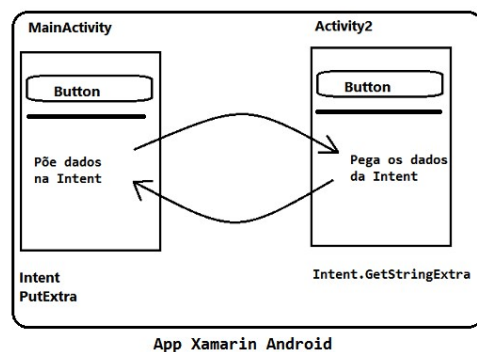
Abaixo temos um diagrama ([obtido da documentação do Xamarin](#)) que mostra os eventos que podem ocorrer durante o ciclo de vida da aplicação:



Neste artigo veremos como passar informações entre atividades distintas.

Vamos usar a nossa aplicação **Android** criada na [primeira parte do artigo](#) onde temos duas atividades:

Abaixo temos uma figura com o diagrama exibindo basicamente o que iremos fazer e quais métodos vamos usar:



Recursos usados:

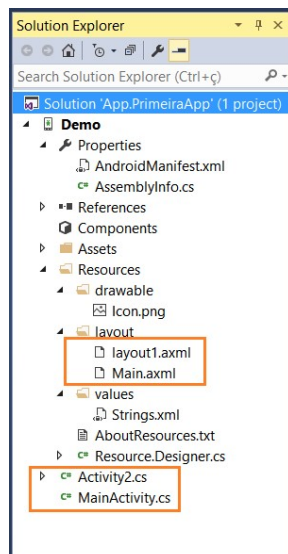
- [Visual Studio Community 2015](#)
- [Xamarin](#)
- Emulador Android virtual ou físico ([veja como emular usando o Vysor](#))

Nota: Baixe e use a versão Community 2015 do VS ela é grátis e é equivalente a versão Professional.

Criando o projeto no VS Community com Xamarin

Abra o projeto **App.PrimeiraApp** o criado na primeira parte do artigo no Visual Studio ou no Xamarin Studio.

A estrutura da nossa solução pode ser vista na figura abaixo:



Observe que temos duas **Activities** ou atividades representadas pelos arquivos : **Activity2.cs** e **MainActivity.cs**

Temos também dois arquivos **.xml** que representam as **views** que estamos exibindo em cada uma das atividades : **Main.xml** e **layout1.xml**

Para passar dados entre as duas atividades vamos usar o conceito de **Intent** do android.

Uma **Intent** ou **intenção** é um conceito abstrato para algum tipo de operação que deverá ser executada no sistema operacional Android.

Intents ou **Intenções** no Android, são estruturas de dados que são objetos de mensagens. Intenções podem solicitar uma operação a ser realizada por algum outro componente no Android e são geralmente usadas para iniciar **Atividades** e **Serviços**.

Todas as intenções são classificados em dois tipos : **Intenções explícitas** e **Intenções implícitas**.

Elas são muitas vezes usadas para iniciar aplicações externas com a intenção de fazer alguma coisa, como fazer uma chamada de telefone, exibir uma página web, ou mapear um endereço.

Como tal, uma **Intent** possui duas informações associadas a ela :

1. Em primeiro lugar, o objetivo da intenção (*fazer uma chamada telefônica*);
2. Em segundo, os dados que a intenção precisa (*como um número de telefone*) para ser executada.

Neste nosso primeiro contato com o conceito veremos um exemplo bem simples. Em outro artigo veremos sua aplicação em um cenário mais complexo e útil.

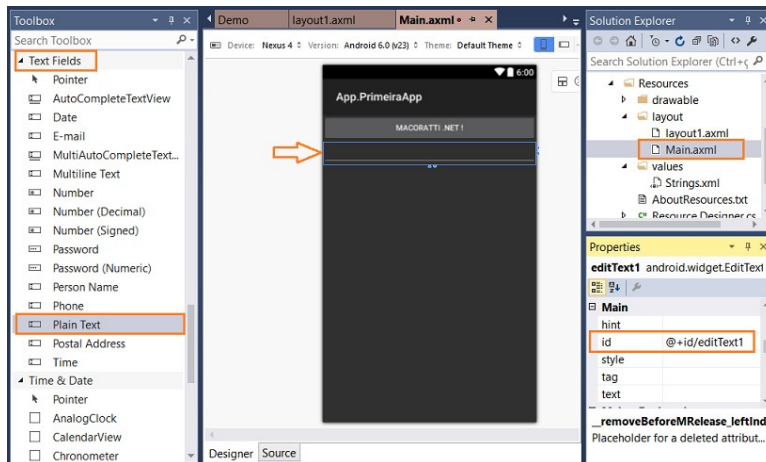
Vamos ter que realizar 3 tarefas tarefas :

