

## Macoratti.net Xamarin Android - Usando um Calendario



Curso de Xamarin Forms Vídeo Aulas  
Desenvolva para Android, iOS e Windows Phone



Neste artigo vou mostrar com usar um calendário no [Xamarin Android](#) usando o Visual Studio 2015 e a linguagem C# e a **API Alliance Calendar**.

Curso C# Vídeo Aulas  
Do básico ao intermediário

Por um preço justo

Muitas vezes precisamos exibir um calendário em nossas aplicações Android e existem diversas formas de você fazer isso. Neste artigo eu mostro como usar a **API Alliance Calendar** para exibir um calendário.

Essa API pode ser obtida através do item **Components** do projeto Android clicando com o botão direito e a seguir selecionando : **Get More Components...**

Ela é bem simples e fácil de usar e é totalmente grátis.

Vamos ver como usar essa API.

### Recursos usados:

- [Visual Studio Community 2015](#) ou Xamarin Studio
- [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 2015

Abra o [VS 2015 Community](#) e clique em **New Project**;

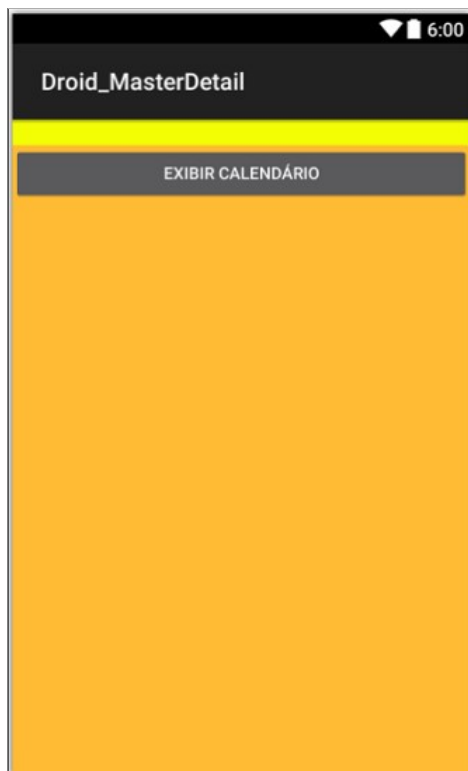
Selecione a linguagem Visual C# e o template **Android -> Blank App(Android)**

Informe o nome um nome adequado ao seu projeto, eu vou usar o nome Droid\_MasterDetail, e clique no botão **OK**;

Abra o arquivo **Main.axml** na pasta **Resources/layout** no modo **Designer** e a seguir inclua uma caixa de texto usando o controle **EditText** e um Button apenas para simular um formulário.

