

Macoratti.net Xamarin Android - Enviando Emails



Neste artigo vou mostrar como podemos enviar emails usando o Visual Studio com Xamarin Android e a linguagem C#.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

E lá vamos nós mais uma vez mostrar como enviar email, desta vez em uma aplicação Android usando o Visual Studio com Xamarin Android e a linguagem C#.

Para enviar email vamos usar criar **Intent** que é um objeto de mensagem que pode ser usado para solicitar uma ação de outro componente de aplicativo. Embora as intents facilitem a comunicação entre componentes de diversos modos, há três casos de uso fundamentais: (retirado do original: <https://developer.android.com/guide/components/intents-filters.html>)

- **Para iniciar uma atividade:**

Uma [Activity](#) representa uma única tela em um aplicativo. É possível iniciar uma nova instância de uma [Activity](#) passando uma [Intent](#) para [startActivity\(\)](#). A [Intent](#) descreve a atividade a iniciar e carrega todos os dados necessários.

Se você deseja receber um resultado da atividade quando ela finalizar, chame [startActivityForResult\(\)](#). Sua atividade recebe o resultado como um objeto [Intent](#) separado no retorno de chamada de [onActivityResult\(\)](#) da atividade. Para saber mais, consulte o guia [Atividades](#).

- **Para iniciar um serviço:**

O [Service](#) é um componente que realiza operações em segundo plano sem interface do usuário. É possível iniciar um serviço para realizar uma operação que acontece uma vez (como baixar um arquivo) passando uma [Intent](#) a [startService\(\)](#). A [Intent](#) descreve o serviço a iniciar e carrega todos os dados necessários.

Se o serviço for projetado com uma interface servidor-cliente, é possível vincular ao serviço em outro componente passando uma [Intent](#) a [bindService\(\)](#). Para obter mais informações, consulte o guia [Serviços](#).

- **Para fornecer uma transmissão:**

Transmissão é uma mensagem que qualquer aplicativo pode receber. O sistema fornece diversas transmissões para eventos do sistema, como quando o sistema inicializa ou o dispositivo inicia o carregamento. Você pode fornecer uma transmissão a outros aplicativos passando um [Intent](#) a [sendBroadcast\(\)](#), [sendOrderedBroadcast\(\)](#) ou [sendStickyBroadcast\(\)](#).

Assim um objeto [Intent](#) carrega informações que o sistema Android usa para determinar o componente a iniciar (como o nome exato do componente ou categoria do componente que deve receber o intent), além de informações que o componente receptor usa para realizar a ação adequadamente (como a ação a tomar e os dados a usar).

No nosso exemplo vamos definir um Intent com uma ação do tipo ACTION_SEND também conhecida como o intent de "compartilhamento", que deve ser usada em um intent com [startActivity\(\)](#) quando houver alguns dados que o usuário possa compartilhar por meio de outro aplicativo, como um aplicativo de e-mail ou de compartilhamento social.

Recursos usados:

- [Visual Studio Community 2017](#) ou Xamarin Studio
- [Xamarin](#)
- Emulador Android virtual ou físico ([veja como emular usando o Vysor](#))

Criando o projeto no Visual Studio 2017 Community

Abra o [VS 2017 Community](#) e clique em **New Project**;

Selecione a linguagem **Visual C#** e o template **Android -> Blank App(Android)**

