



Xamarin Android - Exibindo dados de um banco de dados SQLite existente



Neste artigo vou mostrar como exibir os dados de um banco de dados SQLite existente em uma aplicação Xamarin Android.



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

O Android oferece várias maneiras de armazenar e recuperar dados de um banco de dados e o SQLite é um banco de dados muito leve que vem com o sistema operacional Android.

Eu já mostrei como fazer o CRUD usando o banco de dados SQLite em uma aplicação Xamarin Android neste artigo : [Xamarin Android - CRUD Básico com SQLite - Macoratti.net](#)

Hoje veremos como acessar dados de um banco de dados SQLite já existente. Então primeiro vamos criar um banco de dados SQLite e uma tabela, usando o **SQLite DB Browser** e depois vamos incluir alguns dados na tabela para a seguir criar a aplicação Android para acessar esses dados.

O SQLite DB Browser que iremos usar para criar o banco de dados SQLite pode ser obtido neste link : <http://sqlitebrowser.org/>

O **DB Browser** para SQLite é uma ferramenta de alta qualidade, visual e de código aberto para criar, projetar e editar arquivos de banco de dados compatíveis com o SQLite.

Ele pode ser usado por desenvolvedores que desejam criar bancos de dados, pesquisar e editar dados. Ele usa uma interface familiar semelhante a uma planilha, e você não precisa aprender comandos SQL complicados.

A seguir um resumo sobre os controles e assistentes disponíveis na ferramenta:

- Criar e compactar arquivos de banco de dados
- Cria, define, modifica e exclui tabelas
- Cria, define e exclui índices
- Navega, edita, adiciona e exclui registros
- Pesquisa registros
- Importar e exportar registros como texto
- Importar e exportar tabelas de/para arquivos CSV
- Importe e exporte bancos de dados de/para arquivos dump SQL
- Emita consultas SQL e inspeciona os resultados
- Examina um log de todos os comandos SQL emitidos pelo aplicativo

Agora, sem mais delongas, vamos ao trabalho.

Nota: Esse projeto foi adaptado do original em : <http://www.c-sharpcorner.com/article/xamarin-android-working-with-existing-db/>

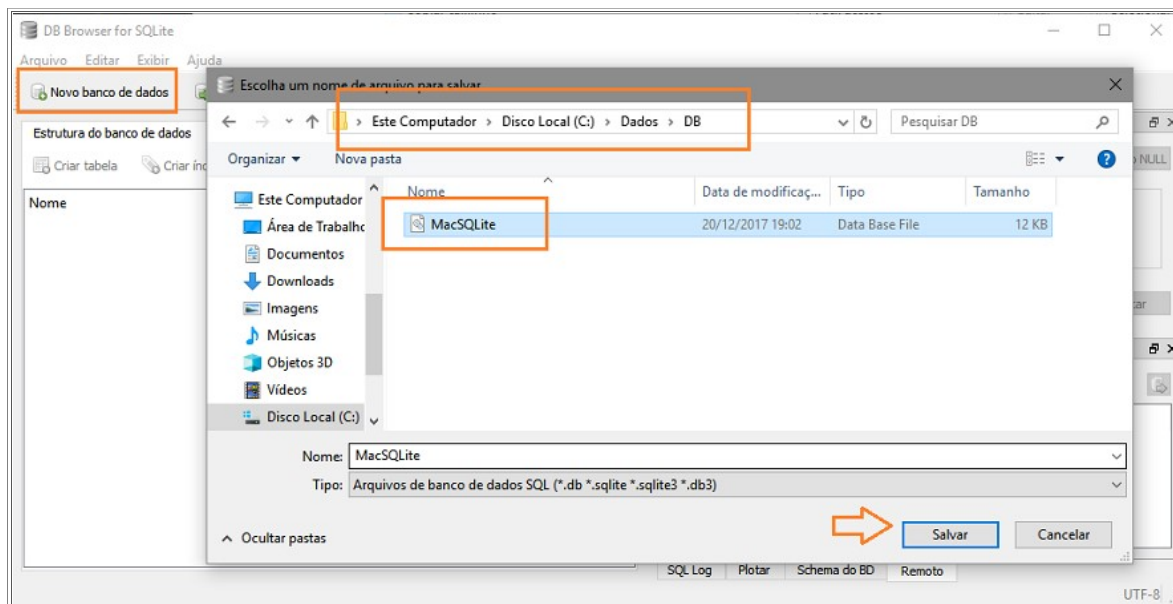
Recursos usados :

