

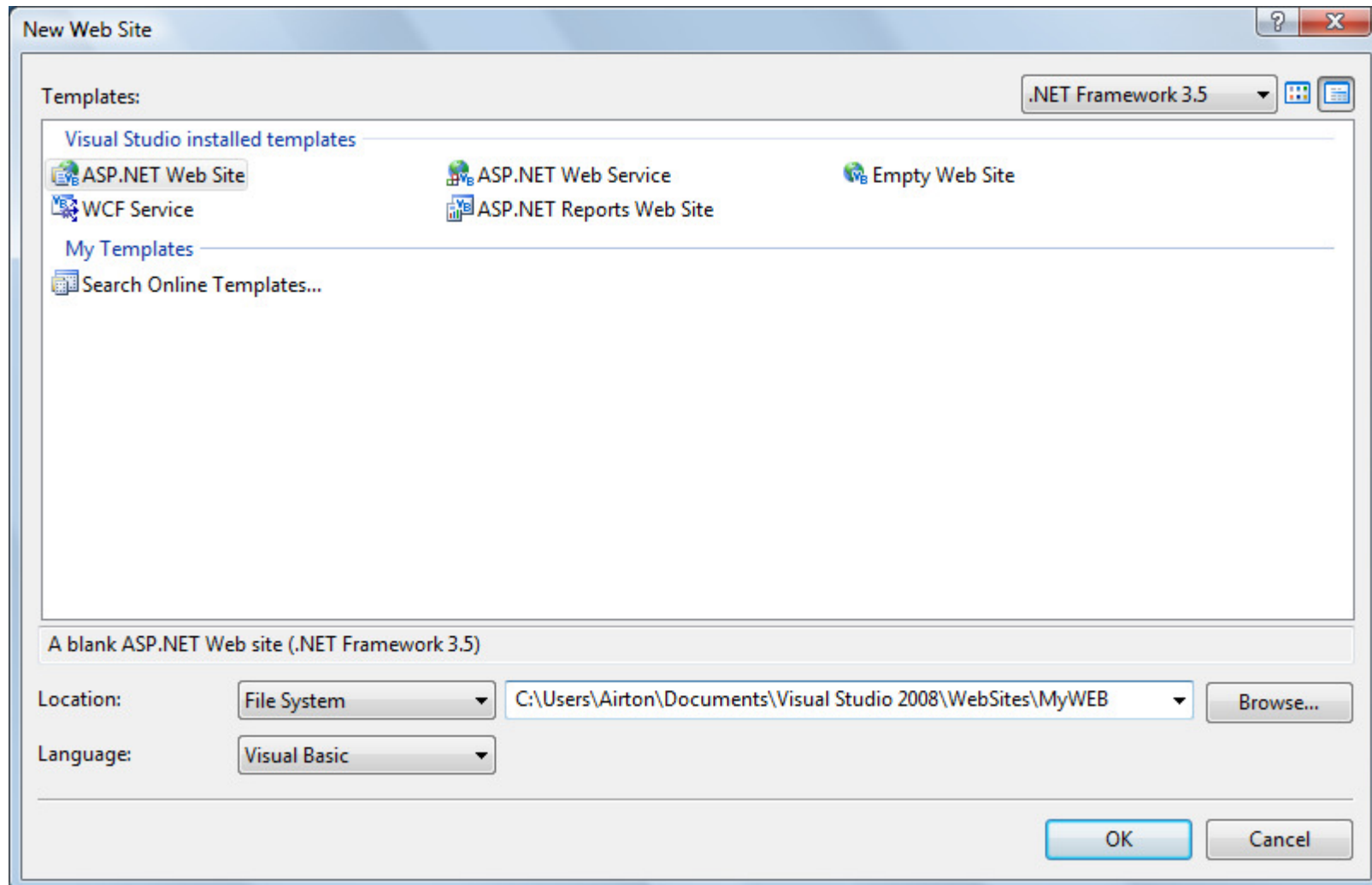
Segue abaixo o código VB.NET necessário para conectar ao BD PostgreSQL #
#####

