

Macoratti.net - C# - Acessando MySQL

Vamos acessar o **MySQL** usando **C#** ? Então vamos lá... 😊

Eu vou usar os seguintes recursos:

- 1- [MySQL 5.0 \(Community Server\)](#) - Baixe e instale usando as configurações padrão sugeridas pelo assistente de instalação;
- 2- [MySQL Administrator e MySQL Query Browser](#) - Baixe e instale usando o assistente;
- 3- [SharpDevelop 2.2](#) ou [Visual C# Express Edition](#)

Para efetuar a conexão usando **C#** vamos usar provedor ADO .NET para **MySQL** :

- 4- [Download Connector/Net 5.1](#) (Podemos usar também a versão 5.2)

Criando o banco de dados e a tabela

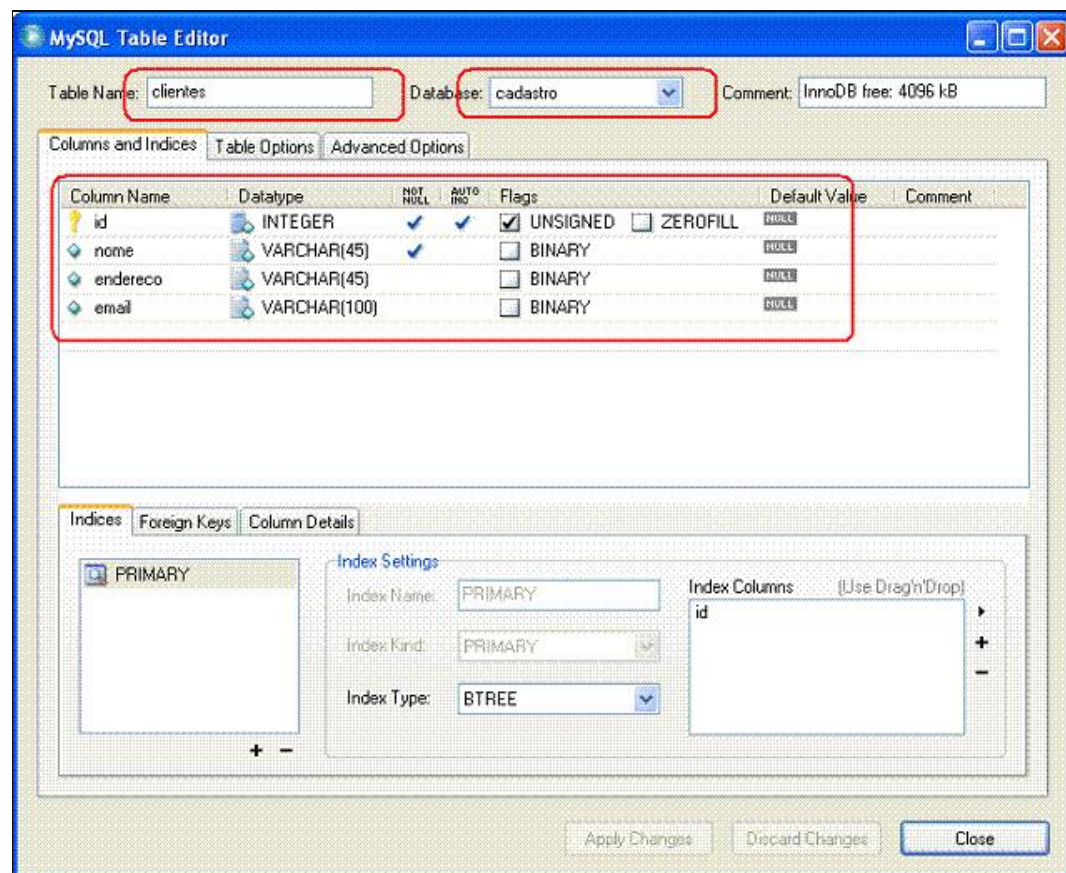
Após fazer o download e instalar as ferramentas indicadas vamos criar um novo banco de dados chamado **Cadastro** no **MySQL**.

