



Neste artigo vamos criar um cronômetro usando o Xamarin Android.



Curso de Xamarin Forms Vídeo Aulas

Desenvolva para Android, iOS e Windows Phone

Neste artigo vamos criar um cronômetro usando os recursos básicos do **Xamarin Android** e a linguagem C#.

Recursos usados:

- [Visual Studio Community 2017](#) ou **Xamarin Studio**
- [Xamarin](#)

Criando o projeto no VS Community 2017

Abra o **VS 2017 Community** e clique em **New Project**;

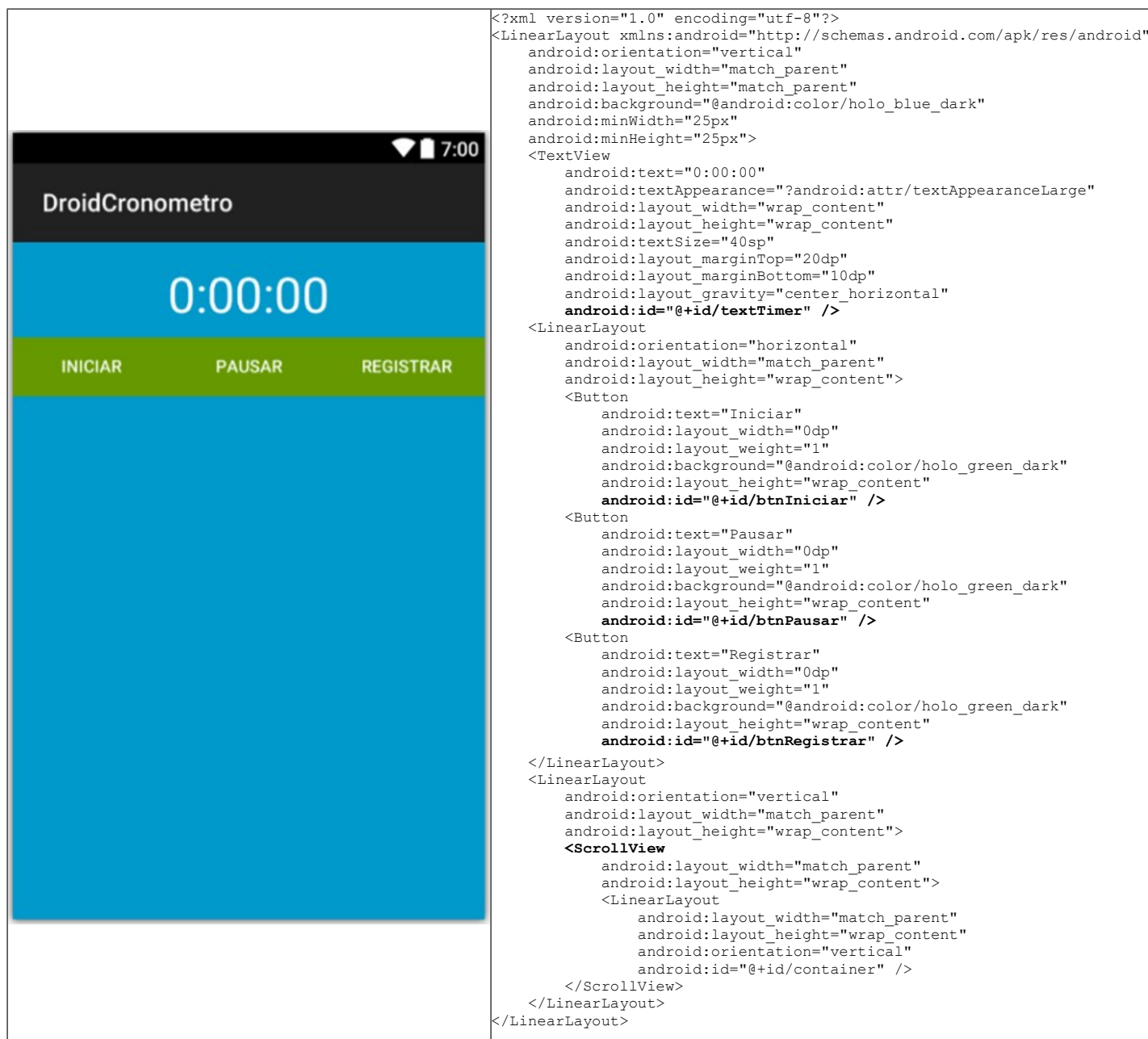
Selecione a linguagem Visual C# e o template **Android -> Blank App(Android)**

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

Abra o arquivo **Main.axml** na pasta **Resources/layout** e no modo **Designer** e a seguir inclua a partir da ToolBox os seguintes controles:

- **1 TextView - txtTimer**
- **3 Buttons - btnIniciar, btnParar, btnRegistrar**
- **1 ScrollView -**

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



Observe que utilizamos o **LinearLayout** para dispor os botões alinhados horizontalmente.

