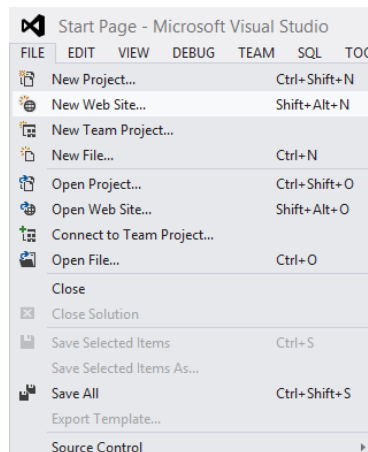


Tutorial de Aplicação Simples usando Web Forms e MySQL

Luiz Eduardo Guarino de Vasconcelos

Abra o Visual Studio.NET

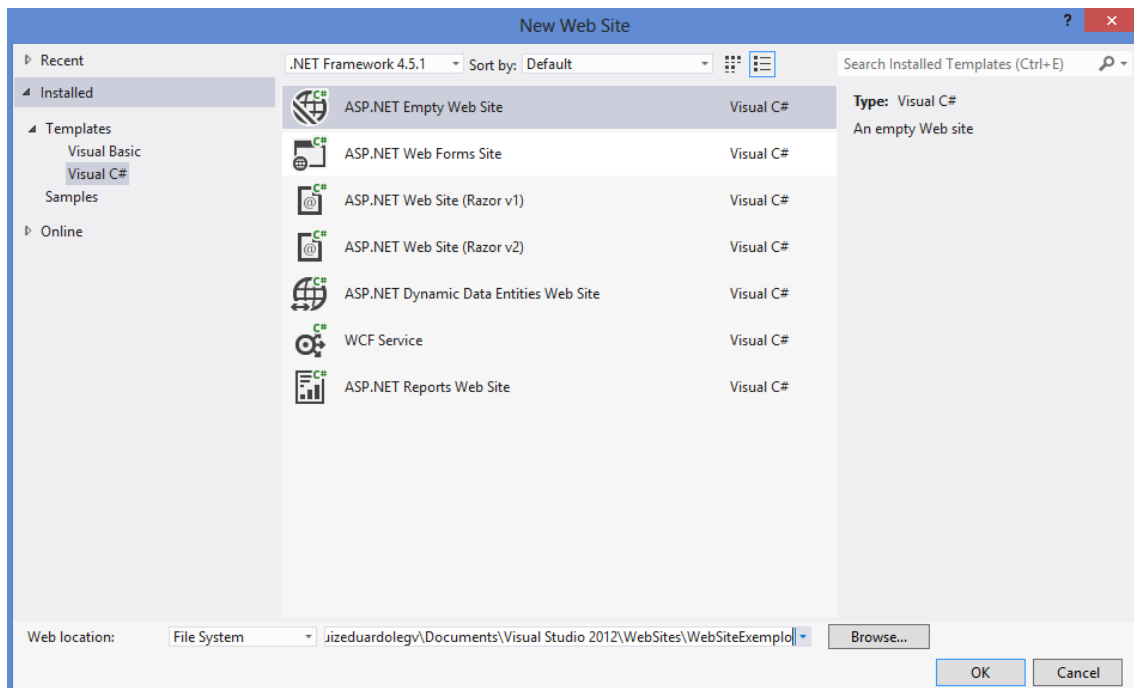
Clique em File > New Web Site



Selecione

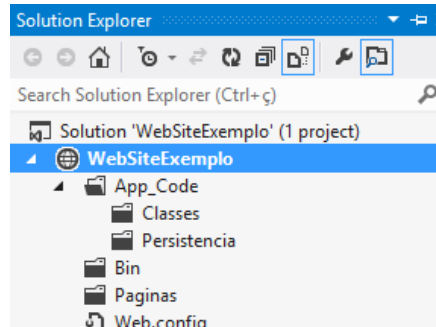
- Visual C#
- .NET Framework 4.5 ou 4.5.1
- ASP.NET Empty Web Site
- Web Location: WebSiteExemplo