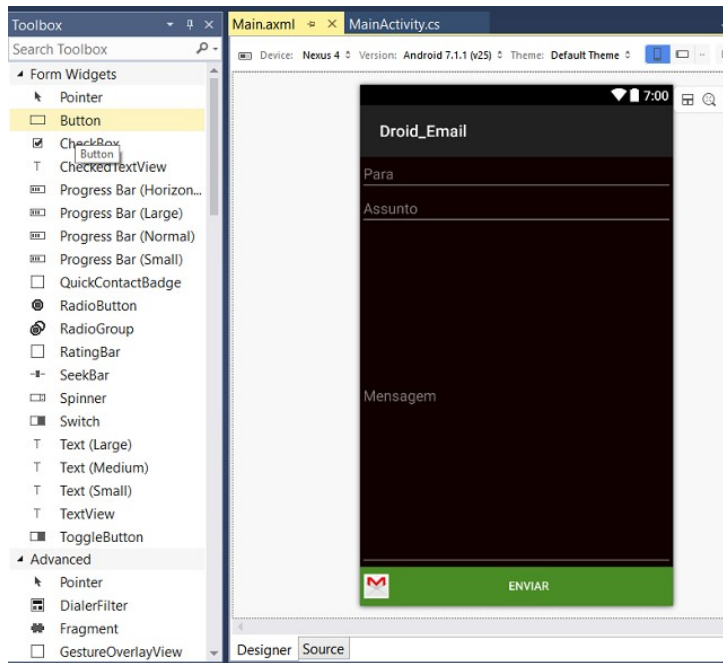
Informe o nome **Droid_Email** e clique no botão **OK**;

Definindo o leiaute da aplicação

Vamos Abrir o arquivo **Main.axml** na pasta **Resources/layout** e no modo **Designer** incluir um novo controle os seguintes controles a partir da **ToolBox** :

- **3 EditText** : [@+id/edtxtPara](#) , [@+id/edtxtAssunto](#) e [@+id/edtxtMensagem](#)
- **1 Button** : [@+id/btnEnviar](#)

Abaixo vemos o leiaute no emulador do Xamarin exibindo a tela e ao lado o respectivo código XML gerado :



```
<?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:background="#100000">
    <EditText
        android:hint="Para"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/edtxtPara" />
    <EditText
        android:hint="Assunto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/edtxtAssunto" />
    <EditText
        android:hint="Mensagem"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:id="@+id/edtxtMensagem" />
    <Button
        android:drawableLeft="@drawable/droidemail"
        android:text="Enviar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="5dp"
        android:background="#498b24"
        android:textColor="#FFF"
        android:id="@+id/btnEnviar" />
</LinearLayout>
```

Agora podemos iniciar a implementação do código para exibir enviar email no arquivo **MainActivity.cs**.

Vamos começar definindo os namespaces usados no projeto:

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

A seguir a declaração da Activity como sendo a principal e a definição do ícone da aplicação:

```
[Activity(Label = "Droid_Email", MainLauncher = true)]
```

No método **OnCreate()** vamos incluir o código abaixo:

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

    // Set our view from the "main" layout resource
    SetContentView(Resource.Layout.Main);
    var edtxtPara = FindViewById<EditText>(Resource.Id.edtxtPara);
    var edtxtAssunto = FindViewById<EditText>(Resource.Id.edtxtAssunto);
    var edtxtMensagem = FindViewById<EditText>(Resource.Id.edtxtMensagem);
    var btnEnviar = FindViewById<Button>(Resource.Id.btnEnviar);

    btnEnviar.Click += (s, e) =>
    {
        Intent email = new Intent(Intent.ActionSend);

        email.PutExtra(Intent.ExtraEmail, new string[] { edtxtPara.Text.ToString() });
        email.PutExtra(Intent.ExtraSubject, edtxtAssunto.Text.ToString());
        email.PutExtra(Intent.ExtraText, edtxtMensagem.Text.ToString());

        email.SetType("message/rfc822");