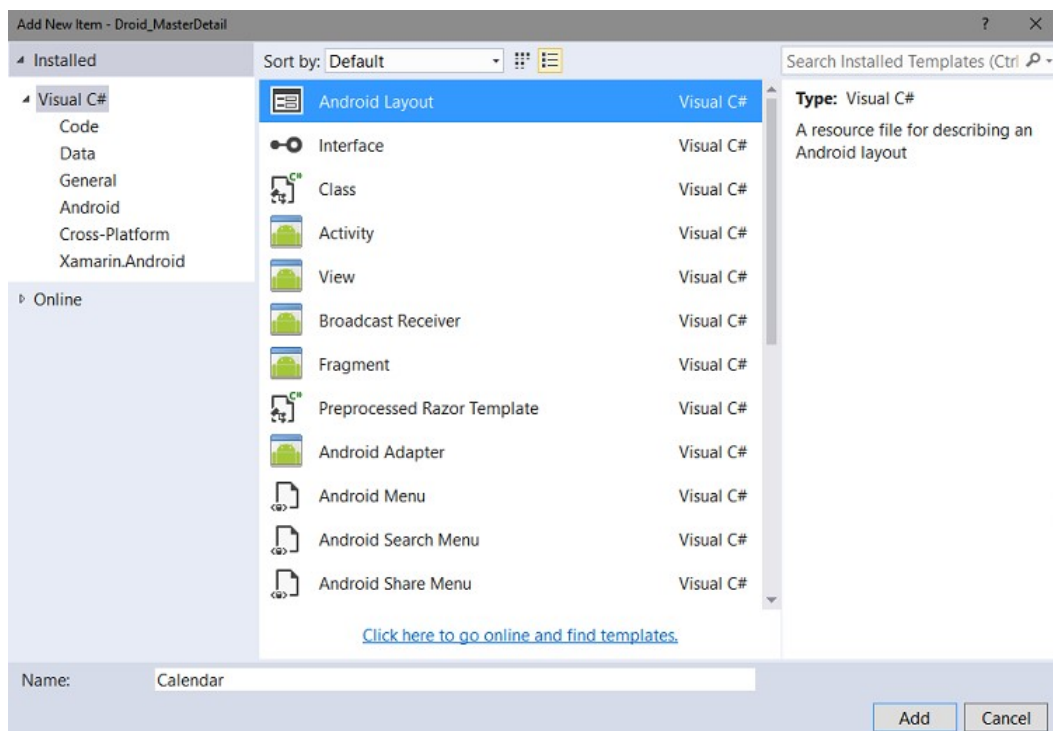
- 1 EditText
- 1 Button - id=@+id/**btnNavegar**

