

Macoratti.net Xamarin

Android - CRUD Básico com SQLite - II



Neste tutorial vou mostrar como implementar as operações de manutenção de dados - **CRUD** - usando um banco de dados **SQLite** no **Xamarin Android** com **VS 2015 Community** e a linguagem **C#**.



Curso ASP .NET MVC 5 - Vídeo Aulas
Do básico ao intermediário - crie site dinâmicos

Na [primeira parte do artigo](#) preparamos tudo que precisamos para poder implementar o CRUD:

- Criamos o projeto No VS 2015;
- Definimos a classe **Aluno**;
- Definimos a camada de acesso a dados **DataBase**;
- Definimos o leiaute da interface no arquivo **Main.axml**;
- Definimos o leiaute que iremos usar para exibir os itens no listview no arquivo **ListViewLayout.axml**;
- Definimos a classe **ListAdapter.cs** que implementa a classe base **BaseAdapter** e cria um **Adapter** para exibir os itens;

Vamos então implementar as operações CRUD.

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**.

Abrindo o projeto no Visual Studio 2015 Community

Abra o projeto **App.Crud_Xamarin** criado na primeira parte do artigo.

A seguir abra o arquivo **MainActivity.cs** e inclua o código abaixo neste arquivo:

```
using Android.App;
using Android.OS;
using Android.Widget;
using App.Crud_Xamarin.Resources;
using App.Crud_Xamarin.Resources.DataBaseHelper;
using App.Crud_Xamarin.Resources.Model;
using System.Collections.Generic;

namespace App.Crud_Xamarin
{
    [Activity(Label = "App.Crud_Xamarin", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        ListView lvDados;
        List<Aluno> listaAlunos = new List<Aluno>();
        DataBase db;

        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);
            SetContentView(Resource.Layout.Main);

            //criar banco de dados
        }
    }
}
```

