

## Macoratti.net Xamarin Android - Login com SQLite - II



Neste tutorial vou mostrar como implementar o login usando um banco de dados **SQLite** no Xamarin Android com **VS 2015 Community** e a linguagem **C#**.

**Curso C# Vídeo Aulas**  
Do básico ao intermediário

Por um preço justo

Na [primeira parte do artigo](#) criamos a solução no Visual Studio, incluímos a referência ao **SQLite** em nosso projeto, definimos o modelo de domínio e definimos os seguintes [layouts](#) e [activities](#):

### Layouts:

- Main.xml
- NovoUsuario.xml
- Login.xml

**Activities** que irão gerenciar a respectiva view:

- MainActivity (Main)
- RegistrarActivity (NovoUsuario)
- LoginActivity (Login)

Podemos usar os seguintes métodos do objeto **SQLiteConnection** para realizar operações com dados:

- **Insert** – Adiciona um novo objeto no banco de dados;
- **Get<T>** – Tenta retornar um objeto usando a chave primária;
- **Table<T>** – Retorna todos os objetos d a tabela;
- **Delete** – Deleta um objeto usando sua chave primária;
- **Query<T>** – Realiza uma consulta SQL que retorna um número de linhas (como objetos);
- **Execute** – Use este método, ao invés de Query, quando você não esperar que linhas sejam retornadas da instrução SQL. (*INSERT, UPDATE e DELETE*).

Vamos agora implementar o código em cada uma das Activities para realizar o login, o registro de um novo usuário e o login com sucesso.

### 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**.

### Implementando o código na Activity : MainActivity

Abra o projeto o **App.Login\_SQLite** criadno no **VS 2015 Community** na primeira parte do artigo.

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

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Widget;
using SQLite;
using System;
using System.IO;

namespace App.Login_SQLite
{
    [Activity(Label = "App.Login_SQLite", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        EditText txtUsuario;
        EditText txtSenha;
        Button btnCriar;
        Button btnLogin;

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

            SetContentView(Resource.Layout.Main);

            btnLogin = FindViewById<Button>(Resource.Id.btnLogin);
            btnCriar = FindViewById<Button>(Resource.Id.btnRegistrar);
            txtUsuario = FindViewById<EditText>(Resource.Id.txtUsuario);
            txtSenha = FindViewById<EditText>(Resource.Id.txtSenha);

            btnLogin.Click += BtnLogin_Click;
            btnCriar.Click += BtnCriar_Click;
        }
    }
}
```

```

        CriarBancoDeDados();
    }

    private void BtnCriar_Click(object sender, EventArgs e)
    {
        StartActivity(typeof(RegistrarActivity));
    }

    private void BtnLogin_Click(object sender, EventArgs e)
    {
        try
        {
            string dbPath = Path.Combine(System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal), "Usuario.db3");
            var db = new SQLiteConnection(dbPath);
            var dados = db.Table<Login>();

            var login = dados.Where(x => x.usuario == txtUsuario.Text && x.senha == txtSenha.Text).FirstOrDefault();

            if (login != null)
            {
                Toast.MakeText(this, "Login realizado com sucesso", ToastLength.Short).Show();
                var atividade2 = new Intent(this, typeof(LoginActivity));
                //pega os dados digitados em txtUsuario
                atividade2.PutExtra("nome", FindViewById<EditText>(Resource.Id.txtUsuario).Text);
                StartActivity(atividade2);
            }
            else
            {
                Toast.MakeText(this, "Nome do usuário e/ou Senha inválida(os)", ToastLength.Short).Show();
            }
        }
        catch (Exception ex)
        {
            Toast.MakeText(this, ex.ToString(), ToastLength.Short).Show();
        }
    }

    private void CriarBancoDeDados()
    {
        try
        {
            string dbPath = Path.Combine(System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal), "Usuario.db3");
            var db = new SQLiteConnection(dbPath);
        }
        catch (Exception ex)
        {
            Toast.MakeText(this, ex.ToString(), ToastLength.Short).Show();
        }
    }
}

```

Este código cria as instâncias dos componentes definidos no layout, cria o banco de dados e os eventos dos botões para logar e registrar um novo usuário.

No evento **Click** do botão de login usamos uma consulta **Linq** :

```
var login = dados.Where(x => x.usuario == txtUsuario.Text && x.senha == txtSenha.Text).FirstOrDefault();
```

para verificar se o usuário existe na tabela Login. Se o usuário existir emitimos uma mensagem e usamos uma **intent** para armazenar o nome do usuário para passar para outra **Activity**.

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.

Criamos uma instância da classe **Intent**() e a seguir usamos o método **PutExtra()** para por os dados na **Intent**.

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

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

A seguir invocamos a Activity **LoginActivity**.

## Implementando o código na Activity : RegistrarActivity

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