Abaixo vemos o leiaute no emulador do Xamarin 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:minWidth="25px"
    android:minHeight="25px"
    android:background="@android:color/holo_orange_light">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#f3fd04"
        android:textColor="#1d1160"
        android:id="@+id/edittxtUrl" />
    <Button
        android:text="Exibir Calendário"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btnNavegar" />
</LinearLayout>
```

A seguir inclua um novo Layout na pasta **Resources/layout** clicando em **Project-> Add New Item** e selecionando a opção **Android Layout** e informando o nome **Calendar**.

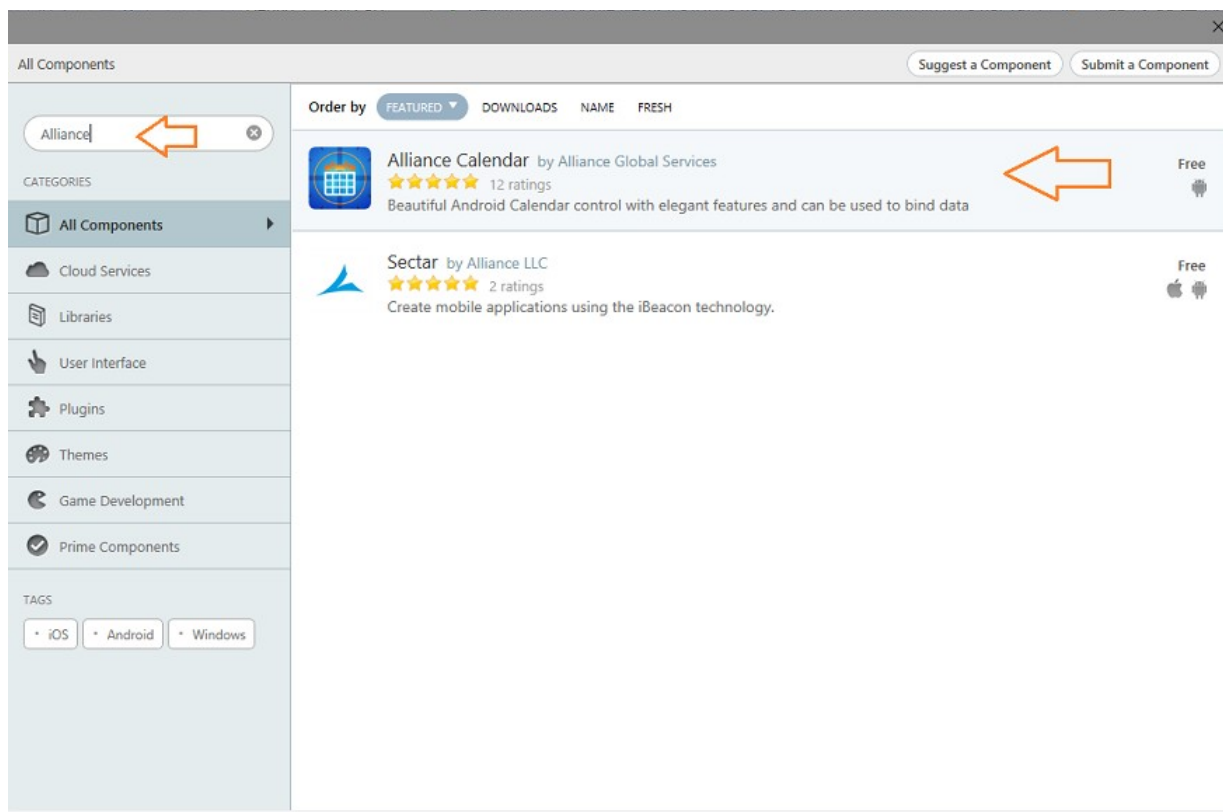


## Referenciando a API no projeto

Precisamos referenciar a API Alliance Calendar no projeto.

Clique no item **Components** com o botão direito do mouse, e a seguir clique em **Get More Components...**

Será aberto a janela de componentes. Digite **Alliance** na caixa de busca e selecione o componente **Alliance Calendar**:



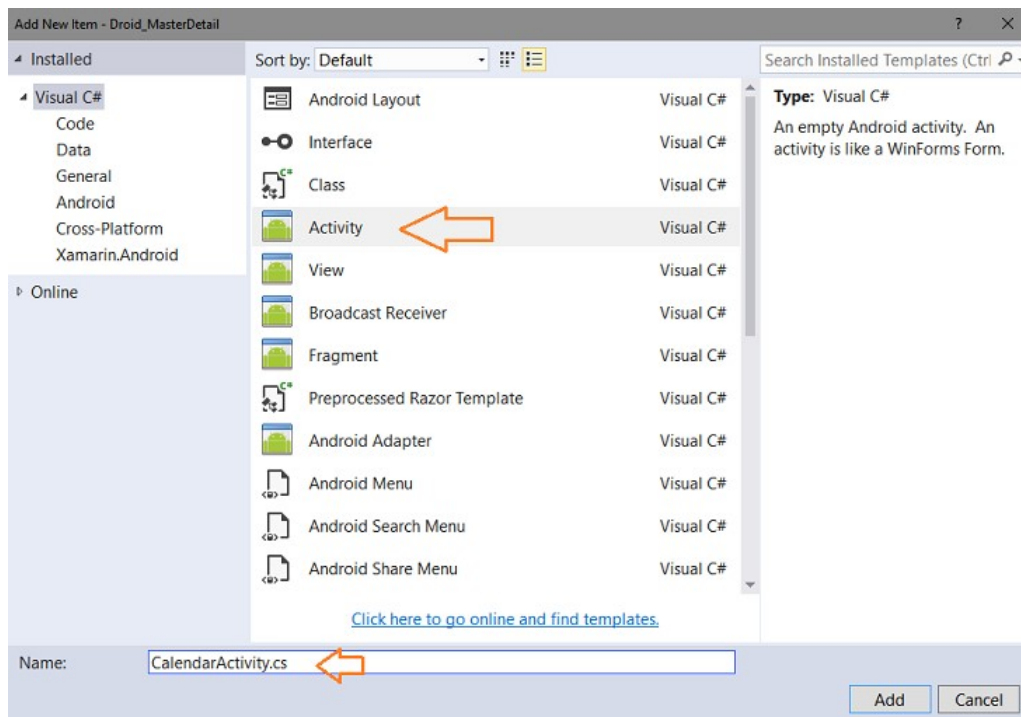
Ele será instalado no seu projeto.

Agora abra o arquivo **Calendar.axml** e inclua o código abaixo neste arquivo:

```
<?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">
    <com.alliance.calendar.CustomCalendar
        android:id="@+id/CalendarControl"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</LinearLayout>
```

Vamos agora criar a **Activity** referente a este arquivo de layout.

Selecione o projeto e no menu **Project** clique em **Add New Item**, e, selecione **Activity** e informe o nome **CalendarActivity**:



## Definindo o código das Activities

Abra o arquivo MainActivity e inclua o código abaixo :

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

namespace Droid_MasterDetail
{
    [Activity(Label = "Droid_MasterDetail", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);

            SetContentView (Resource.Layout.Main);
            var button = FindViewById<Button>(Resource.Id.btnNavegar);
            button.Click += Button_Click;
        }

        private void Button_Click(object sender, System.EventArgs e)
        {
            var intent = new Intent(this, typeof(CalendarActivity));
            StartActivity(intent);
        }
    }
}
```

Este código apenas referencia o arquivo de Layout **Main.axml** e define o evento **Click** para o botão **btnNavegar**.

A seguir no tratamento do evento criamos uma Intent para iniciar a activity **CalendarActivity**.

Abra o arquivo **CalendarActivity** e defina o código a seguir:

```
using System;
using System.Collections.Generic;
using Android.App;
using Android.OS;
using Android.Widget;
using com.alliance.calendar;

namespace Droid_MasterDetail
{
    [Activity(Label = "CalendarActivity", ParentActivity= typeof(MainActivity))]
    public class CalendarActivity : Activity
    {
        CustomCalendar CalendarControl;
        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            SetContentView(Resource.Layout.Calendar);

            CalendarControl = FindViewById<CustomCalendar>(Resource.Id.CalendarControl);
            CalendarControl.NextButtonText = ">>";
            CalendarControl.PreviousButtonText = "<<";

            //CalendarControl.NextButtonVisibility= ViewStates.Invisible;
            //CalendarControl.PreviousButtonStyleId = Resource.Drawable.default_dim_selector;
            //CalendarControl.ShowOnlyCurrentMonth = true;

            CalendarControl.ShowFromDate = new DateTime();
            List<CustomCalendarData> customData = new List<CustomCalendarData>();

            //destacando datas no calendário
            customData = new List<CustomCalendarData>
            {
                new CustomCalendarData(DateTime.Now.AddDays(2)),
                new CustomCalendarData(DateTime.Now.AddDays(4)),
                new CustomCalendarData(DateTime.Now.AddDays(-4))
            };

            CalendarControl.CustomDataAdapter = customData;
            CalendarControl.OnCalendarMonthChange += CalendarControl_CalendarMonthChange;
            CalendarControl.OnCalendarSelectedDate += CalendarControl_CalendarDateSelected;
        }

        private void CalendarControl_CalendarDateSelected(object sender, CalendarDateSelectionEventArgs e)
        {
            Toast.MakeText(this, e.SelectedDate.ToString(), ToastLength.Short).Show();
        }

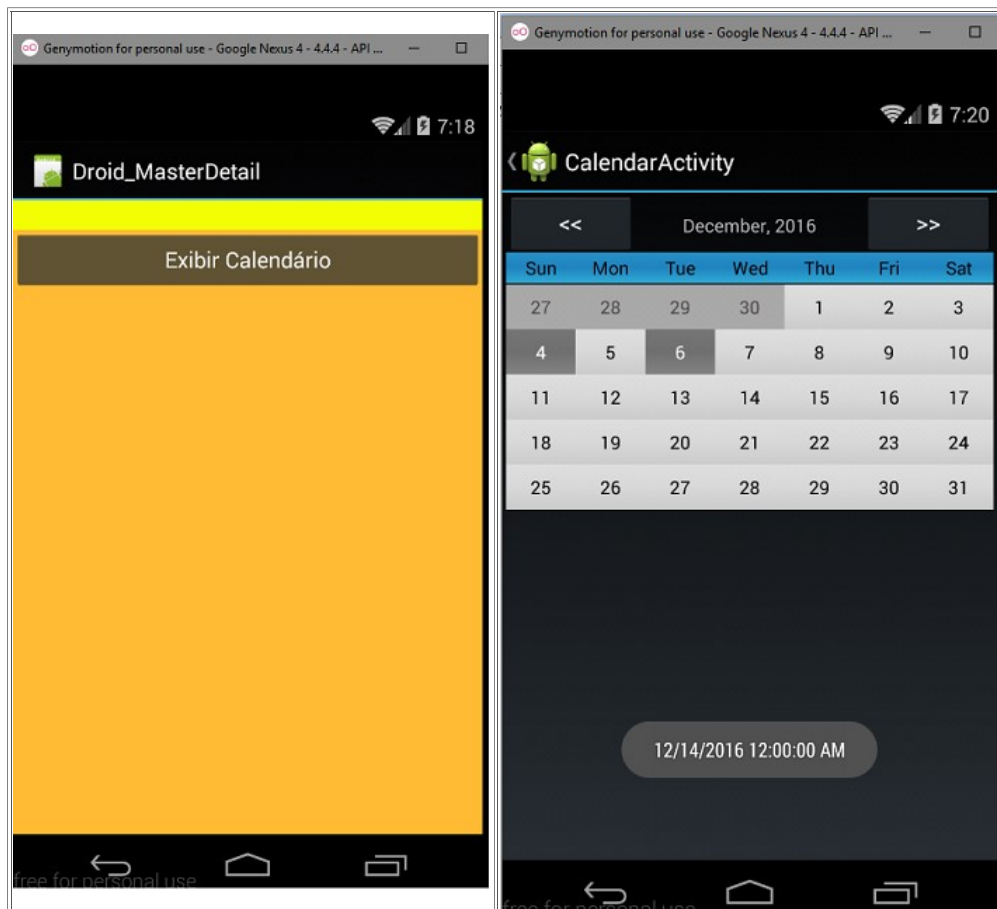
        private void CalendarControl_CalendarMonthChange(object sender, CalendarNavigationEventArgs e)
        {
            if (e.MonthChange == CalendarHelper.MonthChangeOn.Next)
            {
            }
            else if (e.MonthChange == CalendarHelper.MonthChangeOn.Previous)
            {
            }
        }
    }
}
```