Clique em OK



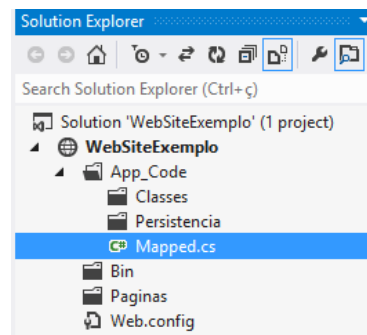
Adicione as pastas a seguir na Solution Explorer:

- App_Code
 - Classes (ficarão as classes de definição – Modelagem)
 - Persistencia (ficarão as classes que terão os métodos que acessarão o banco de dados)
- Bin (ficarão as dlls da aplicação)
- Paginas (ficarão as páginas da aplicação – View)



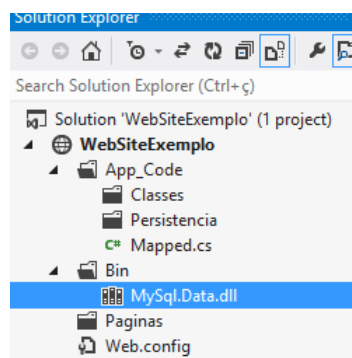
Adicionando a classe de Mapeamento

- Faça o download do arquivo http://www.luizguarino.com.br/site/aspnet_webforms/Mapped.zip
- Descompacte-o
- Adicione-o na pasta App_Code
 - Botão direito na pasta App_Code > Adicionar Item Existente



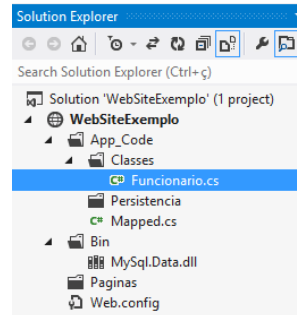
Adicionando DLL para acesso ao MySQL

- Baixe o arquivo http://www.luizguarino.com.br/site/aspnet_webforms/MySQL.Data.dll
- Adicione-o na pasta Bin



Criando uma Classe de definição

- Botão direito na pasta Classes
- Adicionar Nova Classe
- Nome da classe: Funcionario.cs