Rafael de Araújo Andrade

Data: 12-01-2011

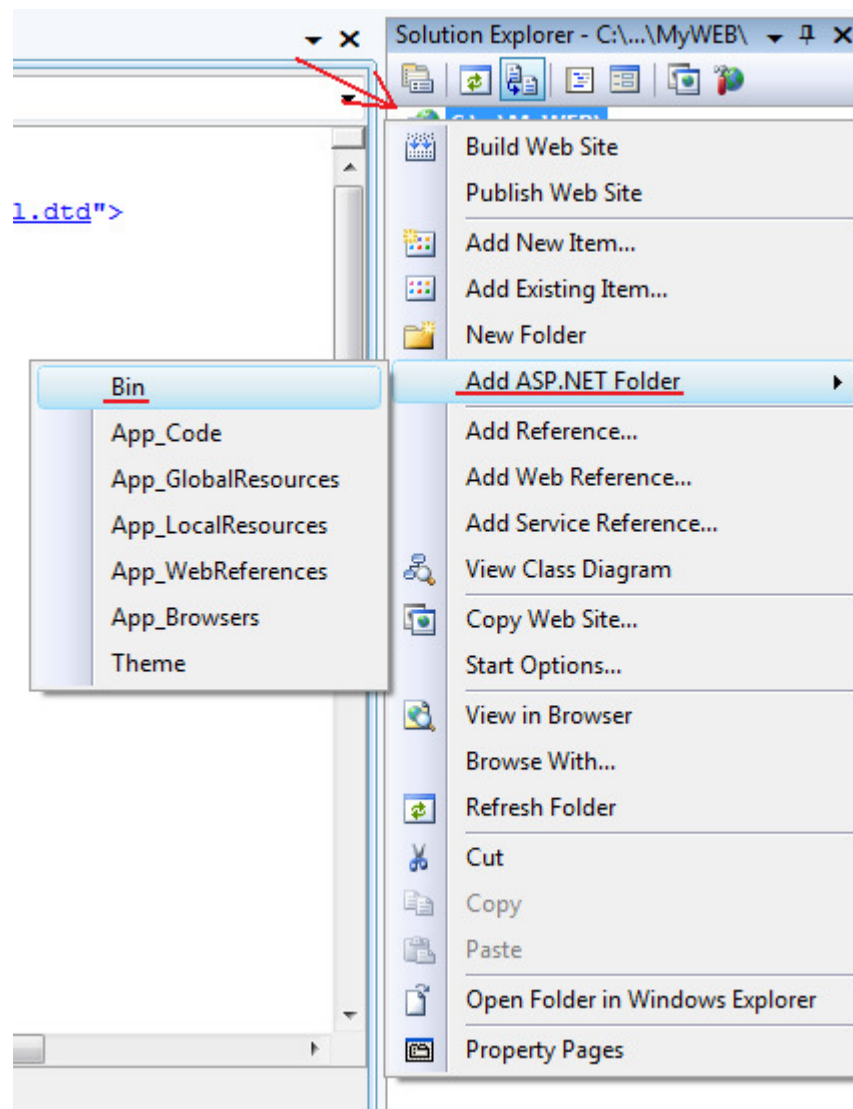
Site: <http://www.truquesemacetes.com>

Menu "File" > "New WebSite"

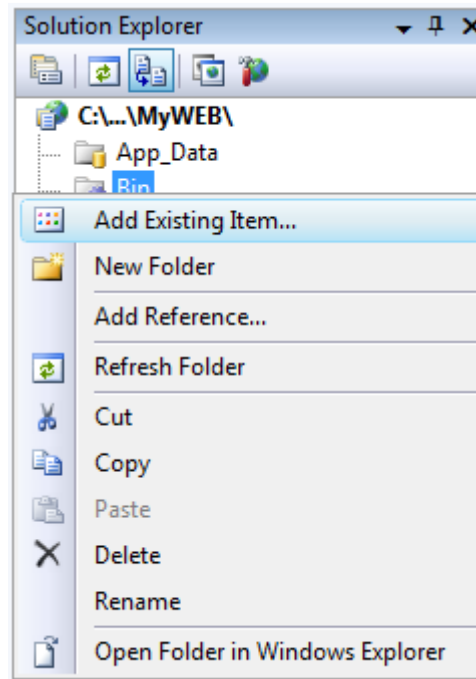


Estique a imagem para ver com detalhes

No projeto em, clique com botão direito sobre ele e em seguida em “Add ASP.NET Folder” > “Bin”



Estique a imagem para ver com detalhes



Estique a imagem para ver com detalhes

Clique com botão direito sobre a pasta “Bin” e clique em “Add Existing Item” e adicione os arquivos:

- Mono.Security.dll
- Npgsql.dll
- policy.2.0.Npgsql.dll

'Importa as API's que serão usadas para comunicação com o BD

```
Imports System.Data
```

```
Imports Npgsql
```