```

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

```

```

using SQLite;
using System;
using System.IO;

namespace App.Login_SQLite
{
    [Activity(Label = "RegistrarActivity")]
    public class RegistrarActivity : Activity
    {
        EditText txtNovoUsuario;
        EditText txtSenhaNovoUsuario;
        Button btnCriarNovoUsuario;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.NovoUsuario);

            btnCriarNovoUsuario = FindViewById<Button>(Resource.Id.btnRegistrar);
            txtNovoUsuario = FindViewById<EditText>(Resource.Id.txtNovoUsuario);
            txtSenhaNovoUsuario = FindViewById<EditText>(Resource.Id.txtSenhaNovoUsuario);

            btnCriarNovoUsuario.Click += BtnCriarNovoUsuario_Click;
        }

        private void BtnCriarNovoUsuario_Click(object sender, System.EventArgs e)
        {
            try
            {
                string dpPath = Path.Combine(System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal), "Usuario.db3");
                var db = new SQLiteConnection(dpPath);

                db.CreateTable<Login>();

                Login tblogin = new Login();
                tblogin.usuario = txtNovoUsuario.Text;
                tblogin.senha = txtSenhaNovoUsuario.Text;

                db.Insert(tblogin);

                Toast.MakeText(this, "Registro incluído com sucesso...", ToastLength.Short).Show();
            }
            catch (Exception ex)
            {
                Toast.MakeText(this, ex.ToString(), ToastLength.Short).Show();
            }
        }
    }
}

```

Neste código no evento **Click** do botão **Registrar**, obtemos o banco de dados e a tabela **Login**, criamos uma instância da classe **Login** e atribuímos os valores informados nas **EditText**.

Para persistir os dados na tabela usando o método **Insert()** do **SQLite**.

## Implementando o código na Activity : LoginActivity

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

```

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

namespace App.Login_SQLite
{
    [Activity(Label = "LoginActivity")]
    public class LoginActivity : Activity
    {
        TextView txtTextoLogin;

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

            SetContentView(Resource.Layout.Login);

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

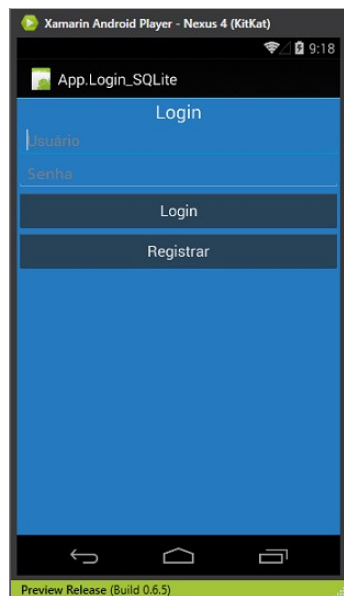
```

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

Primeiro estamos obtendo o texto colocado em '**nome**'; se os dados existirem eles serão exibidos no controle **txtTextoLogin**.

Executando o projeto iremos obter o seguinte resultado:

1- A view **Main.axml** sendo apresentada com opções login e/ou registro de novo usuário:



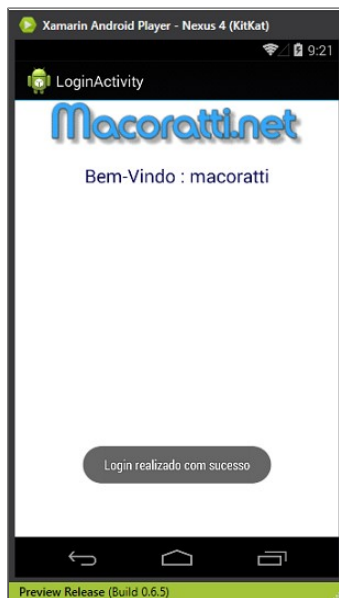
2- A view **NovoUsuario.axml** sendo apresentada para o registro de um novo usuário:



3- A view de **Main.axml** sendo apresentada e o login sendo realizado sem sucesso:



4- A view de **Login.axml** sendo apresentada para o login realizado com sucesso:



Pegue o projeto aqui : [App.Login\\_SQLite.zip](#) (sem as referências)

Deus nunca foi visto por alguém. O Filho unigênito (Jesus), que está no seio do Pai, esse o revelou.  
João 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 - 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](#) 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/api/type/Android.App.AlertDialog/>
- [Xamarin Android - Tratando eventos de forma declarativa](#)
- [Seção Mobile/Xamarin do site Macoratti .net](#)

---

[José Carlos Macoratti](#)