CriarBancoDados();

```
lvDados = FindViewById<ListView>(Resource.Id.lvDados);

var txtNome = FindViewById<EditText>(Resource.Id.txtNome);
var txtIdade = FindViewById<EditText>(Resource.Id.txtIdade);
var txtEmail = FindViewById<EditText>(Resource.Id.txtEmail);

var btnIncluir = FindViewById<Button>(Resource.Id.btnIncluir);
var btnEditar = FindViewById<Button>(Resource.Id.btnEditar);
var btnDeletar = FindViewById<Button>(Resource.Id.btnDeletar);

//carregar Dados
CarregarDados();

//botão Incluir
btnIncluir.Click += delegate
{
    Aluno aluno = new Aluno()
    {
        Nome = txtNome.Text,
        Idade = int.Parse(txtIdade.Text),
        Email = txtEmail.Text
    };
    db.InserirAluno(aluno);
    CarregarDados();
};

//botão editar
btnEditar.Click += delegate
{
    Aluno aluno = new Aluno()
    {
        Id = int.Parse(txtNome.Tag.ToString()),
        Nome = txtNome.Text,
        Idade = int.Parse(txtIdade.Text),
        Email = txtEmail.Text
    };
    db.AtualizarAluno(aluno);
    CarregarDados();
};

//botão deletar
btnDeletar.Click += delegate
{
    Aluno aluno = new Aluno()
    {
        Id = int.Parse(txtNome.Tag.ToString()),
        Nome = txtNome.Text,
        Idade = int.Parse(txtIdade.Text),
        Email = txtEmail.Text
    };
    db.DeletarAluno(aluno);
    CarregarDados();
};

//evento itemClick do ListView
lvDados.ItemClick += (s, e) =>
{
    for (int i = 0; i < lvDados.Count; i++)
    {
        if (e.Position == i)
            lvDados.GetChildAt(i).SetBackgroundColor(Android.Graphics.Color.Chocolate);
        else
            lvDados.GetChildAt(i).SetBackgroundColor(Android.Graphics.Color.Transparent);
    }

    //vinculando dados do listview
}
```

```

var lvtxtNome = e.View.FindViewById<TextView>(Resource.Id.txtvNome);
var lvtxtIdade = e.View.FindViewById<TextView>(Resource.Id.txtvIdade);
var lvtxtEmail = e.View.FindViewById<TextView>(Resource.Id.txtvEmail);

txtNome.Text = lvtxtNome.Text;
txtNome.Tag = e.Id;
txtIdade.Text = lvtxtIdade.Text;
txtEmail.Text = lvtxtEmail.Text;
};
}

//rotina para criar o banco de dados
private void CriarBancoDados()
{
    db = new DataBase();
    db.CriarBancoDeDados();
}

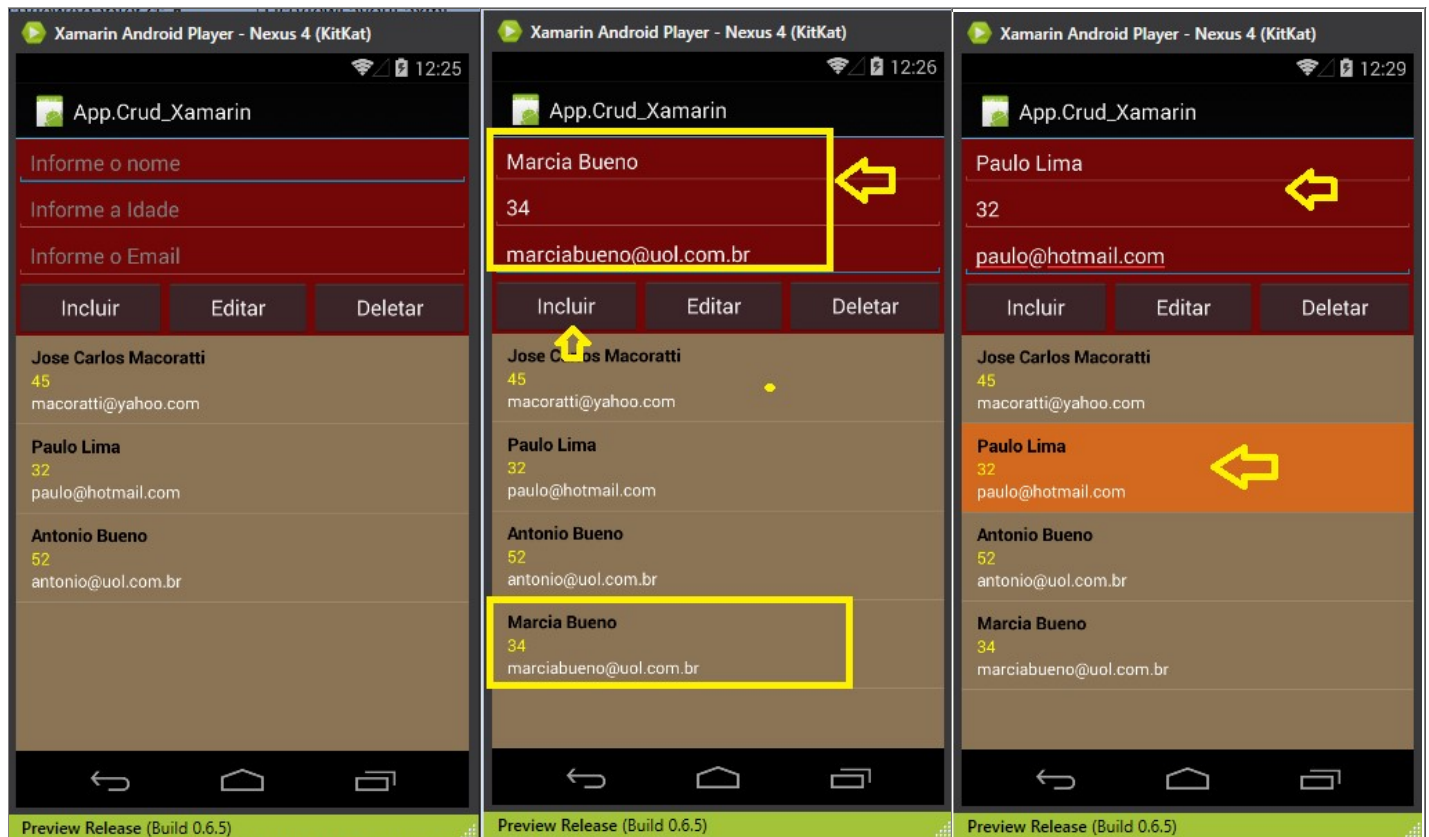
//Obtem todos os alunos da tabela Aluno e exibe no ListView
private void CarregarDados()
{
    listaAlunos = db.GetAlunos();
    var adapter = new ListViewAdapter(this, listaAlunos);
    lvDados.Adapter = adapter;
}
}
}

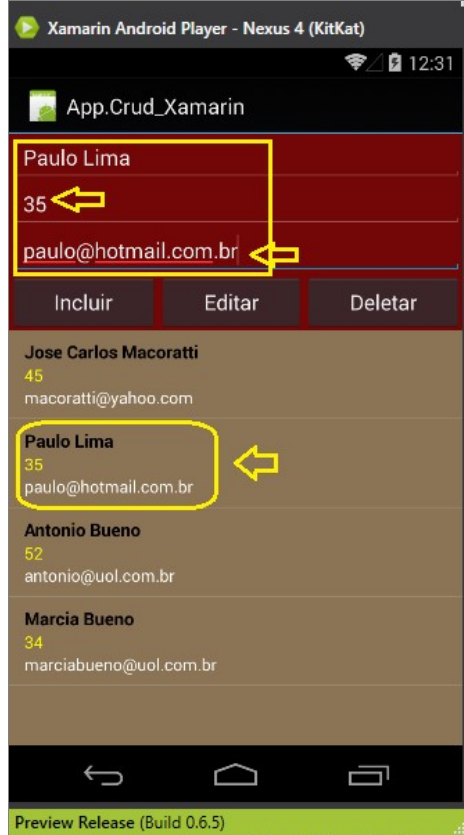
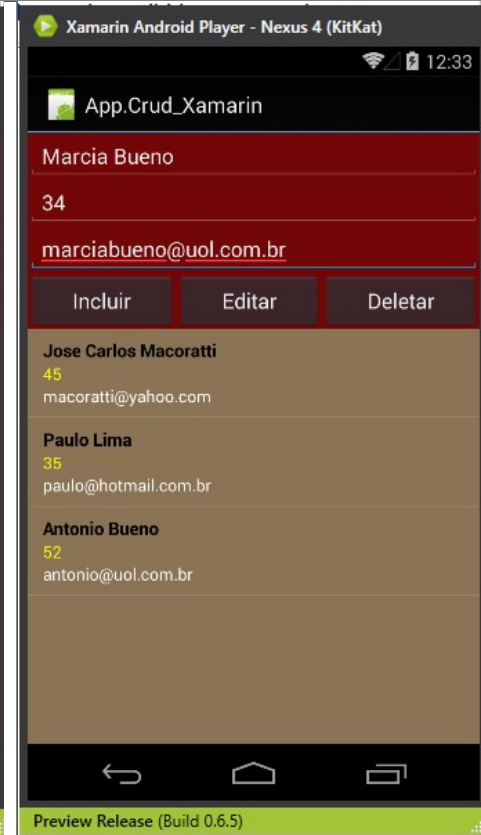
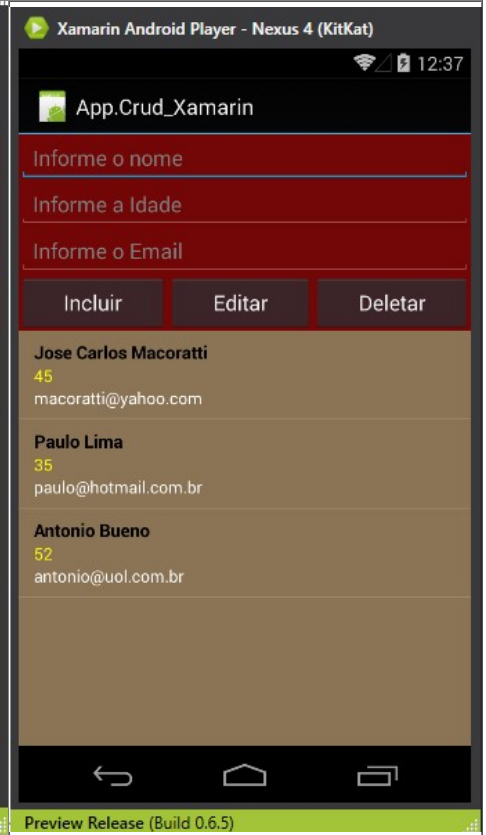
```