Código da classe Funcionario.cs

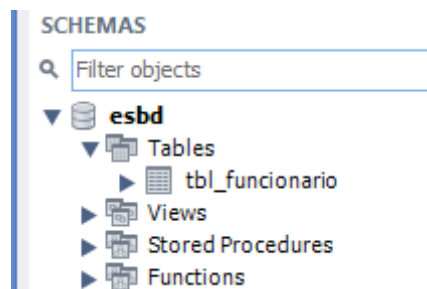
```
using System;
using System.Web;

namespace WebSiteExemplo.Classes
{
    /// <summary>
    /// Summary description for Funcionario
    /// </summary>
    public class Funcionario
    {
        //propriedades
        public int Codigo { get; set; }
        public string Nome { get; set; }
        public double Salario { get; set; }
        public string Cracha { get; set; }

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

Criando o banco de dados

- No MySQL Query Browser ou MySQL Workbench, crie o schema (database) esbd
- Crie a tabela tbl_funcionario



A tabela deve ter os seguintes campos

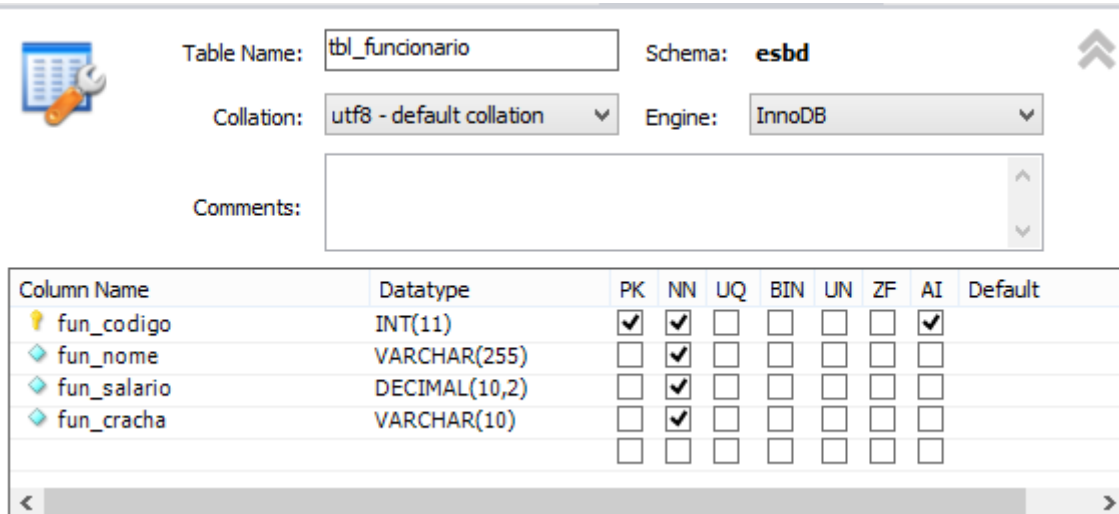






Table Name: Schema: **esbd**

Collation: Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
 fun_codigo	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
 fun_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"/>	
 fun_salario	DECIMAL(10,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 fun_cracha	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

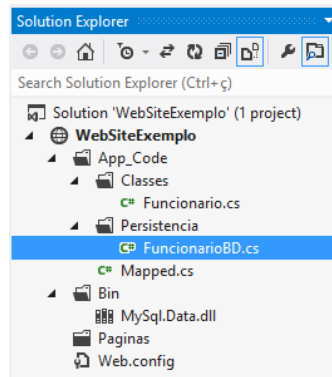
Adicionando a string de conexão na aplicação

- Abra o arquivo web.config na Solution Explorer
- Adicione a tag appSettings dentro da tag Configuration, conforme mostrado a seguir
 - Em Database, informe o nome do seu schema (ex: esbd)
 - Em Data Source, informe o local ou IP do servidor (ex: localhost)
 - Em User Id, informe o usuário que acessará o banco de dados (ex: root)
 - Em Password, informe a senha de acesso ao banco de dados. Caso seu banco de dados não tenha senha, deixe Password=;

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

Criando a classe que terá os métodos de acesso ao banco de dados relacionados a classe Funcionario

- Botão direito na pasta Persistencia
- Adicione Nova Classe
- Nome da classe: FuncionarioBD.cs



Alterando o código da classe FuncionarioBD.cs

```
using FATEC; //para acesso a classe Mapped
using System;
using System.Web;
using WebSiteExemplo.Classes; //para acesso a classe Funcionario
using System.Data; //para uso de DataSet
```

```
namespace WebSiteExemplo.Persistencia
{
    /// <summary>
    /// Summary description for FuncionarioBD
    /// </summary>
    public class FuncionarioBD
    {
        //métodos
        //insert

        //selectall

        //select

        //update

        //delete

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

Adicionando o método de cadastro. Insira o código abaixo do comentário //insert

```
//métodos
//insert
public bool Insert(Funcionario funcionario)
{
    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    string sql = "INSERT INTO tbl_funcionario(fun_nome, fun_salario, fun_cracha) VALUES (?nome, ?salario, ?cracha)";

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command(sql, objConexao);

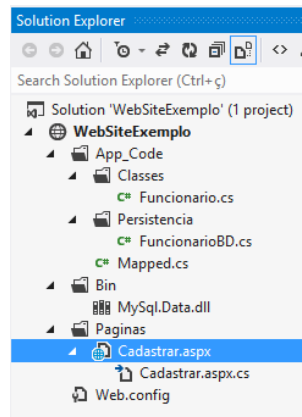
    objCommand.Parameters.Add(Mapped.Parameter("?nome", funcionario.Nome));
    objCommand.Parameters.Add(Mapped.Parameter("?salario", funcionario.Salario));
    objCommand.Parameters.Add(Mapped.Parameter("?cracha", funcionario.Cracha));

    objCommand.ExecuteNonQuery();
    objConexao.Close();
    objCommand.Dispose();
    objConexao.Dispose();

    return true;
}
```

Adicionando uma Página para Cadastro de Funcionario

- Na pasta Paginas
- Botão direito, Adicionar Novo Web Form
- Nome do web form: Cadastrar.aspx



Cria a seguinte tela de cadastro. Os componentes usados são Label, Textbox e Button.