- [Visual Studio 2017 Community](#)
- [SQLite DB Browser](#)
- [SQLite.Net-PCL NuGet Package](#)

Criando o banco de dados SQLite com o SQLite DB Browser

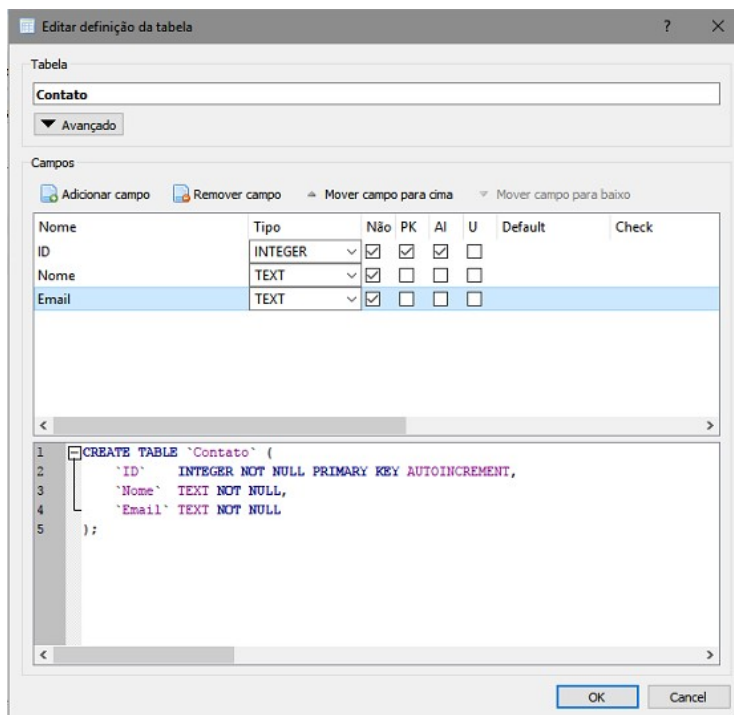
Abra o **SQLite DB Browser** e clique em **Novo banco de dados** e a seguir escolha um local e informe o nome do banco de dados que deseja criar.

Eu vou criar um banco de dados na pasta : [c:\dados\DB\](#) chamado **MacSqlite** :