'Declare estas variáveis, recomendável colocar logo após o início da classe

```
'Partial Class [NOME_DA_CLASSE]
```

```
Dim Conexao As String = ConfigurationManager.AppSettings("StrConection")
```

'Pega a String de Conexão do Web.Config (Página 4)

```
'ConfigurationManager.AppSettings("StrConection")
```

```
Dim conn As NpgsqlConnection = New NpgsqlConnection(Conexao)
```

```
Dim SQL As String
```

```
Dim Comando As NpgsqlCommand
```

```
Dim da As Npgsql.NpgsqlDataAdapter
```

```
Shared ds As DataSet
```

Código de Alteração do BD (INSERT, DELETE, UPDATE, PROCEDURES...)

```
'Função que faz a comunicação e Operação e retorna 1 se conseguiu e 0 se falhou
Private Function AlteraDB() As Integer
```

```
Try
```

```
    'Abre Conexão
    conn.Open()
```

```
    'Cria SQL que será executada no BD
    'Aqui você pode usar INSERT, UPDATE, DELETE, EXECUÇÃO DE PROCEDURES...
    'Não usar SELECT, pois para obter os dados eles serão armazenados num DATASET na memória, veja a próxima página
    SQL = "INSERT INTO codbar(codbarra2) VALUES(" & txtCodigo.Text & ");"
```

```
    'Associa o SQL com a conexão do Banco de Dados
    Comando = New NpgsqlCommand(SQL, conn)
```

```
    'Efetua a Operação
    Return Comando.ExecuteNonQuery()
```

```
    'Tira a variável da memória
    Comando.Dispose()
```

```
    'Fecha Conexão
    conn.Close()
```

```
Return 1
```

```
Catch
```

```
    MsgBox("Falha na Operação !!!" & Chr(13) & Chr(13) & "Possíveis Causas:" & Chr(13) & _
        "- BD Offline" & Chr(13) & "- String de Conexão errada" & Chr(13) & "- SQL incorreta", MsgBoxStyle.Critical)
```

```
Return 0
```

```
End Try
```

```
End Function
```