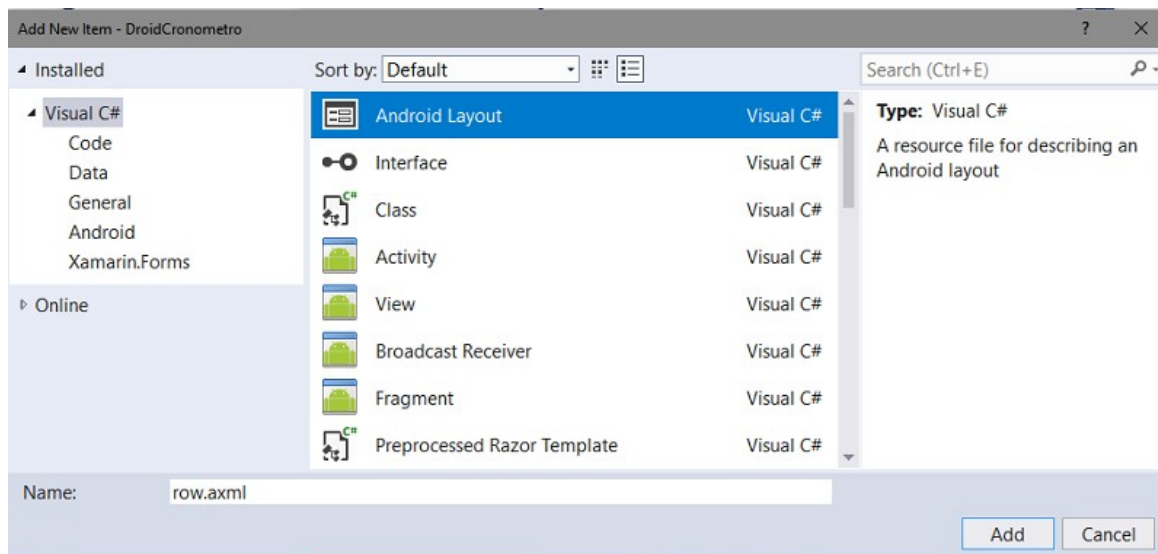
O controle **ScrollView** será usado como contêiner para exibir outra view usando o recurso de **Fragments**.

Podemos pensar nos **Fragments** como módulos de interface. Eles permitem que o desenvolvedor divida a interface do usuário em partes isoladas, reutilizáveis que podem ser executadas em atividades separadas. Em tempo de execução, as atividades em si vão decidir quais **Fragments** usar.

Usando o recurso **Fragments** podemos dividir nossa tela e dar comportamento a cada pedaço (*fragmento*) de uma view com um **Fragment**.

