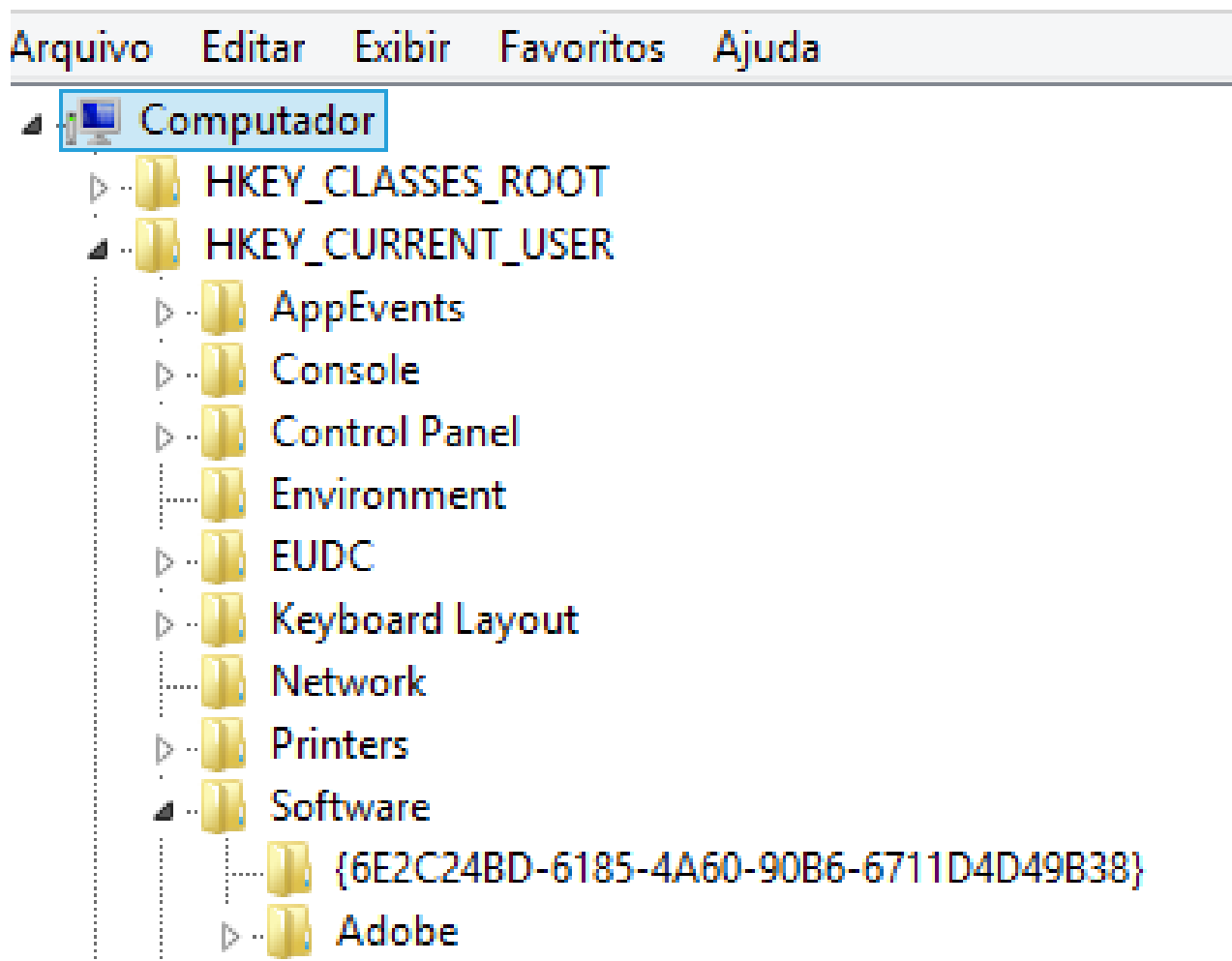


## Curso Técnico em Informática

### ACESSO AO REGISTRO DO WINDOWS

Quando precisamos gravar algum tipo de informação onde o banco de dados não pode ser utilizado, utilizamos outras formas, como um arquivo ".txt", ".ini" entre outros. Podemos também, dependendo da informação que será guardada, utilizar o próprio registro do Windows. Um exemplo para essa gravação é o caminho de um banco de dados a ser usado no MYSQL. Em nosso exemplo a informação será guardada na chave:

#### HKEY\_CURRENT\_USER\Software

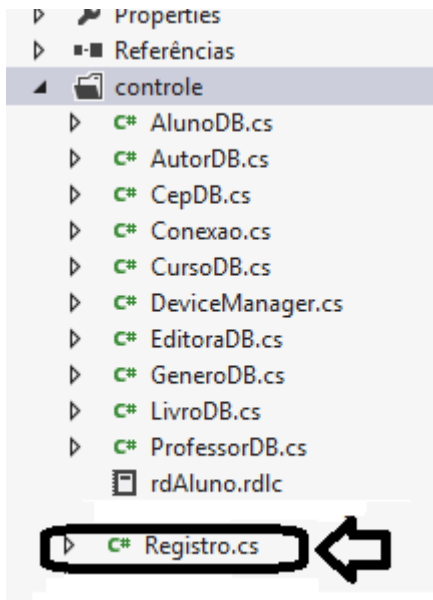


Note que a chave deve ser criada para cada usuário a utilizar o computador, já que a chave será registrada no usuário padrão do sistema. A classe do C# responsável pela manipulação do registro é a classe Registry. Nela temos dois métodos principais:

**CreateSubKey** – usado para criar chaves no registro.

**OpenSubKey** – usado para ler chave já criadas no registro.

Acrescente a pasta controle uma nova classe chamada Registro:

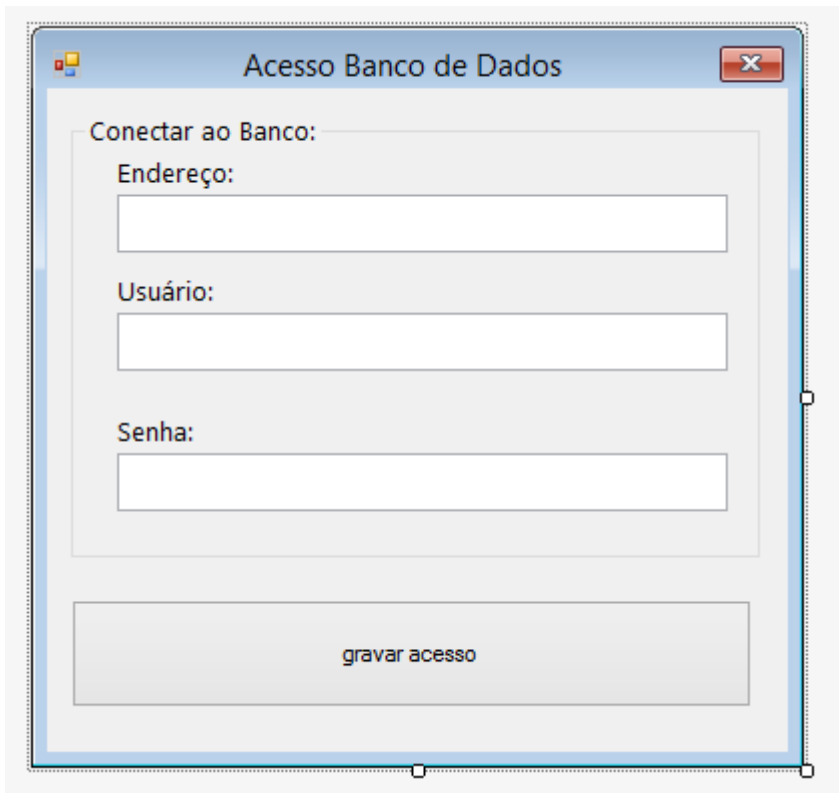
**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