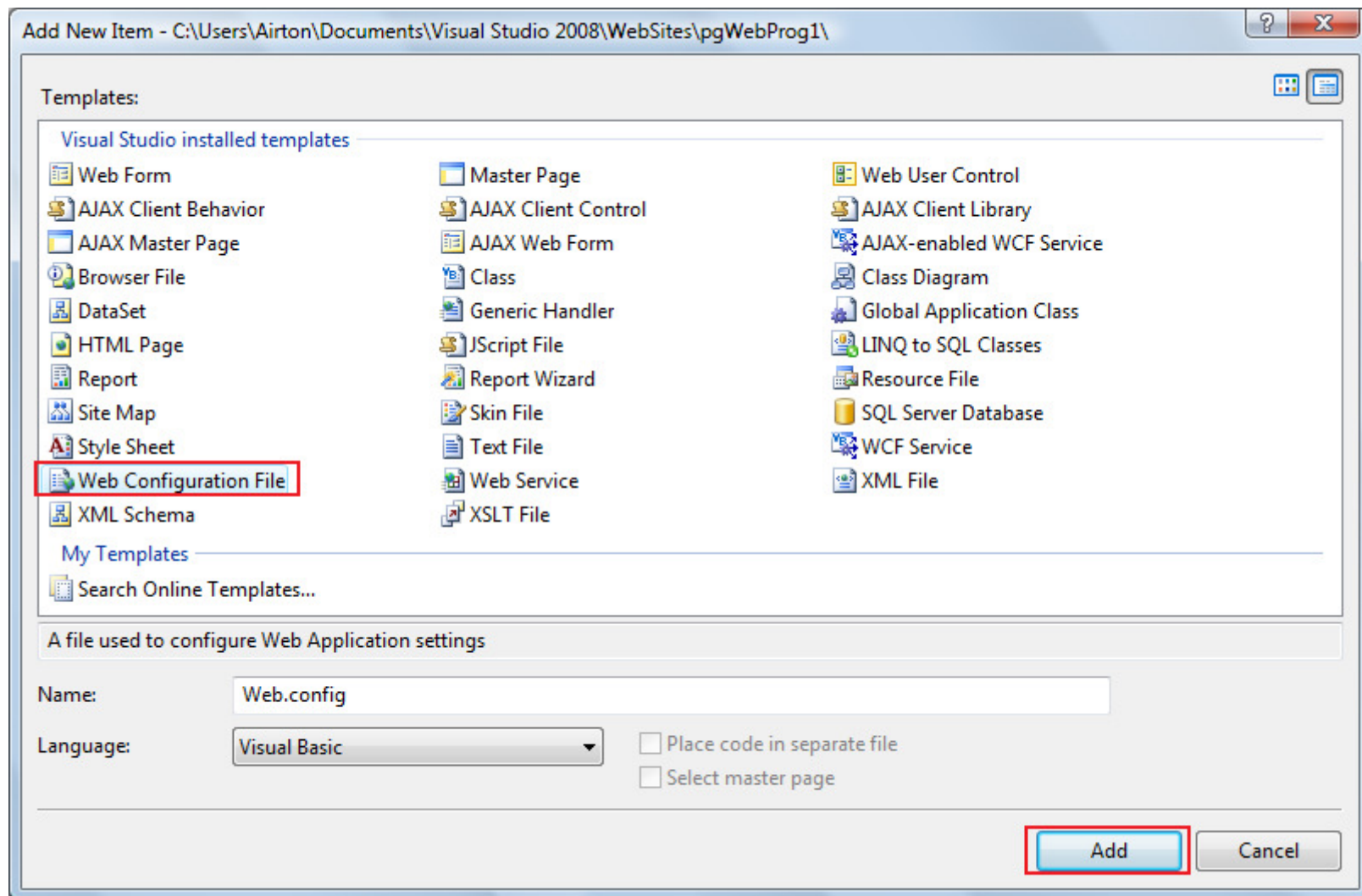
Código de Carregamento de uma Select

'Função para carregar uma select do BD e armazenar num DataSet (na memória) e carrega em qualquer objeto, seja um Gridview (que é este caso) como também em TextBox, DropDownList, Label...

```
Private Sub CarregaBD()  
  
    Try  
        ' Abre a conexão  
        conn.Open()  
  
        'Declara e atribui a Query  
        SQL = "SELECT id,titulo,descricao,link,visivel FROM TestePag WHERE visivel = TRUE ORDER BY id;"  
        'Dim SQL As String = "Select * From usuarios order by nome limit" & Str(ddlNumero.Items.Item(1)) & ";"  
  
        'Instancia o NpgsqlDataAdapter que é responsavel pela comunicação isto significa efetuar a 'leitura dos dados de  
        um banco de dados e preencher um dataset, efetuar a manutenção de 'dados e, em seguida, escrever de volta no banco de dados  
        com as informações atualizadas do 'dataset. Na verdade, um data adapter pode mover dados entre qualquer fonte de dados e um  
        'dataset.  
        da = New NpgsqlDataAdapter(SQL, conn)  
  
        'Instacia um DataSet  
        ds = New DataSet()  
  
        conn.Close()  
  
        'Popula o DataAdapter com o que retornou do DataSet  
        ' "TestePag" é o nome desses dados armazenados através desta select, é possível armazenar vários conjuntos de dados  
        apenas mudando este nome  
        da.Fill(ds, "TestePag")  
  
        'Atribui um Datasource a Grid  
        GridView1.DataSource = ds.Tables("TestePag").DefaultView  
  
        'Aciona a leitura da fonte de dados  
        GridView1.DataBind()  
  
        'Exemplo de carregamento de um "célula" (que neste caso é a coluna Título, seguindo a ordem da SELECT) da tabela  
        nomeada "TestePag" na Linha 0, "Rows(0)", (onde no VB.NET o primeiro registro é o 0) na coluna 1, "Item(1)", (onde 1 é o 2ª  
        coluna). Ex.: Se uma tabela tem 10 linhas (registros), o primeiro é 0 e o último é 9  
        lblTitulo.Text = ds.Tables("TestePag").Rows(0).Item(1)  
  
    Catch ex As Exception  
    End Try  
  
End Sub
```

Web.Config

No projeto em, clique com botão direito sobre ele e em seguida em “Add New Item”



Substitua o código `<appSettings />` por:

```
<appSettings>
  <add key="StrConection" value="server=localhost;Port=5432;User Id=postgres;Password=root;Database=MyDB"/>
</appSettings>
```

'Onde Server é o IP (localhost = 127.0.0.1 que é o IP da placa de Rede Local)