Abra o **MySQL Administrator** e clique em **Catalogs**;

A seguir clique com o botão direito do mouse na área onde são exibidos os banco de dados e selecione **Create New Schema**;

Informe o nome do banco de dados e a seguir clique no botão **Create Table**;

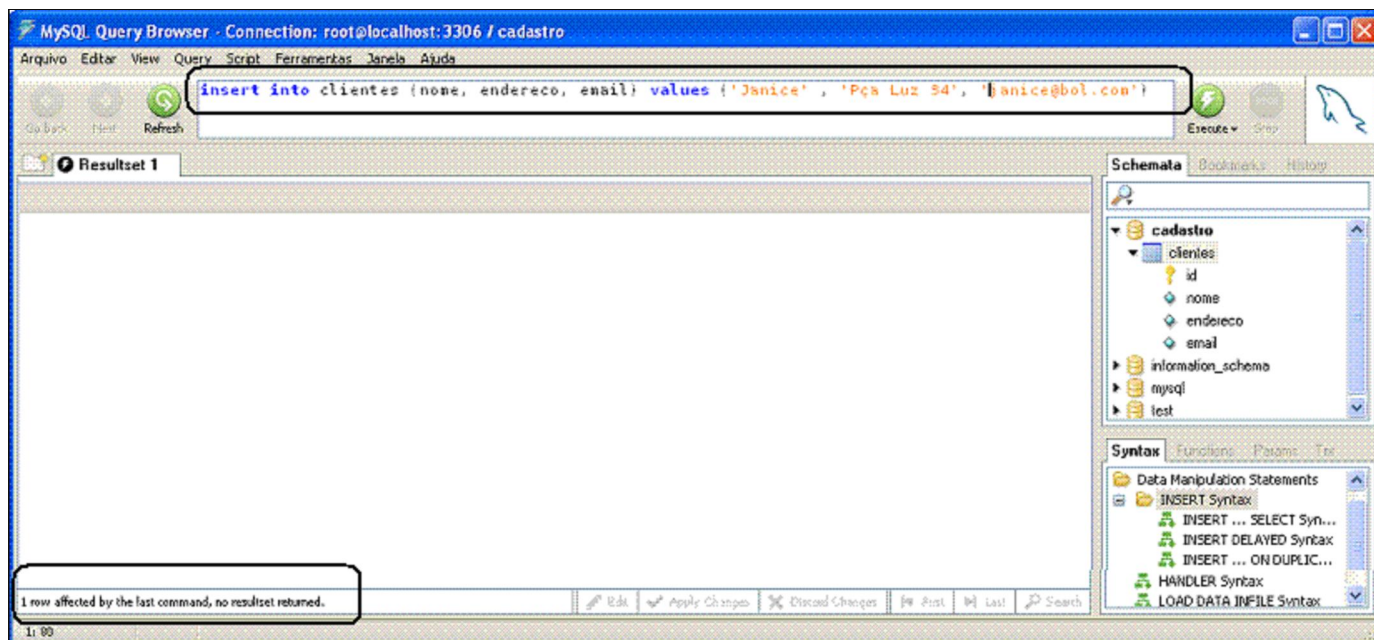
A seguir informe o nome da tabela **Clientes** e defina as colunas: **id** , **nome**, **endereço** e **email** conforme abaixo;



Para incluir alguns dados na tabela abra o **MySQL Query Browser** e digite a instrução **INSERT INTO** conforme figura abaixo e clique em **Execute**;

INSERT INTO Clientes(nome,endereco, email) values ('Janice', 'Pça da Luz, 54', 'janice@bol.com.br')