Vamos codificar dois métodos, um que cria chaves de controle dentro do Windows e outro que vai ler dados destas chaves. A classe vai ser do tipo estática, assim não precisaremos criar objetos para usar os métodos.

```
static class Registro
{
    public static String lerRegistro(String caminho, String chave)
    {
        string PATH = @caminho;
        RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(PATH);
        if (registryKey != null && registryKey.GetValue(chave) != null)
            return registryKey.GetValue(chave).ToString();
        else return "";
    }

    public static void criaRegistro(String caminho, String chave, String valor)
    {
        Microsoft.Win32.RegistryKey registryKey =
            Registry.CurrentUser.CreateSubKey(@caminho);
        registryKey.SetValue(chave, valor);
        registryKey.Close();
    }
}
```

Vamos utilizar esta classe registro criando uma entrada para ler e salvar os dados de nosso servidor de banco de dados, no caso IP, login e senha. Para isso crie um novo Formulário chamado **FormBancoDados.cs** em visão:

**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

The image shows a Windows-style dialog box titled "Acesso Banco de Dados". Inside the dialog, there is a group box labeled "Conectar ao Banco:". Within this group box, there are three text input fields. The first is labeled "Endereço:", the second "Usuário:", and the third "Senha:". Below the group box, outside of it, is a button labeled "gravar acesso". The dialog box has a standard Windows title bar with a close button (X) in the top right corner.

Nome dos campos:

```
.GroupBox gbLogin;  
.TextBox Servidor;  
.Label label3;  
.TextBox txtPassword;  
.TextBox txtUser;  
.Label label2;  
.Label label1;  
.Button btnConectar;
```

Vamos codificar o botão para gravar os dados de acesso ao banco de dados mysql:

---

**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

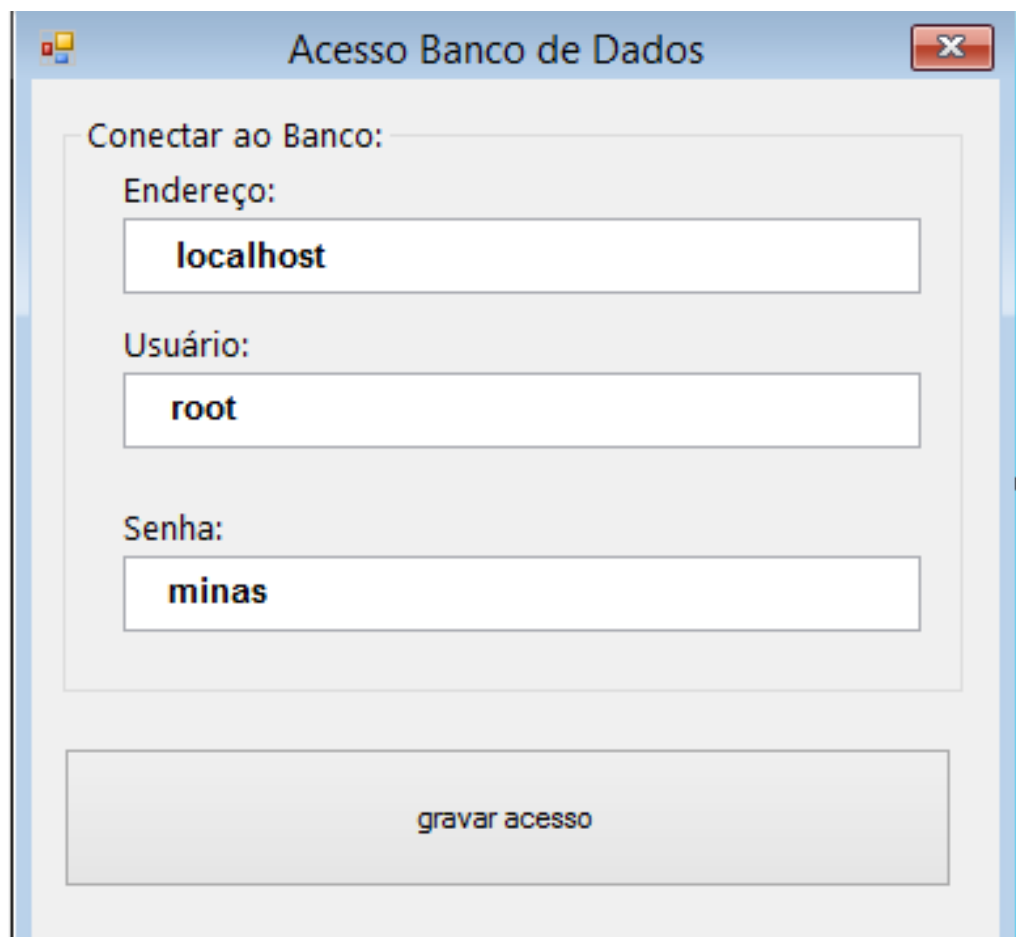
```
private void btnConectar_Click(object sender, EventArgs e)
{
    // vamos assumir que o banco de dados sempre vai se chamar
    // biblioteca

    controle.Registro.criaRegistro("biblioteca", "server", Servidor.Text);
    controle.Registro.criaRegistro("biblioteca", "user", txtUser.Text);
    controle.Registro.criaRegistro("biblioteca", "password", txtPassword.Text);
    MessageBox.Show("Dados gravados com sucesso....");
    this.Dispose();
}
```