        StartActivity(Intent.CreateChooser(email, "Enviar email Via"));
    };
}
```

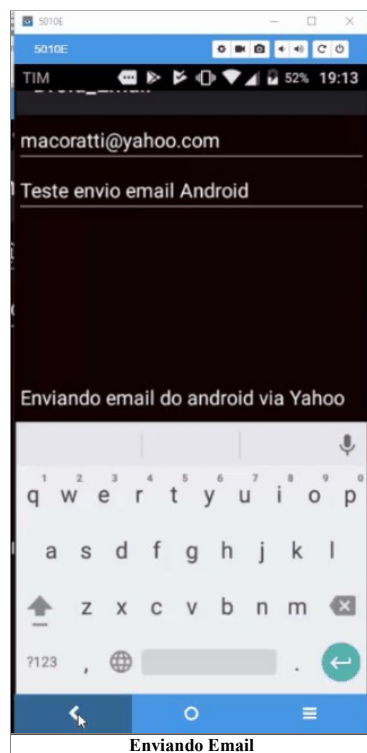
No evento **Click** do botão para enviar o email criamos uma **Intent** com uma **Action ActionSend** que permite enviar dados a alguém. A quem os

dados estão sendo entregues não está especificado; cabe ao destinatário desta ação solicitar ao usuário onde os dados devem ser enviados.

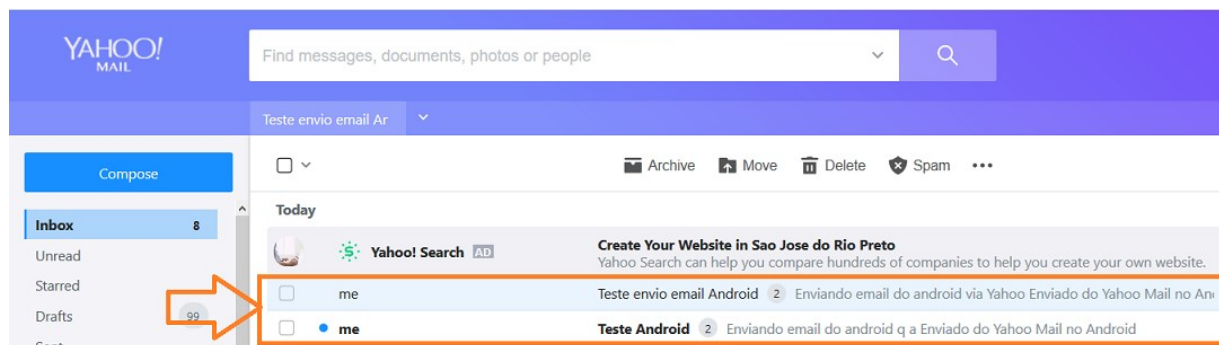
A seguir definimos o destino, o assunto e a mensagem e ao definir o tipo **mime** da intent para **message/rfc822** fazemos com que o aplicativo de email seja iniciado. Se existirem mais de um aplicativo capaz de enviar email, o usuário receberá uma lista e poderá selecionar um deles.

Ao final iniciamos uma **Activity** usando **intent.CreateChooser** que permite que o usuário escolha outro aplicativo de email e não somente o padrão. Esse recurso é muito útil para escolher qual aplicativo de email usar para enviar o email.

Executando o projeto iremos obter o seguinte resultado obtido usando o emulador **Vysor** e o meu dispositivo físico **Alcatel**:



Verificando a minha caixa de entrada no **Yahoo** constatamos que o email foi enviado com sucesso:



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

"Jesus lhes respondeu, e disse: A minha doutrina não é minha, mas daquele que me enviou.

Se alguém quiser fazer a vontade dele, pela mesma doutrina conhecerá se ela é de Deus, ou se eu falo de mim mesmo."

[João 7:16,17](#)

[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 - Video Aulas](#)

Quer aprender C# ??

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

- [Curso C# Basico - Video 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](#)
- [Macoratti .net | Facebook](#)
- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macoratti\) | Twitter](#)
- [Xamarin - Desenvolvimento Multiplataforma com C# ... - Macoratti.net](#)
- [Xamarin Android - Primeira Aplicação Android : Activity ... - Macoratti](#)
- [Xamarin Android - Intents : Trabalhando com mais de uma ... - Macoratti](#)
- [Xamarin Android - Fazendo chamadas telefônicas - Macoratti](#)
- [Xamarin Android - Enviando SMS - Macoratti](#)
- [Xamarin Android - Enviando notificações - I - Macoratti](#)

[José Carlos Macoratti](#)