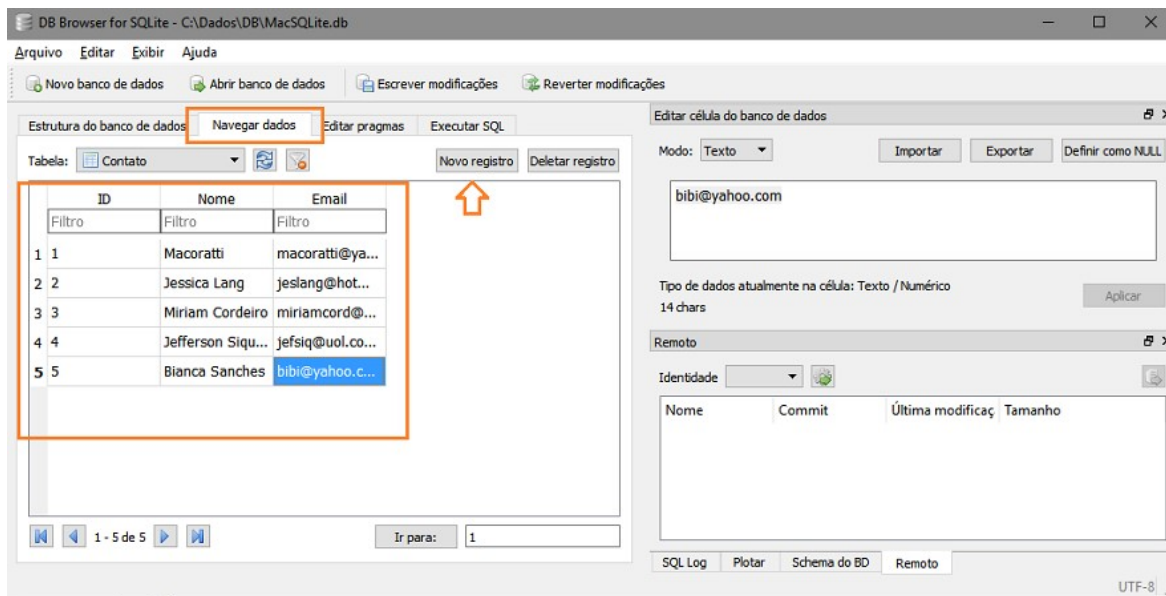


A seguir vamos criar uma tabela chamada **Contato** com os seguintes campos:

- **ID - INTEGER, Primary Key - Auto-Increment**
- **Nome - TEXT**
- **Email - TEXT**



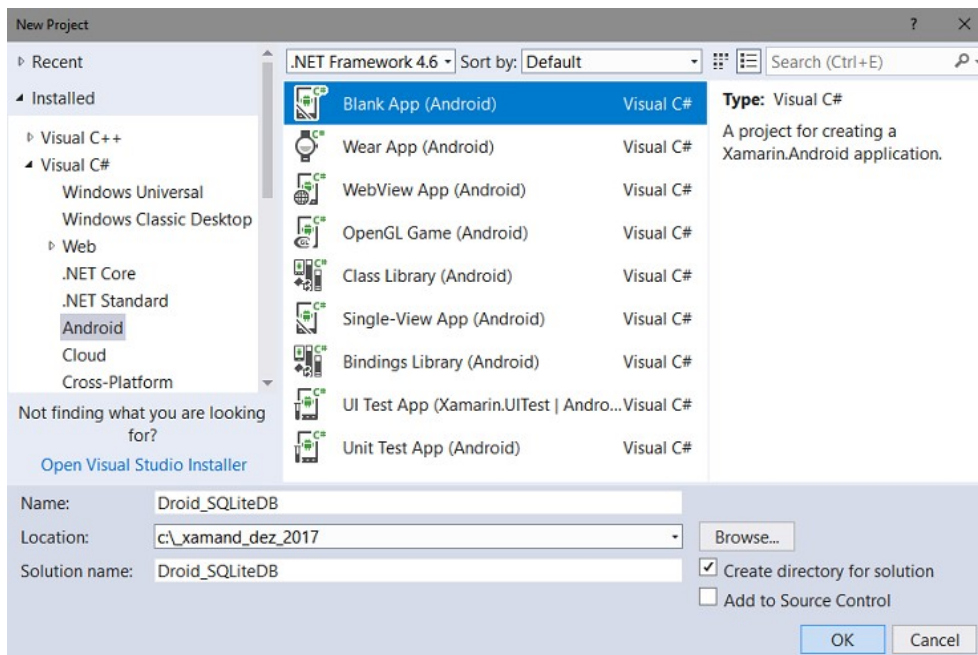
Para encerrar essa etapa inclua alguns dados na tabela **Contato**:



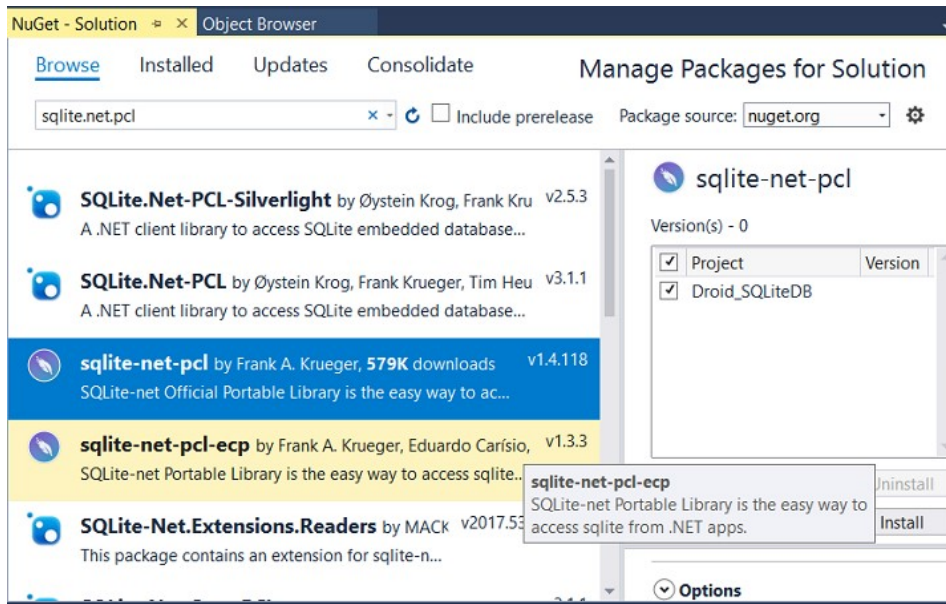
Criando a aplicação Xamarin Android

Abra o VS 2017 Community e clique em **Visual C# -> Android**;

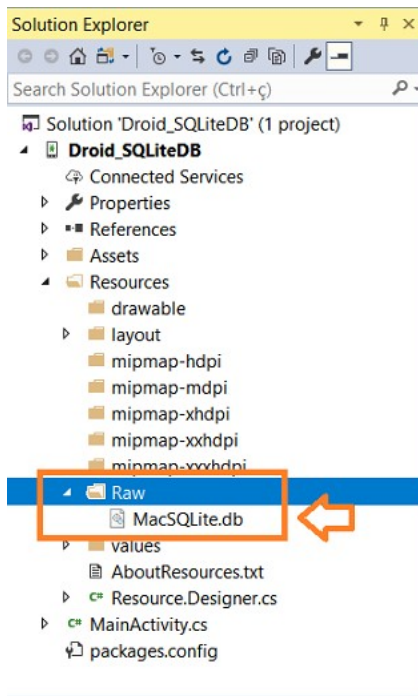
Escolha o template **Blank App (Android)** e informe o nome **Droid_SQLiteDB** :



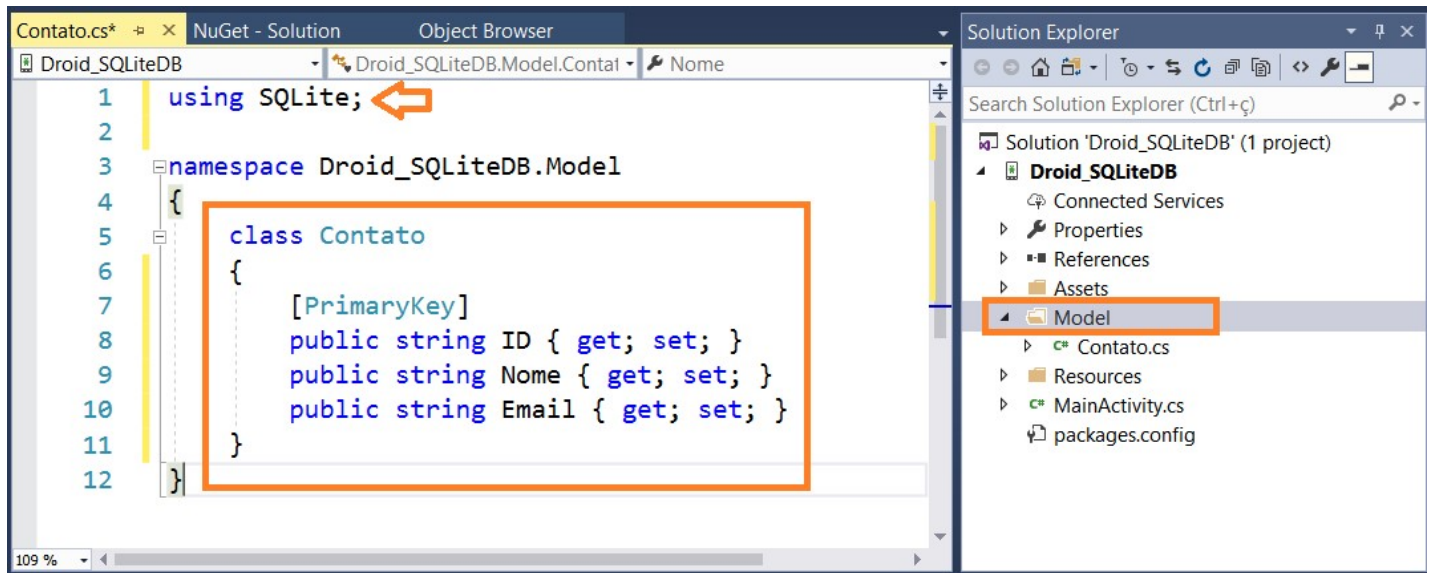
A seguir inclua o pacote **sqlite-net-pcl** no projeto via menu **Tools** em **Manage Nuget Packages for Solution**:



Agora abra o projeto e crie uma pasta chamada **Raw** dentro da pasta **Resources** e copie o banco de dados SQLite criado para esta pasta:



Crie uma pasta Model no projeto e nesta pasta crie uma classe Contato incluindo o código abaixo:



Abra o arquivo de layout **Main.axml** na pasta **Resources/Layout** e inclua o código abaixo :

```

<?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">
    <Button
        android:text="Carregar Dados"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btnCarregaDados" />
    <LinearLayout
        android:orientation="vertical"
        android:minWidth="25px"
        android:minHeight="25px"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/container" />
</LinearLayout>

```



Esta página de layout será a página inicial da aplicação apresentando um botão para carregar os dados da tabela **Contato**.

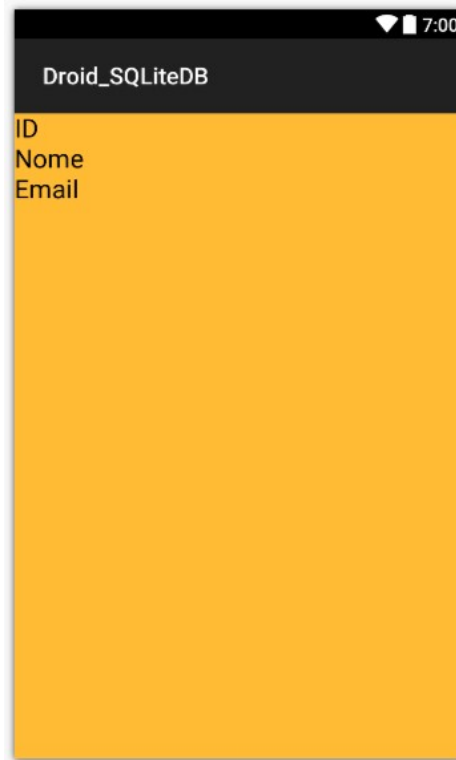
Vamos incluir um novo arquivo de layout chamado **row.axml** na pasta **Resources/Layout** onde vamos definir o layout para exibir as informações obtidas do banco de dados.

No menu **Project** clique em **Add New Item** e a seguir em **Android Layout** e informe o nome **row.axml** :

A seguir 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"
    android:background="@android:color/holo_orange_light">
    <TextView
        android:text="ID"
        android:textColor="White"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"

        android:layout_height="wrap_content"
        android:id="@+id/txtID" />
    <TextView
        android:text="Nome"
        android:textColor="White"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/txtNome" />
    <TextView
        android:text="Email"
        android:textColor="White"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/txtEmail" />
</LinearLayout>
```



Temos no código 3 **TextViews** onde vamos exibir o **ID**, o **Nome** e o **Email** do contato obtido da tabela **Contato**.

Criando o código para copiar e acessar o banco de dados

Vamos agora criar uma rotina que vai copiar o banco de dados que criamos e que colocamos na pasta **Raw** para o Android.

Crie uma pasta **Service** no projeto e nesta pasta crie uma classe chamada **DBHelper** com o seguinte código:

```
using Android.Content;
using Android.Database.Sqlite;
using System;
using System.IO;

namespace Droid_SQLiteDB.Service
{
    public class DBHelper : SQLiteOpenHelper
    {
        private static string DB_PATH = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
        private static string DB_NAME = "MacSQLite.db";
        private static int VERSION = 1;
        private Context context;

        public DBHelper(Context context) : base(context, DB_NAME, null, VERSION)
        {
            this.context = context;
        }
        private string GetSQLiteDBPath()
        {
            return Path.Combine(DB_PATH, DB_NAME);
        }
        public override SQLiteDatabase WritableDatabase
        {
            get
```



```
{
    return CriaSQLiteDB();
}
}
private SQLiteDatabase CriaSQLiteDB()
{
    SQLiteDatabase sqlitedb = null;
    string path = GetSQLiteDBPath();
    Stream streamSQLite = null;
    FileStream streamWriter = null;
    Boolean isSQLiteInit = false;
    try
    {
        if (File.Exists(path))
            isSQLiteInit = true;
        else
        {
            streamSQLite = context.Resources.OpenRawResource(Resource.Raw.MacSQLite);
            streamWriter = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Write);
            if (streamSQLite != null && streamWriter != null)
            {
                if (CopiaSQLiteDB(streamSQLite, streamWriter))
                    isSQLiteInit = true;
            }
        }
        if (isSQLiteInit)
            sqlitedb = SQLiteDatabase.OpenDatabase(path, null, DatabaseOpenFlags.OpenReadOnly);
    }
    catch { }
    return sqlitedb;
}
private bool CopiaSQLiteDB(Stream streamSQLite, FileStream streamWriter)
{
    bool isSuccess = false;
    int lenght = 256;
    Byte[] buffer = new Byte[lenght];
    try
    {
        int bytesRead = streamSQLite.Read(buffer, 0, lenght);
        while (bytesRead > 0)
        {
            streamWriter.Write(buffer, 0, bytesRead);
            bytesRead = streamSQLite.Read(buffer, 0, lenght);
        }
        isSuccess = true;
    }
    catch { }
    finally
    {
        streamSQLite.Close();
        streamWriter.Close();
    }
    return isSuccess;
}
public override void OnCreate(SQLiteDatabase db)
{
}
public override void OnUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
}
}
```

A classe **DBHelper** herda de **SQLiteOpenHelper** que é uma classe de ajuda para gerenciar a criação o gerenciamento de versão do banco de dados. Usamos essa classe para ler o arquivo SQLite e copiá-lo para [System.Environment.SpecialFolder.Personal](#).

Acessando e exibindo os dados do SQLite

Para concluir abra o arquivo **MainActivity** e inclua o código abaixo:

```
using Android.App;
using Android.Widget;
using Android.OS;
using Android.Database.Sqlite;
using Droid_SQLiteDB.Model;
using Droid_SQLiteDB.Service;
using System.Collections.Generic;
using Android.Database;
using Android.Views;
using Android.Content;

namespace Droid_SQLiteDB
{
    [Activity(Label = "Droid_SQLiteDB", MainLauncher = true)]
    public class MainActivity : Activity
    {
        DBHelper db;
        SQLiteDatabase sqliteDB;
        LinearLayout container;
        Button btnGetDados;
        List<Contato> listaContatos = new List<Contato>();