Agora no início do evento load em **FormMDI** vamos chamar o formulário **FormBancoDados** para setar o acesso ao banco de dados MySQL caso a chave não exista no registro. se a chave não existir no registro e gravar as configurações:

```
private void FormMDI_Load(object sender, EventArgs e)
{
    string server = controle.Registro.lerRegistro("biblioteca", "server");
    // se não há um servidor
    if (server == "") {
        // chama a configuracao
        visao.FormBancoDados fr = new visao.FormBancoDados();
        fr.ShowDialog();
        // checa se salvou os dados
        if (controle.Registro.lerRegistro("biblioteca", "server") == "") {
            // encerra se não configurou
            System.Environment.Exit(0);
        }
    }
}
```

Note que usamos uma nova leitura do registro para verificar se o aluno gravou uma configuração válida, se não o programa é fechado.

**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

Para testar modifique todos os eventos de conexão nas classes DB para usar os valores do registro, por exemplo em **AlunoDB**:

```
class AlunoDB
{
    // Conexao con = new Conexao("localhost", "biblioteca", "root", "minas");
    // agora será assim:
    Conexao con = new Conexao(Registro.lerRegistro("biblioteca", "server"),
                              "biblioteca",
                              Registro.lerRegistro("biblioteca", "user"),
                              Registro.lerRegistro("biblioteca", "password")
    );

    public void Consultar(System.Windows.Forms.BindingSource bs)
    {
```

Perceba que estamos considerando que o nome do banco de dados é biblioteca, é possível gerenciar através de uma chave o nome do banco de dados, neste caso estamos assumindo um banco de dados padrão.

Nota: para remover uma chave do registro crie um método chamado apagar que utilize o seguinte comando:

---

**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

```
Registry.LocalMachine.DeleteSubKey(PATH);
```

Note que nós não estamos criptografando a senha no registro. Caso deseje use um método de criptografia para a senha antes de gravar a informação no registro.

## TESTANDO CONEXÃO DE REDE

Para testar a conexão de rede com o banco de dados mysql, iremos recorrer ao recurso de soquete, que permite acessar uma porta TCP ou UDP e verificar a sua conectividade usando a classe TcpClient. A definição da classe usará a seguinte sintaxe:

```
System.Net.Sockets.TcpClient client = new System.Net.Sockets.TcpClient();
```

Para isso vamos criar seguinte método no formulário FormMDI:

```
private void testarServidor()
{
    System.Net.Sockets.TcpClient client = new System.Net.Sockets.TcpClient();

    try
    {
        client.Connect(
            controle.Registro.lerRegistro("biblioteca", "server"), 3306);
    }

    catch (System.Net.Sockets.SocketException ex)
    {
        MessageBox.Show("O endereço do servidor Mysql está fora do ar."
            + ex.Message, "Alerta",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

    finally
    {
        client.Close();
    }
}
```