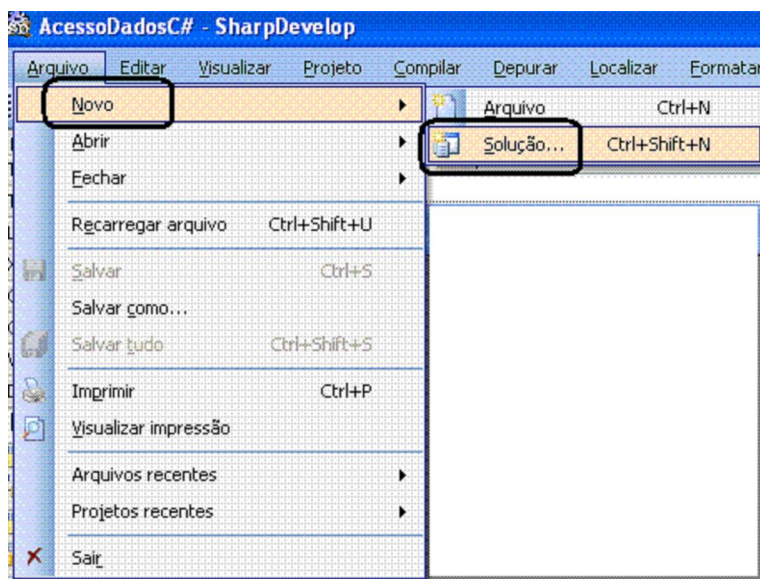
O campo **id** foi definido como chave primária e incremental de forma que o seu valor é atribuído pelo **MySQL**;



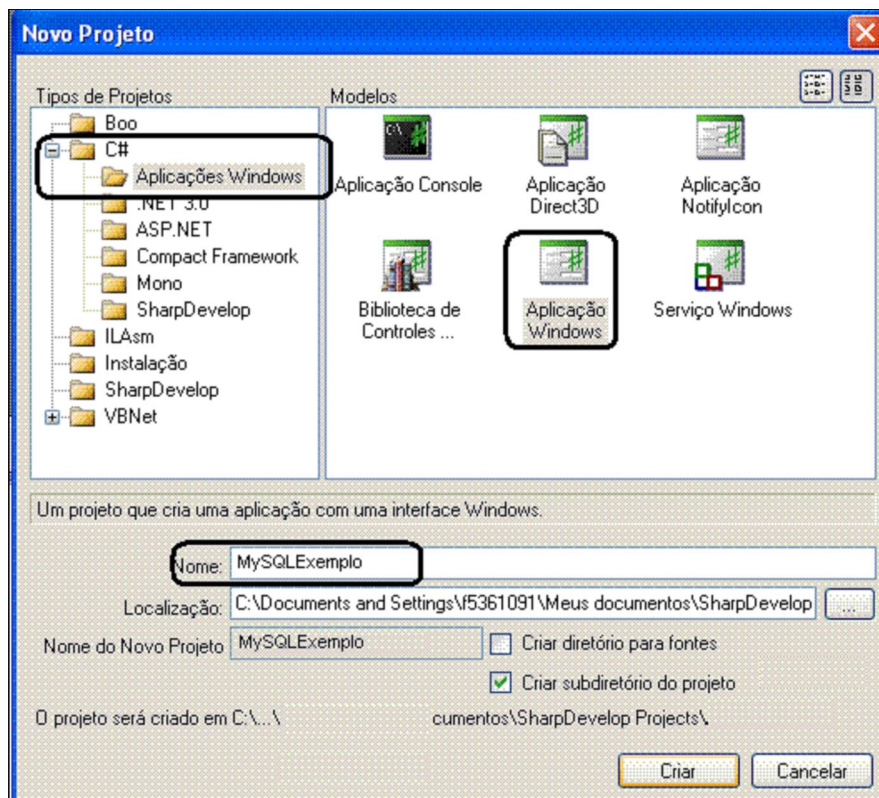
Ao final teremos o banco de dados **Cadastro** e a tabela **Clientes** com alguns dados no **MySQL**. Estamos prontos para começar.

Vamos agora criar uma nova solução chamada **MySQLExemplo** usando o **SharpDevelop**.