Cadastro de funcionário

Nome: (ID: txtNome)

Salário: (ID: txtSalario)

Crachá: (ID: txtCracha)

(ID: btnSalvar, TEXT: Salvar)

[lblMensagem] (ID: lblMensagem, TEXT=vazio)

Cadastrando um novo funcionário.

- Adicione os namespaces a seguir no topo do código da página
 - using WebSiteExemplo.Classes;
 - using WebSiteExemplo.Persistencia;

- Dê dois cliques no botão btnSalvar e adicione o seguinte código

```

Funcionario funcionario = new Funcionario();
funcionario.Nome = txtNome.Text;
funcionario.Salario = Convert.ToDouble(txtSalario.Text);
funcionario.Cracha = txtCracha.Text;

FuncionarioBD bd = new FuncionarioBD();
if (bd.Insert(funcionario))
{
    lblMensagem.Text = "Funcionário cadastrado com sucesso";

    txtNome.Text="";
    txtCracha.Text="";
    txtSalario.Text="";
    txtNome.Focus();
}
else
{
    lblMensagem.Text = "Erro ao salvar.";
}

```

Deixe a página Cadastrar.aspx como a primeira a ser executada (botão direito na página Cadastrar.aspx, Set As Start Page)

Execute o projeto.

Preencha os campos corretamente.

Clique em Salvar.

Listando os Funcionários

- Na classe FuncionarioBD.cs
- Adicione o código a seguir abaixo do comentário //selectall

```

//selectall
public DataSet SelectAll()
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_funcionario", objConexao);
    objDataAdapter = Mapped.Adapter(objCommand);
    objDataAdapter.Fill(ds);

    objConexao.Close();

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

    return ds;
}

```

Adicionando uma página para Listar os Funcionários

- Na pasta Paginas
- Adicionar Novo Web Form
- Nome: Listar.aspx
- Definir como Set As Start Page
- Arrastar um GridView para essa página

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

No código da página, importe os namespaces

```
using WebsiteExemplo.Persistencia;  
using System.Data;
```

No Page_Load da página, adicione o seguinte código

```
private void Carrega()  
{  
    FuncionarioBD bd = new FuncionarioBD();  
    DataSet ds = bd.SelectAll();  
    GridView1.DataSource = ds.Tables[0].DefaultView;  
    GridView1.DataBind();  
}  
protected void Page_Load(object sender, EventArgs e)  
{  
    Carrega();  
}
```

Execute o projeto. A Lista deverá ser exibida.

Caso queira, adicione um link da Lista para a página de Cadastro.

Alterando a Lista para permitir Alterar e Excluir funcionários.

- Na página Lista.aspx, em modo Source, deixe o GridView a seguinte forma

```
<asp:GridView ID="GridView1" runat="server" OnRowCommand="GridView1_RowCommand">  
    <Columns>  
        <asp:TemplateField>  
            <ItemTemplate>  
                <asp:LinkButton ID="lbAlterar" runat="server" CommandName="Alterar"  
                    CommandArgument="<%# Bind("fun_codigo")%>">Alterar</asp:LinkButton>  
            </ItemTemplate>  
        </asp:TemplateField>  
        <asp:TemplateField>  
            <ItemTemplate>  
                <asp:LinkButton ID="lbDeletar" runat="server" CommandName="Deletar"  
                    CommandArgument="<%# Bind("fun_codigo")%>">Excluir</asp:LinkButton>  
            </ItemTemplate>  
        </asp:TemplateField>  
    </Columns>
```



```
</asp:GridView>
```