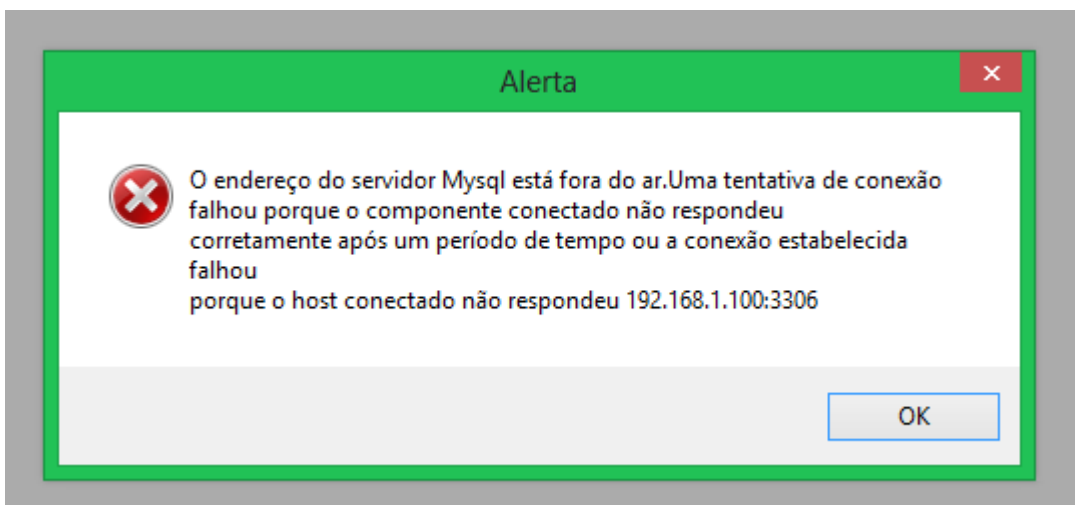
**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

No evento load chame o método:

```
private void FormMDI_Load(object sender, EventArgs e)
{
    testarServidor(); 
    string server = controle.Registro.lerRegistro("biblioteca", "server");

    // se não há um servidor
    if (server == "") {
        // chama a configuracao
        visao.FormBancoDados fr = new visao.FormBancoDados();
        fr.ShowDialog();
    }
}
```

Teste a conexão de rede (se o mysql estiver inativo a seguinte mensagem deve aparecer):



## TELA DE BOAS VINDAS EM C#

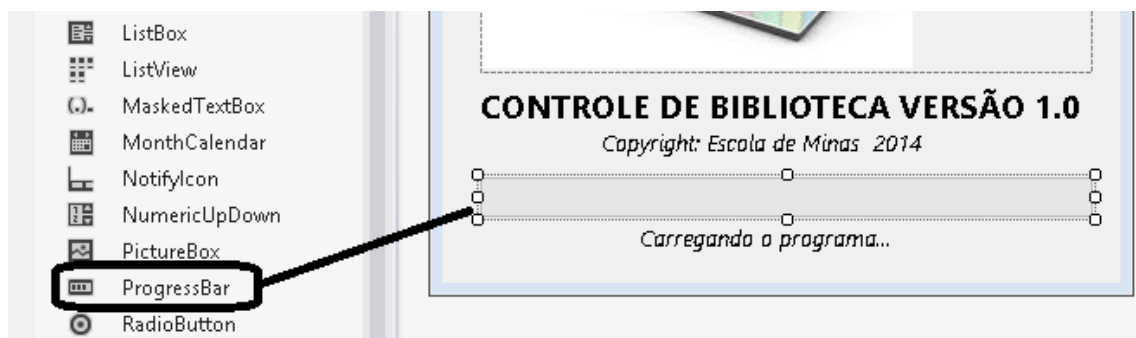
Acrescente um novo formulário chamado **FormWelcome.cs** e monte a seguinte visualização:

**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

Componentes a serem usados:

PictureBox pbLogo  
Label lbTitulo1  
Label lbTitulo2  
ProgressBar porcentual  
Label lbMensagem

Atenção ao percentual que deve ser uma barra de progressão:

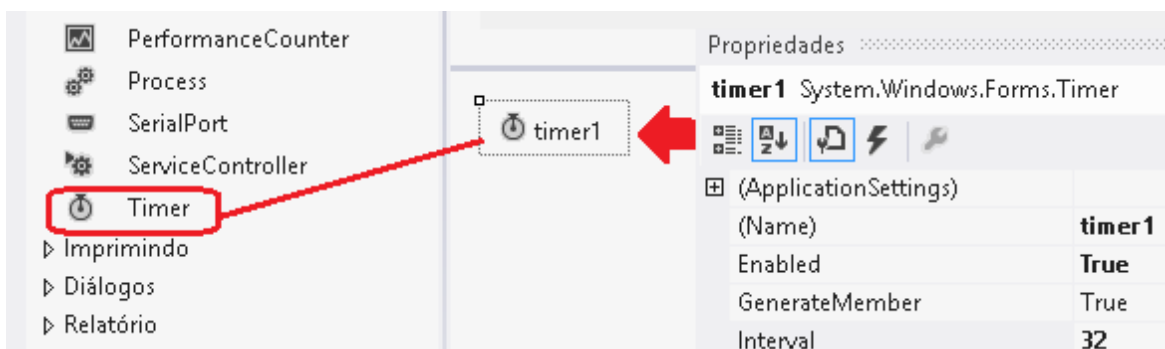


Mude o estilo da borda do formulário para **none** e a posição do formulário para centralizado:



**Etec Dr Demétrio Azevedo Junior – Itapeva-SP**

Acrescente ao formulário um Timer, para controlar a barra de progresso percentual e codifique o evento Tick da mesma e ajuste a propriedade **Interval** para 32 milissegundos e **Enabled** para **true**:



```
private void timer1_Tick(object sender, EventArgs e)
{
    // incrementa a barra de percentual
    porcentual.Increment(1);
    if (porcentual.Value == 100)
        timer1.Stop();
}
```

Em **FormMdi**, acrescente logo após o construtor a chamada a nossa tela de "boas vindas":

---

Etec Dr Demétrio Azevedo Junior – Itapeva-SP

```
public partial class FormMDI : Form
{
    public visao.FormAluno frAluno = null;

    // aqui a criação do formulário livro
    public visao.FormLivro frLivro = null;
    public visao.FormProfessor frProfessor = null;

    public FormMDI()
    {
        InitializeComponent();
    }
```

crie aqui o método



```
public void startBemVindo()
{
    Application.Run(new visao.FormWelcome());
}
```

Vamos executar o método acima na forma de Thread. Uma Thread permite que uma função seja executada sem encerrar a anterior, isto permite que os eventos da janela "bem-vindo" sejam executados sem encerrar a rotina atual. Modifique o construtor **FormMDI** conforme abaixo:

Etec Dr Demétrio Azevedo Junior – Itapeva-SP

```
public FormMDI()  
{  
    // a linha abaixo cria um objeto que contém o metodo  
    // a ser executado paralelamente  
    ThreadStart start = new ThreadStart(startBemVindo);  
    // passa start para a variavel t que contém a linha  
    // de execução  
    Thread t = new Thread(start);  
    // executa a linha paralelamente ao código atual  
    t.Start();  
    // aqui o truque. Aguarda 5 segundos antes de executar  
    // o FormMDI, assim a tela de boas vindas parece aparecer  
    // antes fdo FormMDI  
    Thread.Sleep(5000);  
    InitializeComponent();  
    // após mostrar boas vindas por 5 segundos encerra a janela  
    // impedindo que ela seja aberta novamente  
    t.Abort();  
}
```

Veja o resultado visual:

