

Login

Criar um novo website

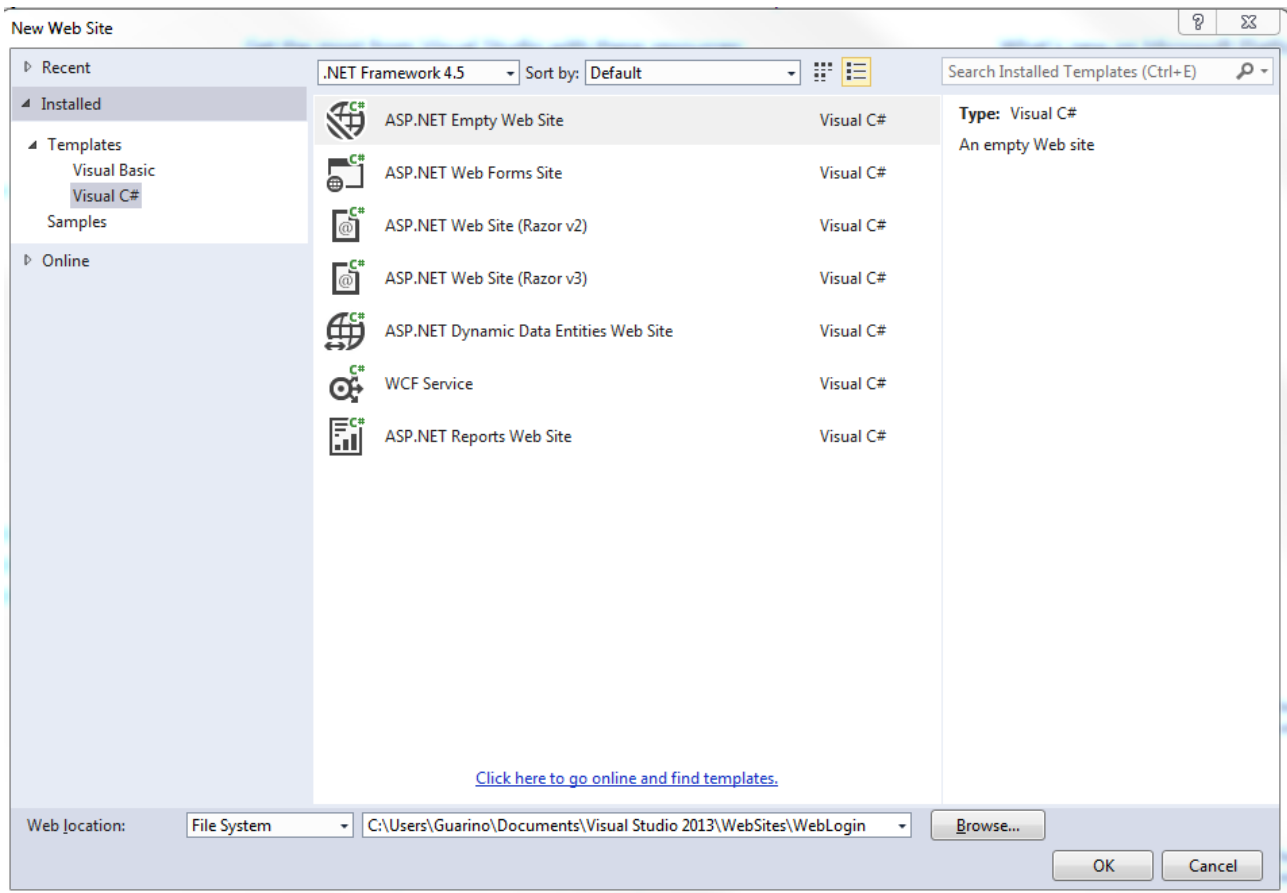
File > New > WebSite

Framework 4.5

Visual C#

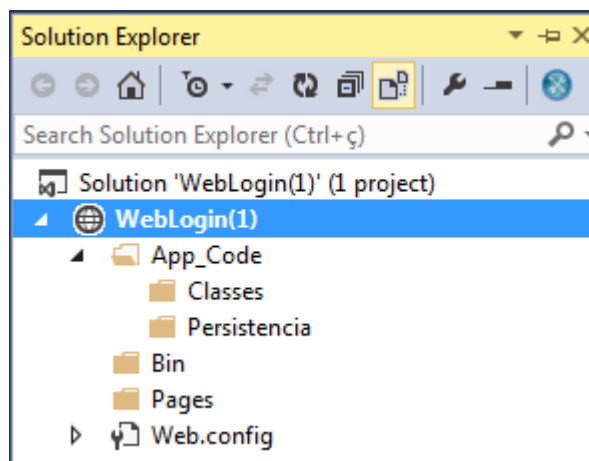
ASP.NET Empty Web Site

Nome do projeto: WebLogin



Estrutura do projeto

Criar pastas e deixar a estrutura do projeto conforme a figura.



DLL

Na pasta Bin > Add > Existing Item...

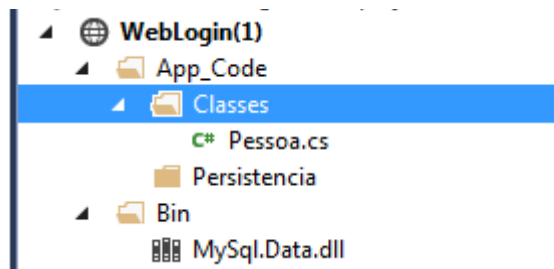
Adicionar o arquivo MySql.Data.dll, disponível em

<http://www.luizguarino.com.br/site/pi/MySql.Data.dll>

Nova Classe

Adicione uma nova classe na pasta App_Code/Classes

Nome da classe: Pessoa



Defina o namespace dessa classe como

```
namespace WebLogin.Classes
```

A classe Pessoa deve ficar como:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebLogin.Classes
{
    /// <summary>
    /// Summary description for Pessoa
    /// </summary>
    public class Pessoa
    {

        public int Codigo { get; set; }
        public string Nome { get; set; }
        public string Email { get; set; }
        public string Senha { get; set; }
        public int Tipo { get; set; }

        public Pessoa()
        {
            //
            // TODO: Add constructor logic here
            //
        }
    }
}
```

Novo Schema

No MySQL Workbench, criar um novo schema chamado bdlogin

Tabela

Criar a tabela no banco de dados

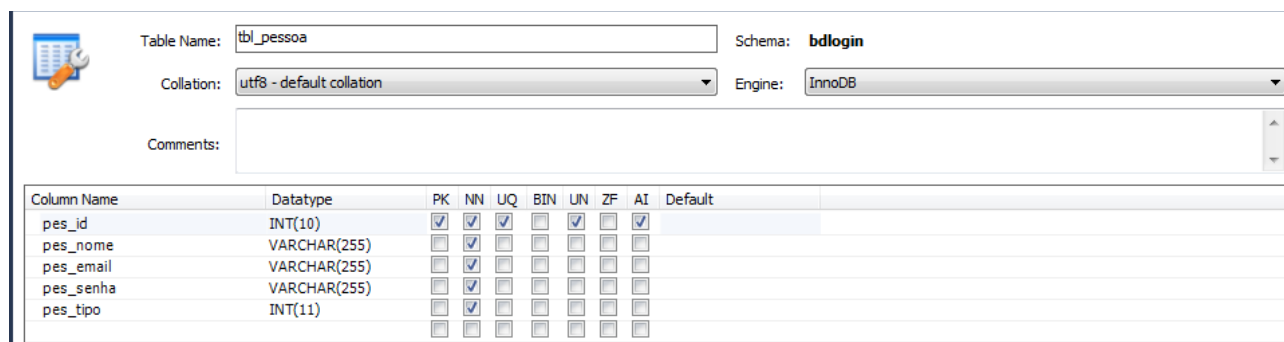


Table Name: Schema: **bdlogin**

Collation: Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
pes_id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
pes_nome	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
pes_email	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
pes_senha	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
pes_tipo	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Inserir Registros

Adicione alguns registros nessa tabela, diretamente no banco de dados. Estou assumindo, na coluna pes_tipo, que 0 é Administrador e 1 é Cliente.

	pes_id	pes_nome	pes_email	pes_senha	pes_tipo
	1	Guarino	du@gmail.com	123	0
	2	Luiz	luiz@gmail.com	1234	1
	3	Eduardo	edu@gmail.com	123	0
▶*	NULL	NULL	NULL	NULL	NULL

Adicionando o Mapped

Na pasta App_Code > Add > Existing Item...

Adicionar o arquivo Mapped.cs, disponível em

<http://www.luizguarino.com.br/site/pi/Mapped.zip>

Descompacte o arquivo antes de adicionar a classe Mapped.

Depois de adicioná-la ao projeto, altere o namespace da classe para

`namespace WebLogin`

String de conexão

No arquivo web.config, adicione a string de conexão