O código acima foi obtido da página da API do calendário em : <https://components.xamarin.com/gettingstarted/alliance-calendar-component/true>

Eu apenas alterei o atributo **[Activity...]** incluindo o código em azul de forma a tornar a página **MainActivity** a página pai. Assim teremos um botão para retornar para esta página na tela do calendário.

```
[Activity(Label = "CalendarActivity", ParentActivity= typeof(MainActivity))]
```

Executando o projeto usando o emulador **Genymotion** iremos obter o seguinte resultado:



Na primeira tela ao clicar no botão - **Exibir Calendário** - veremos o calendário ser exibido na tela e no topo da mesma vemos o botão < (**Back**) para retornar.

Foram implementados alguns eventos como a seleção da data que exibe um alerta com a data selecionada.

Você pode implementar facilmente outros recursos para personalizar o calendário para o seu projeto.

Pegue o projeto aqui : [📁 Droid\\_MasterDetail.zip](#) (sem as referências)

**Ninguém pode servir a dois senhores; porque ou há de odiar um e amar o outro, ou se dedicará a um e desprezará o outro. Não podeis servir a Deus e a Mamom.**

**Mateus 6:24**

[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](#) 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](#)
- [VB.NET 2005 - Controles - Macoratti.net](#)
- [Seção de Jogos do site Macoratti .net](#)
- [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/api/type/Android.App.AlertDialog/>
- [Xamarin Android - Tratando eventos de forma declarativa](#)
- [Seção Mobile/Xamarin do site Macoratti .net](#)
- <https://developer.android.com/reference/android/app/Activity.html>
- <https://developer.xamarin.com/api/type/Android.Widget.ProgressBar/>