1- Criar as duas Views e ajustar o leiaute

Ajustar o leiaute das duas telas representadas pelos arquivos XML : **Main.xml** e **layout1.xml**

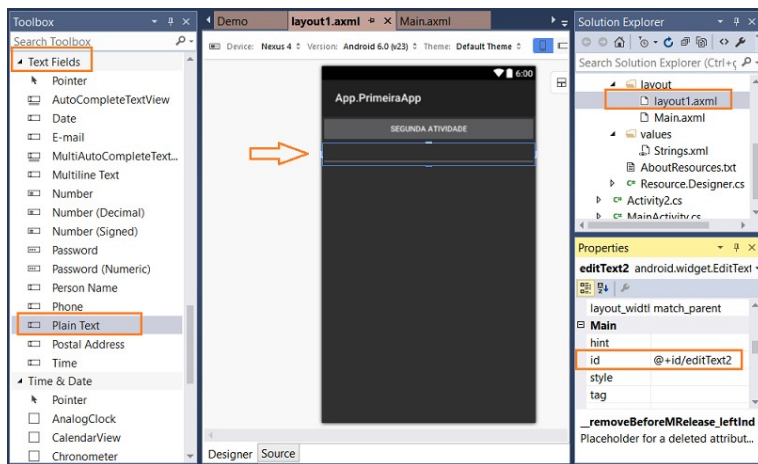
Vamos incluir em cada um destes arquivos a definição de um campo de edição de texto incluindo a partir da **ToolBox** o elemento **PlainText**:

1- Main.xml



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/MyButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/Hello" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText1" />
</LinearLayout>
```

2- layout.xml



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:minWidth="25px"
    android:minHeight="25px">
    <Button
        android:text="Segunda Atividade"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/macButton" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText2" />
</LinearLayout>
```

O objetivo é digitar o texto na primeira atividade (**Main.xml**) e transferir os dados informados em **editText1** para a segunda atividade(**layout.xml**) colocando-os em **editText2**.

2- Colocar os dados na atividade

Abra o arquivo **MainActivity.cs** e inclua o código abaixo:

```
using Android.App;
using Android.OS;
using Android.Widget;

namespace App.PrimeiraApp
{
    [Activity(Label = "App.PrimeiraApp", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        int count = 1;
```

```
protected override void OnCreate(Bundle bundle)
{
    base.OnCreate(bundle);

    // Set our view from the "main" layout resource
    SetContentView(Resource.Layout.Main);

    // Get our button from the layout resource,
    // and attach an event to it
    Button button = FindViewById<Button>(Resource.Id.MyButton);

    button.Click += delegate
    {
        button.Text = string.Format("{0} clique!", count++);
        var atividade2 = new Intent(this, typeof(Activity2 ));
        atividade2.PutExtra("MeusDados", "Dados da atividade Activity2");

        //pega os dados digitados em editText1
        atividade2.PutExtra("nome", FindViewById<EditText>(Resource.Id.editText1).Text);

        StartActivity(atividade2 );
    };
}
}
```

O código em azul cria uma instância da classe **Intent()** e a seguir usamos o método **PutExtra()** para por os dados na **Intent**.

Estamos colocando um texto na variável '**MeusDados**' e estamos obtendo o texto digitado no campo **editText1** na variável '**nome**' usando o método **PutExtra()**.

O método **PutExtra()** é usado para colocar dados estendidos em um **Intent**.