```
<configuration>
  <appSettings>
    <add key="strConexao" value="Database=bdlogin;Data Source=localhost;User
    Id=root;Password=senha_do_banco; pooling=false"/>
  </appSettings>
  ...
</configuration>
```

Nova Classe

No projeto, adicione uma nova classe na pasta App_Code/Persistencia, chamada PessoaBD. Defina o namespace como

```
namespace WebLogin.Persistencia
```

Adicione um método para realizar a autenticação, passando como parâmetro o e-mail e a senha. Adicione um método para recuperar as informações da Pessoa, passando como parâmetro o id da pessoa.

A classe deverá ficar como:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using WebLogin.Classes;

namespace WebLogin.Persistencia
{
    /// <summary>
    /// Summary description for PessoaBD
    /// </summary>
    public class PessoaBD
    {
        public Pessoa Autentica(string email, string senha)
        {
            Pessoa obj = null;

            System.Data.IDbConnection objConexao;
            System.Data.IDbCommand objCommand;
            System.Data.IDataReader objDataReader;

            objConexao = Mapped.Connection();
            objCommand = Mapped.Command("SELECT * FROM tbl_pessoa WHERE pes_email = ?email
and pes_senha = ?senha", objConexao);

            objCommand.Parameters.Add(Mapped.Parameter("?email", email));
            objCommand.Parameters.Add(Mapped.Parameter("?senha", senha));

            objDataReader = objCommand.ExecuteReader();
```

```

while (objDataReader.Read())
{
    obj = new Pessoa();
    obj.Codigo = Convert.ToInt32(objDataReader["pes_id"]);
    obj.Nome = Convert.ToString(objDataReader["pes_nome"]);
    obj.Email = Convert.ToString(objDataReader["pes_email"]);
    obj.Tipo= Convert.ToInt32(objDataReader["pes_tipo"]);
}

objDataReader.Close();
objConexao.Close();

objCommand.Dispose();
objConexao.Dispose();
objDataReader.Dispose();

return obj;
}

public Pessoa Select(int id)
{
    Pessoa obj = null;

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataReader objDataReader;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_pessoa WHERE pes_id = ?codigo",
objConexao);
    objCommand.Parameters.Add(Mapped.Parameter("?codigo", id));

    objDataReader = objCommand.ExecuteReader();
    while (objDataReader.Read())
    {
        obj = new Pessoa();
        obj.Codigo = Convert.ToInt32(objDataReader["pes_id"]);
        obj.Nome = Convert.ToString(objDataReader["pes_nome"]);
        obj.Email = Convert.ToString(objDataReader["pes_email"]);
        obj.Tipo = Convert.ToInt32(objDataReader["pes_tipo"]);
    }

    objDataReader.Close();
    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();
    objDataReader.Dispose();

    return obj;
}

public PessoaBD()
{
    //
    // TODO: Add constructor logic here
    //
}
}
}

```

Adicionando pastas para cada tipo de pessoa

Adicione as pastas Administrador, Cliente e Erro dentro de Pages.

Dentro da pasta Pages/Administrador, adicione um Webform chamado Index.aspx.

Dentro da pasta Pages/Cliente, adicione um Webform chamado Index.aspx.

Dentro da pasta Pages/Erro, adicione um Webform chamado AcessoNegado.aspx.

Dentro da pasta Pages, adicione um Webform chamado Login.aspx

Pages/Administrador/Index.aspx

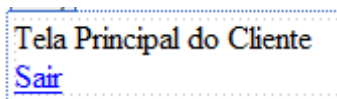
Colocar apenas um Label e um LinkButton



Componente	ID	Text
Label	lblTitulo	Tela Principal do Administrador
LinkButton	lbSair	Sair

Pages/Cliente/Index.aspx

Colocar apenas um Label



Componente	ID	Text
Label	lblTitulo	Tela Principal do Clinte
LinkButton	lbSair	Sair

Login.aspx

Adicione os componentes:

Componente	ID	Text
Label	lblTitulo	Login
Label	lblEmail	Email
Textbox	txtEmail	
Label	lblSenha	Senha
Textbox	txtSenha	
Button	btnEntrar	Entrar
Label	lblMensagem	

Login

Email

Senha

Entrar

[!Mensagem]

Pages/Erro/AcessoNegado.aspx

Colocar um Label e um Hyperlink.