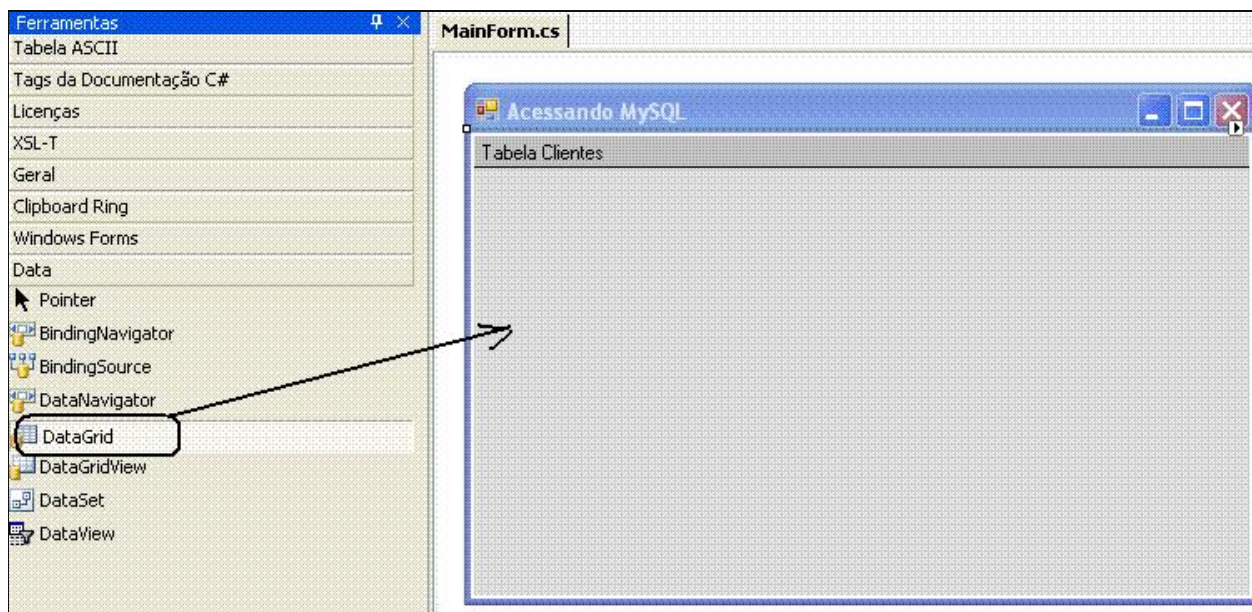
Abra o **SharpDevelop** e no menu **Arquivo** selecione **Novo** e em seguida **Solução**;



Na janela **Novo Projeto** selecione **C#->Aplicações Windows** e na janela **Modelos** marque **Aplicação Windows** e informe o nome da solução, no meu caso: **MySQLExemplo**;



No formulário padrão **MainForm.cs** vamos incluir um controle **DataGrid** a partir da guia **Data** na janela de Ferramentas:



Na janela de código vamos primeiro declarar os namespaces:

```
using System.Data;
using MySql.Data.MySqlClient;
```

A seguir vamos incluir o código para fazer a conexão com o banco de dados **Cadastro no MySQL**, acessar a tabela **Clientes** e exibir os seus dados no **DataGrid**;

```
public class MainForm : System.Windows.Forms.Form
{
    private System.Windows.Forms.DataGrid mDataGrid;
    private MySqlConnection mConn;
    private MySqlDataAdapter mAdapter;
    private DataSet mDataSet;
    public MainForm()
    {
```

```
InitializeComponent();

//define o dataset
mDataSet = new DataSet();

//define string de conexao e cria a conexao
mConn = new MySqlConnection("Persist Security Info=False;server=localhost;
database=Cadastro;uid=root;
server=localhost;database=Cadastro;uid=root;pwd=xxxx");

try{
    //abre a conexao
    mConn.Open();
}
catch(System.Exception e)
{
    MessageBox.Show(e.Message.ToString());
}

//verificva se a conexão esta aberta
if (mConn.State == ConnectionState.Open)
{
    //cria um adapter usando a instrução SQL para acessar a tabela Clientes
    mAdapter = new MySqlDataAdapter("SELECT * FROM Clientes", mConn);
    //preenche o dataset via adapter
    mAdapter.Fill(mDataSet, "Clientes");
    //atribui a resultado a propriedade DataSource do DataGridView
    mDataGrid.DataSource = mDataSet;
    mDataGrid.DataMember = "Clientes";
}
}
```

No código acima eu vou falar um pouco sobre a string de conexão usada. Ele é definida dessa forma para o provedor **MySQL Connector/Net (.NET)**.

```
mConn = new MySqlConnection( Persist Security Info=False;
"Server=localhost;" +
"DataBase=Cadastro;" +
"Uid=root;" +
"Pwd=xxx;");
```

Um objeto **MySqlConnection** representa uma sessão para uma fonte de dados **MySQL**. Quando criamos uma instância **MySqlConnection** todas as propriedades são definidas para os seus valores iniciais. A seguir temos os valores padrão no construtor:

Propriedades	Valor Inicial
ConnectionString	empty string ("")
ConnectionTimeout	15
Database	empty string ("")
DataSource	empty string ("")
ServerVersion	empty string ("")

O atributo **Persist Security Info** para uma string de conexão especifica se a fonte de dados pode persistir uma informação segura como as credenciais de autenticação do usuário. O valo padrão é mantido como **False**.

Se **Persist Security Info** for definido como **False** e a conexão foi aberta , o valor retornado pela string de conexão não irá exibir a chave nem a senha.

Se o valor for definido como **True** a informação da conexão - incluindo a senha - pode ser obtida pela consulta a conexão , permitindo assim uma forma de acesso não confiável as informações do usuário quando a conexão for persistida.

A propriedade **Server(DataSource)** indica o nome ou endereço de rede da instância do **MySQL** com o qual será

feita a conexão. Múltiplas instâncias podem ser definidas separadas por &.

A propriedade **DataBase** define o nome do banco de dados atual a ser usado após a conexão ser aberta. Ela não é atualizada dinamicamente, desta forma, se você alterar o banco de dados usando uma instrução SQL poderá ocorrer um erro. Para alterar o banco de dados use o método **ChangeDataBase** para refletir o novo banco de dados.

A propriedade **Uid** representa a conta que será usada para acessar o MySQL.

A propriedade **Pwd** representa a senha da conta que esta sendo usada para o login no MySQL.

Após encerrar o trabalho com a conexão o objeto **MySqlConnection** não é fechado, e , por isso você deve fechar a conexão explicitamente chamando **MySqlConnection.Close** ou **MySqlConnection.Dispose**.

Para fechar a conexão sobrescrevemos o método **Dispose** conforme abaixo:

```
protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        if (mConn.State ==
            ConnectionState.Open)
        {
            mConn.Close();
        }
    }
    base.Dispose(disposing);
}
```

Ao usar o comando **Open** o **MySqlConnection** tenta obter uma conexão a partir do [pool de conexão](#) , se não existir uma conexão disponível, será estabelecida uma nova conexão com uma instância do MySQL.

```
mConn.Open();
```


Ao executar o projeto iremos obter o seguinte resultado no controle **DataGrid**;



id	nome	endereco	email
1	Macoratti	Rua Teste 100	macoratti@yahoo.
2	Jefferson	Rua Luz 50	jefferon@yahoo.co
3	Miriam	Av. Doze , 109	miriam@yahoo.co
4	Janice	Pça Luz 34	janice@bol.com
*			

Este foi apenas uma pequena introdução ao acesso a dados com **MySQL** e **C#**. Aguarde em breve artigos mais avançados...

Um bom estudo para você... 😊

Pegue o projeto completo aqui:  [MySQLExemplo.zip](#)

Eu sei é apenas C# mas eu gosto...

referências:

- [MySQL - fundamentos SQL](#)

- [ASP.NET - Acessando o MySQL em páginas ASP.NET](#)
- [VB e MYSQL - Gerenciando dados](#)
- [Visual Basic - Acessando MySQL com ADO](#)
- [Acessando o MySQL](#)
- [VB 2005 - Acessando o MySQL 5.0](#)
- [Instalando e usando o ODBC .Net Data Provider](#)
- [Criando uma conexão Genérica com banco de dados](#)
- [MySQL - Importando e Exportando tabelas](#)
- [Novos provedores de dados .NET](#)

José Carlos Macoratti