Neste código definimos em cada evento **Click** dos botões as operações **CRUD** para incluir, alterar e excluir informações da tabela **Aluno**.

No evento **ItemClick** do **ListView** estamos obtendo as informações do aluno selecionado e exibindo nos controles **EditText**. Note que durante esta operação atribuímos o valor do Id do aluno para a propriedade **Tag** do controle EditText.

Executando o projeto iremos obter:



<p>A tela de apresentação da aplicação exibindo os dados cadastrados no ListView, os Buttons e as EditText para receber a entrada do usuário</p>	<p>Inclusão de um novo aluno no banco de dados. O usuário informou os dados e clicou no botão Incluir.</p>	<p>Selecionando um item na ListView o mesmo é destacado e exibido nos controles.</p>
		
<p>Alteramos a idade e o email do aluno selecionado e clicamos no botão Editar para alterar os dados e exibi-los no Listview</p>	<p>Selecionando um aluno e clicando no botão Deletar o mesmo é excluído e removido do ListView</p>	<p>Encerrando e abrindo a aplicação novamente constatamos que o registro foi realmente removido da tabela.</p>

Pegue o projeto completo aqui : [📦 App.Crud Xamarin.zip](#) (sem as referências)

Meus filhinhos, não amemos de palavra, nem de língua, mas por obra e em verdade.

E nisto conhecemos que somos da verdade, e diante dele asseguraremos nossos corações;

Sabendo que, se o nosso coração nos condena, maior é Deus do que o nosso coração, e conhece todas as coisas.

1 João 3:18-20

[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 \(@macorati\) | 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/api/type/Android.App.AlertDialog/>
- [Xamarin Android - Tratando eventos de forma declarativa](#)

[José Carlos Macoratti](#)