Label	Text: Acesso Negado
Hyperlink	NavigateUrl: ~/Pages/Login.aspx Text: Voltar para a tela de Login

Acesso Negado
[Voltar para a tela de Login](#)

Tela Login.

Adicionar os namespaces

```
using WebLogin.Classes;
using WebLogin.Persistencia;
```

Criar os métodos a seguir no Login.aspx.cs

```
private bool IsPreenchido(string str)
{
    bool retorno = false;
    if (str != string.Empty)
    {
        retorno = true;
    }
    return retorno;
}

private bool UsuarioEncontrado(Pessoa pessoa)
{
    bool retorno = false;
    if (pessoa != null)
    {
        retorno = true;
    }
    return retorno;
}
```

No click do botão:

```
string email = txtEmail.Text.Trim();
string senha = txtSenha.Text.Trim();

if (!IsPreenchido(email))
{
    lblMensagem.Text = "Preencha o email";
    txtEmail.Focus();
    return;
}
if (!IsPreenchido(senha))
{
    lblMensagem.Text = "Preencha a senha";
    txtSenha.Focus();
    return;
}

PessoaBD bd = new PessoaBD();
Pessoa pessoa = new Pessoa();

pessoa = bd.Autentica(email, senha);

if (!UsuarioEncontrado(pessoa))
{
    lblMensagem.Text = "Usuário não encontrado";
    txtEmail.Focus();
    return;
}

Session["ID"] = pessoa.Codigo;

switch (pessoa.Tipo)
{
    case 0:
        Response.Redirect("Administrador/Index.aspx");
        break;
    case 1:
        Response.Redirect("Cliente/Index.aspx");
        break;
    default:
        break;
}
```


Administrador/Index.aspx

Adicionar os namespaces

```
using WebLogin.Classes;
using WebLogin.Persistencia;
```

Adicionar o método que verifica se é administrador (tipo ==0)

```
private bool IsAdministrador(int tipo)
{
    bool retorno = false;
    if (tipo == 0)
    {
        retorno = true;
    }
    return retorno;
}
```

No Page_Load dessa página

```
protected void Page_Load(object sender, EventArgs e)
{
    int codigo = Convert.ToInt32( Session["ID"]);
    PessoaBD bd = new PessoaBD();
    Pessoa pessoa = bd.Select(codigo);
    if (!IsAdministrador(pessoa.Tipo))
    {
        Response.Redirect("../Erro/AcessoNegado.aspx");
    }
    else
    {
        lblTitulo.Text = "Bem vindo (Administrador) : " + pessoa.Nome;
    }
}
```

2 cliques no LinkButton para alterar o evento click do componente

```
protected void lbSair_Click(object sender, EventArgs e)
{
    Session.Abandon();
    Session.Clear();
    Session.RemoveAll();
    Response.Redirect("../Login.aspx");
}
```

Cliente/Index.aspx

Adicionar os namespaces

```
using WebLogin.Classes;  
using WebLogin.Persistencia;
```

Adicionar o método que verifica se é cliente (tipo ==1)

```
private bool IsCliente(int tipo)  
{  
    bool retorno = false;  
    if (tipo == 1)  
    {  
        retorno = true;  
    }  
    return retorno;  
}
```

No Page_Load dessa página

```
protected void Page_Load(object sender, EventArgs e)  
{  
    int codigo = Convert.ToInt32(Session["ID"]);  
    PessoaBD bd = new PessoaBD();  
    Pessoa pessoa = bd.Select(codigo);  
    if (!IsCliente(pessoa.Tipo))  
    {  
        Response.Redirect("../Erro/AcessoNegado.aspx");  
    }  
    else  
    {  
        lblTitulo.Text = "Bem vindo (Cliente) : " + pessoa.Nome;  
    }  
}
```

2 cliques no LinkButton para alterar o evento click do componente

```
protected void lbSair_Click(object sender, EventArgs e)  
{  
    Session.Abandon();  
    Session.Clear();  
    Session.RemoveAll();  
    Response.Redirect("../Login.aspx");  
}
```

Colocar a página Login como a primeira página a ser executada (Set As Start Page).

Executar a aplicação

Logar como Administrador e Sair.

Logar como Cliente e Sair

Caso o Administrador tente acessar a página do Cliente (ou vice-versa), a tela Acesso Negado será exibida.