Criando o Layout row

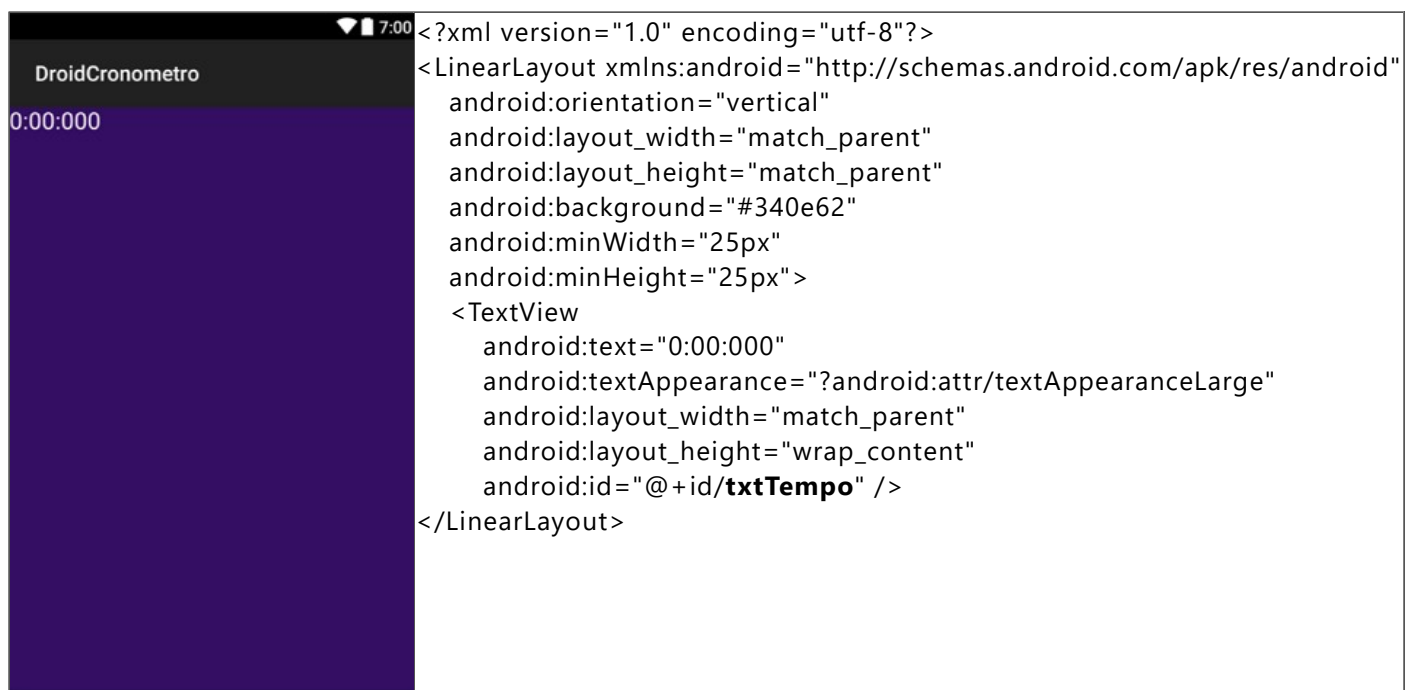
Vamos criar outro Layout na pasta layout selecionando a pasta e no menu **Project** escolhendo o template **Android Layout** e informando o nome **row.xml**:



A seguir selecione o arquivo **row.xml** e inclua a partir da ToolBox o controle

- **TextView - txtTempo**

Abaixo vemos o leiaute no emulador do Xamarin e ao lado o respectivo código XAML gerado :



Implementando o código na MainActivity

Agora vamos definir o código no arquivo **MainActivity.cs** vinculado a nossa view **Main.xml** e usar a view row.xml usando fragments para exibir os tempos do cronômetro.

```
using Android.App;
using Android.Widget;
using Android.OS;
using System.Timers;
using Android.Views;
using Android.Content;

namespace DroidCronometro
```

```
{
    [Activity(Label = "DroidCronometro", MainLauncher = true)]
    public class MainActivity : Activity
    {
        Button btnIniciar, btnPausar, btnRegistrar;
        TextView txtTimer;
        LinearLayout container;
        Timer timer;
        int mins = 0, segs = 0, milesegs = 1;

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

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

            btnIniciar = FindViewById<Button>(Resource.Id.btnIniciar);
            btnPausar = FindViewById<Button>(Resource.Id.btnPausar);
            btnRegistrar = FindViewById<Button>(Resource.Id.btnRegistrar);
            container = FindViewById<LinearLayout>(Resource.Id.container);
            txtTimer = FindViewById<TextView>(Resource.Id.textTimer);

            btnIniciar.Click += BtnIniciar_Click;
            btnPausar.Click += BtnPausar_Click;
            btnRegistrar.Click += btnRegistrar_Click;
        }

        private void btnRegistrar_Click(object sender, System.EventArgs e)
        {
            LayoutInflater inflater = (LayoutInflater)BaseContext.GetService(Context.LayoutInflaterService);
            View addView = inflater.Inflate(Resource.Layout.row, null);
            TextView txtConteudo = addView.FindViewById<TextView>(Resource.Id.txtTempo);
            txtConteudo.Text = txtTimer.Text;
            container.AddView(addView);
        }

        private void BtnPausar_Click(object sender, System.EventArgs e)
        {
            timer.Stop();
            timer = null;
        }

        private void BtnIniciar_Click(object sender, System.EventArgs e)
        {
            timer = new Timer();
            timer.Interval = 1;
            timer.Elapsed += Timer_Elapsed;
            timer.Start();
        }

        private void Timer_Elapsed(object sender, ElapsedEventArgs e)
        {
            milesegs++;
            if(milesegs >= 1000)
            {
                segs++;
                milesegs = 0;
            }
        }
    }
}
```

```
    }  
    if(segs==59)  
    {  
        mins++;  
        segs = 0;  
    }  
    RunOnUiThread(() =>  
    {  
        txtTimer.Text = string.Format("{0}:{1:00}:{2:000}", mins, segs, milesegs);  
    });  
}  
}
```