3- Obter os dados da atividade

Agora abra o arquivo **Activity2.cs** e inclua o código abaixo neste arquivo:

```
using Android.App;
using Android.OS;
using Android.Widget;

namespace App.PrimeiraApp
{
    [Activity(Label = "Segunda Atividade")]
    public class Activity2 : Activity
    {
        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            // Create your application here
            SetContentView(Resource.Layout.layout1);

            Button button = FindViewById<Button>(Resource.Id.macButton);

            string texto = Intent.GetStringExtra("MeusDados") ?? "Dados não disponíveis";
            button.Text = texto;

            //pega os dados obtidos na primeira atividade e exibe no TextField
            FindViewById<TextView>(Resource.Id.editText2).Text = Intent.GetStringExtra("nome") ?? "Erro ao obter os dados";

            button.Click += delegate
            {
                StartActivity(typeof(MainActivity));
            };
        }
    }
}
```

Neste código estamos obtendo os dados colocados na atividade **Activity2** usando o método **GetStringExtra()** que pega os dados da **Intent**.

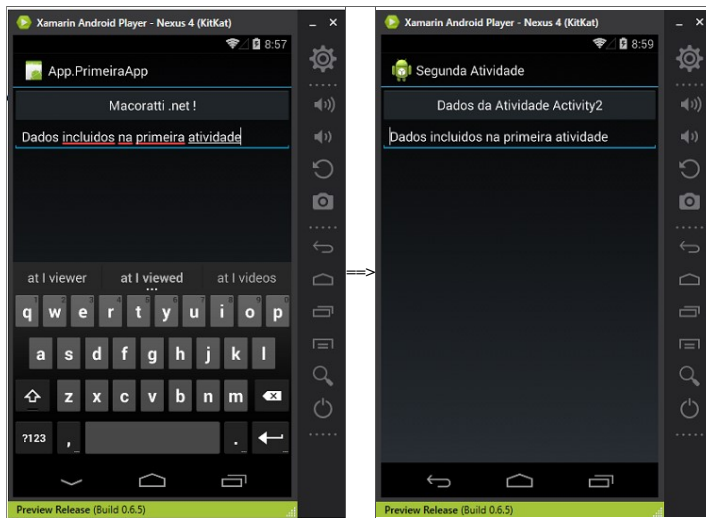
Primeiro estamos obtendo o texto colocado em '**MeusDados**'; se os dados existirem eles serão armazenados na string **texto** e exibidos na propriedade **Text** do controle **Button**.

A seguir estamos obtendo os valores colocados em '**nome**' e colocando os dados no componente **editText2** na segunda atividade.

Executando projeto teremos o seguinte resultado:

1- Na primeira tela temos a atividade **MainActivity** sendo executada e nela pomos os dados na **Intent**, e, também digitamos algum texto no campo **editText1**;

2- Na segunda tela iniciamos a atividade **Activity2** e obtemos os dados da **Intent** e exibimos no controle **Button**. Também obtemos os dados colocados em **editText1** e exibimos em **editText2**:



Como vemos os dados foram transferidos entre as atividades.

Você deve estar se perguntando como transferir objetos complexos entre as atividades ?

Podemos fazer isso usando a serialização e o **Parcelable**, mas isso é assunto para outro artigo.

Na [continuação do artigo](#) veremos como gerenciar o estado da Activity persistindo e restaurando informações do estado.

Pegue o projeto completo aqui : [App.PrimeiraApp 2.zip](#) (sem as referências)

Porque a palavra da cruz é loucura para os que perecem; mas para nós, que somos salvos, é o poder de Deus.
1 Coríntios 1:18

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

Quer migrar para o VB .NET ?

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Básico - Vídeo Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#) NEW

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Super DVD C# - Recursos de aprendizagens e vídeo aulas para C#](#)
- [Seção C# do site Macoratti.net](#)
- [Seção ASP .NET do site Macoratti.net](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW
- [Macoratti.net | Facebook](#)
- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macoratti\) | Twitter](#)
- [Xamarin - Desenvolvimento Multiplataforma com C# ... - Macoratti.net](#)
- [Xamarin - Apresentando Xamarin.Forms - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Criando sua primeira ... - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Anatomia da aplicação - Macoratti.net](#)
- <https://developer.xamarin.com/recipes/android/fundamentals/intent/>
- <https://developer.xamarin.com/recipes/android/fundamentals/activity/>