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

            // Set our view from the "main" layout resource
            SetContentView(Resource.Layout.Main);
            db = new DBHelper(this);
            sqliteDB = db.WritableDatabase;
            container = FindViewById<LinearLayout>(Resource.Id.container);
            btnGetDados = FindViewById<Button>(Resource.Id.btnCarregaDados);
            btnGetDados.Click += delegate
            {
                ObtemDados();
            };
        }
        private void ObtemDados()
        {
            ICursor dadosSelecionados = sqliteDB.RawQuery("select * from Contato", new string[] { });
            if (dadosSelecionados.Count > 0)
            {
                dadosSelecionados.MoveToFirst();
                do
                {
                    Contato contato = new Contato();
                    contato.ID = dadosSelecionados.GetString(dadosSelecionados.GetColumnIndex("ID"));
                    contato.Nome = dadosSelecionados.GetString(dadosSelecionados.GetColumnIndex("Nome"));
                    contato.Email = dadosSelecionados.GetString(dadosSelecionados.GetColumnIndex("Email"));
                    listaContatos.Add(contato);
                }
                while (dadosSelecionados.MoveNext());
                dadosSelecionados.Close();
            }
        }
    }
}
```



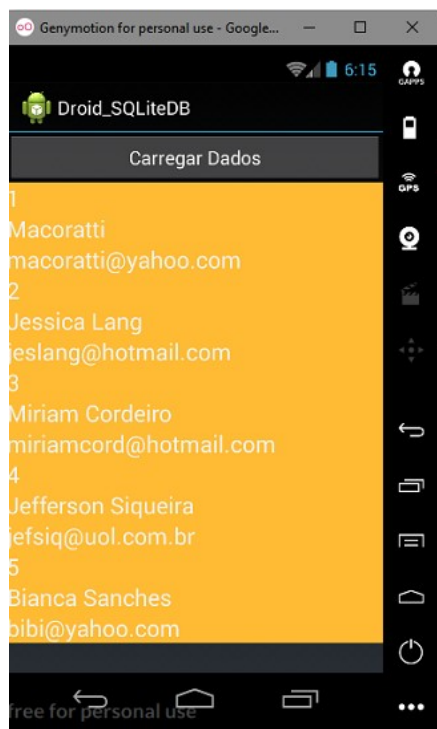
```

    }
    foreach (var item in listaContatos)
    {
        LayoutInflater inflater = (LayoutInflater)BaseContext.GetService(Context.LayoutInflaterService);
        View addView = inflater.Inflate(Resource.Layout.row, null);
        TextView txtID = addView.FindViewById<TextView>(Resource.Id.txtID);
        TextView txtNome = addView.FindViewById<TextView>(Resource.Id.txtNome);
        TextView txtEmail = addView.FindViewById<TextView>(Resource.Id.txtEmail);
        txtID.Text = item.ID;
        txtNome.Text = item.Nome;
        txtEmail.Text = item.Email;
        container.AddView(addView);
    }
}
}
}

```

Neste código acessamos o banco de dados SQLite e obtemos todos os dados da tabela **Contato** (`select * from Contato`) exibindo-os no layout row.xml.

Executando o projeto iremos obter o seguinte resultado:



Essa abordagem é uma solução adequada apenas para banco de dados pequenos.

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

"E estava ali um homem que, havia trinta e oito anos, se achava enfermo. Jesus disse-lhe: Levanta-te, toma o teu leito, e anda. Logo aquele homem ficou são; e tomou o seu leito, e andava. E aquele dia era sábado."
João 5:5-9

[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#](#)
- [Seção C# do site Macoratti.net](#)
- [Super DVD C#](#)
- [Super DVD Visual Basic](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [Xamarin Android - Primeira Aplicação Android : Activity ... - Macoratti.net](#)
- [Xamarin Android - Enviando notificações - I - Macoratti.net](#)
- [Xamarin Android - Usando a Câmera - Macoratti.net](#)
- [Xamarin Android - Globalização de aplicações Android - Macoratti.net](#)
- [Xamarin Android - Enviando SMS - Macoratti.net](#)
- [Mobile / Xamarin Android - Macoratti.net](#)
- <https://blog.reigndesign.com/blog/using-your-own-sqlite-database-in-android-applications/>

[José Carlos Macoratti](#)