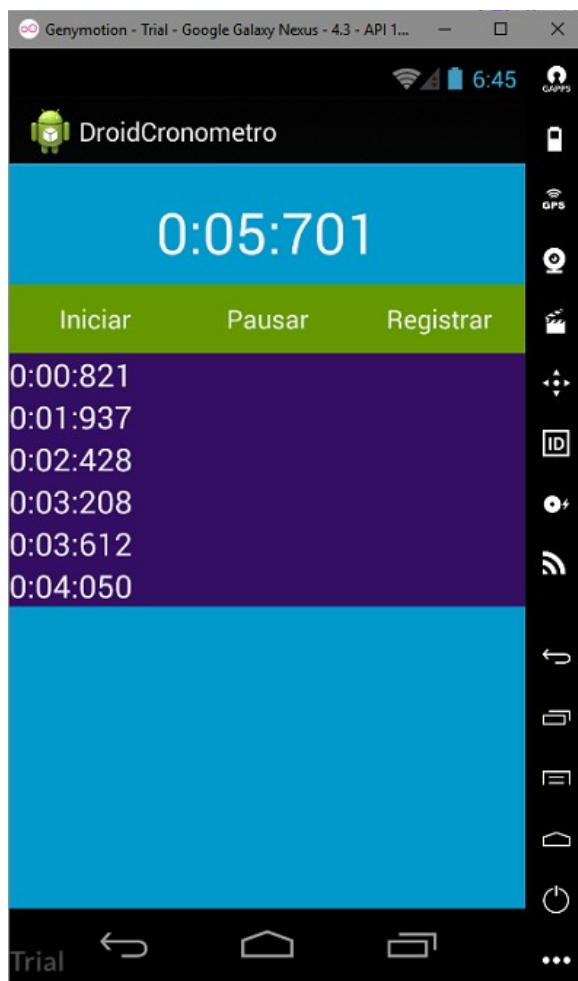
No botão **Registrar** estamos usando a classe **LayoutInflater** que instancia um arquivo XML de layout em seus objetos View correspondentes.

O método [inflate\(\)](#) usa três argumentos:

- O **ID** do recurso do layout que você quer inflar;
- O [ViewGroup](#) que será pai do layout inflado. Passar o container é importante para que o sistema aplique os parâmetros de layout à view raiz do layout inflado, especificado pela view pai em que está ocorrendo;
- Um **booleano** que indica se o layout inflado deve ser anexado à [ViewGroup](#) (o segundo parâmetro) durante a inflação (*neste caso, isto é falso, pois o sistema já está inserindo o layout inflado no container, retornar como verdadeiro criaria um grupo de vistas redundante no layout final*).

Este é o modo de criar um fragmento que fornece um layout.

Executando o projeto e fazendo algumas medições de tempo iremos obter o seguinte resultado:



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

E a vida eterna é esta: que te conheçam, a ti só, por único Deus verdadeiro, e a Jesus Cristo, a quem enviaste.

[João 17:3](#)

[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# 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](#)

Quer aprender a criar aplicações Web Dinâmicas
usando a ASP .NET MVC 5 ?

- [Curso ASP .NET MVC 5 - Vídeo Aulas](#)

Gostou ?  [Compartilhe no Facebook](#)  [Compartilhe no Twitter](#)

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](#)
- [Seção Vídeo Aulas do site Macoratti .net](#)
- [Xamarin Android - Apresentando e usando Fragments - Macoratti.net](#)
- [Xamarin Android - Criando Tabs - Macoratti.net](#)
- [Xamarin Android - CRUD Básico com SQLite - Macoratti.net](#)
- [Seção Xamarin Android do portal Macoratti .net](#)

[José Carlos Macoratti](#)