Na classe FuncionarioBD.cs, crie os métodos de Select, Update e Delete

```
//select
public Funcionario Select(int id)
{
    Funcionario obj = null;

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

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

    objDataReader = objCommand.ExecuteReader();
    while (objDataReader.Read())
    {
        obj = new Funcionario ();
        obj.Codigo = Convert.ToInt32(objDataReader["fun_codigo"]);
        obj.Nome = Convert.ToString(objDataReader["fun_nome"]);
        obj.Salario = Convert.ToDouble(objDataReader["fun_salario"]);
        obj.Cracha = Convert.ToString (objDataReader["fun_cracha"]);
    }

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

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

    return obj;
}

//update
public bool Update(Funcionario funcionario)
{
    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    string sql = "UPDATE tbl_funcionario SET fun_nome=?nome, fun_salario=?salario, fun_cracha=?cracha WHERE
fun_codigo=?codigo";

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command(sql, objConexao);

    objCommand.Parameters.Add(Mapped.Parameter("?nome", funcionario.Nome));
    objCommand.Parameters.Add(Mapped.Parameter("?salario", funcionario.Salario));
    objCommand.Parameters.Add(Mapped.Parameter("?cracha", funcionario.Cracha));
    objCommand.Parameters.Add(Mapped.Parameter("?codigo", funcionario.Codigo));

    objCommand.ExecuteNonQuery();
    objConexao.Close();
    objCommand.Dispose();
    objConexao.Dispose();

    return true;
}

//delete
public bool Delete(int id)
{
    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    string sql = "DELETE FROM tbl_funcionario WHERE fun_codigo=?codigo";

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command(sql, objConexao);

    objCommand.Parameters.Add(Mapped.Parameter("?codigo", id));

    objCommand.ExecuteNonQuery();
    objConexao.Close();
    objCommand.Dispose();
    objConexao.Dispose();

    return true;
}
```

Crie a página para Alteração de Funcionário na pasta Paginas. O nome da página (web form) deve ser Alterar.aspx e deve ter o seguinte layout.

Alteração de funcionário

Nome: (ID: txtNome)

Salário: (ID: txtSalario)

Crachá: (ID: txtCracha)

Salvar (ID: btnSalvar, TEXT: Salvar)

[lblMensagem] (ID: lblMensagem, TEXT=vazio)

No código da página Listar.aspx (ou seja, no arquivo Listar.aspx.cs), adicione o seguinte código após o fim do código do Page_Load

```
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
    int codigo = 0;
    switch (e.CommandName)
    {
        case "Alterar":
            codigo = Convert.ToInt32(e.CommandArgument);
            Session["ID"] = codigo;
            Response.Redirect("Alterar.aspx");
            break;
        case "Deletar":
            codigo = Convert.ToInt32(e.CommandArgument);
            FuncionarioBD bd = new FuncionarioBD();
            bd.Delete(codigo);
            Carrega();
            break;
        default:
            break;
    }
}
```

No Page_Load da página Alterar.aspx adicione o seguinte código.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        FuncionarioBD bd = new FuncionarioBD();
        Funcionario funcionario = bd.Select(Convert.ToInt32(Session["ID"]));
        txtNome.Text = funcionario.Nome;
        txtSalario.Text = funcionario.Salario.ToString();
        txtCracha.Text = funcionario.Cracha;
    }
}
```

No botão Salvar da página Alterar.aspx, coloque o código a seguir:

```
protected void btnSalvar_Click(object sender, EventArgs e)
{
    FuncionarioBD bd = new FuncionarioBD();
    Funcionario funcionario = bd.Select(Convert.ToInt32(Session["ID"]));
    funcionario.Nome = txtNome.Text;
    funcionario.Salario = Convert.ToDouble(txtSalario.Text);
    funcionario.Cracha = txtCracha.Text;

    if (bd.Update(funcionario))
    {
        lblMensagem.Text = "Funcionário alterado com sucesso";
        txtNome.Focus();
    }
    else
    {
        lblMensagem.Text = "Erro ao salvar.";
    }
}
```

Deixe a página Listar.aspx como Set As Start Page.

Execute o projeto.

O CRUD (Cadastro, Alteração, Exclusão e Lista) deve funcionar.

Qualquer dúvida, envie-me um e-mail (du.guarino@gmail.com)

Abraço

Guarino