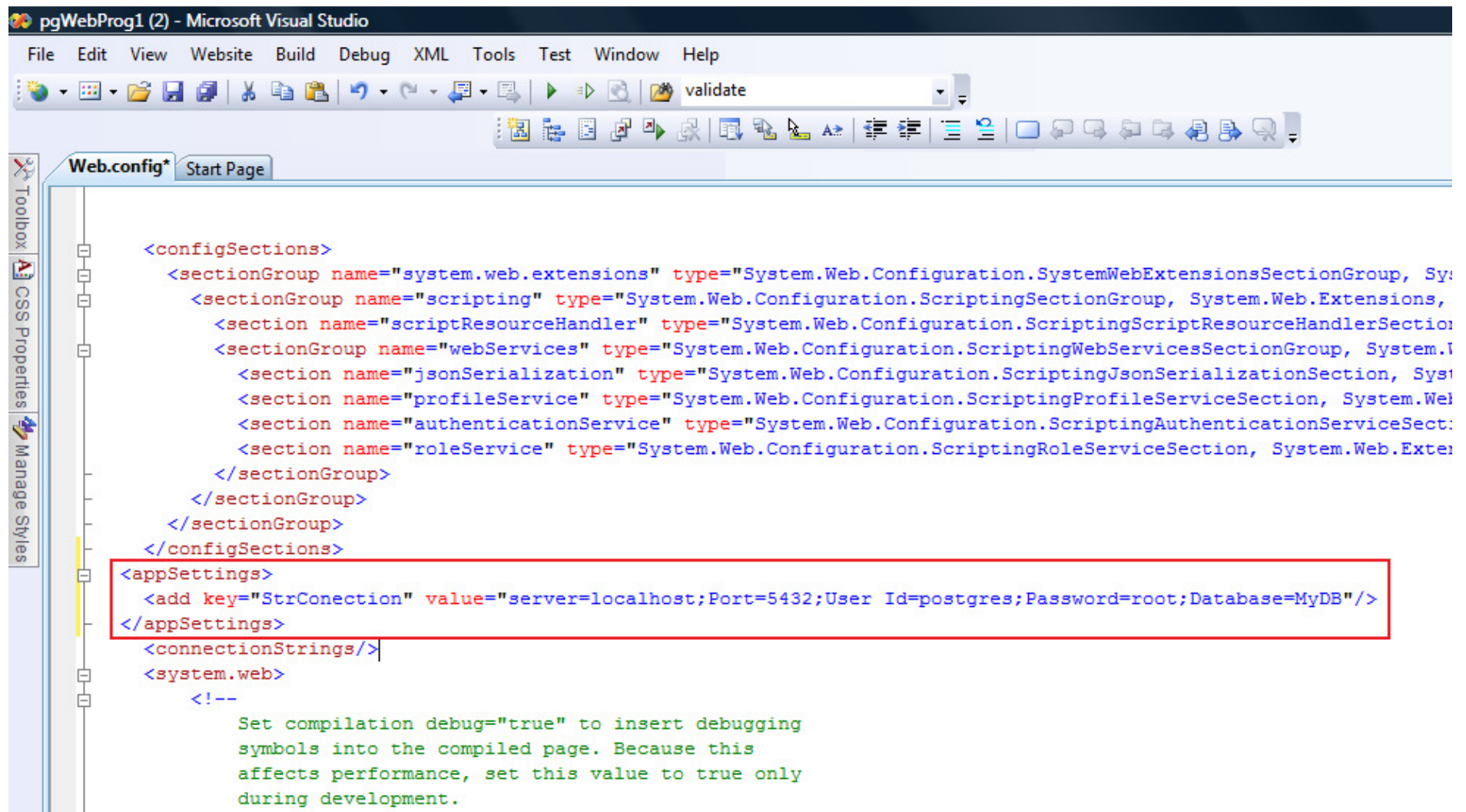
'Port é a porta onde seu postgres está configurado para se comunicar, se você fez a instalação padrão a porta é esta mesma.

'User Id é o Nome do Usuário que está conectando ao BD.

'Password => Digite a senha do usuário

'Database => Nome do Banco que você quer se comunicar

'Key é o nome identificador desta String de conexão, ela é usada na declaração `ConfigurationManager.AppSettings("StrConection")`



Pronto, agora que começa a diversão.

Este MicroTutorial abre portas pra tudo que você imagina numa aplicação.

Agora que você sabe carregar registros em Labes e HyperLinks (Label com link), tente fazer paginação onde há apenas uma página e só muda o conteúdo, é o que há em economia de tempo.

Atenciosamente

Rafael de Araújo Andrade

"Por uma internet com conteúdo rico, auto-explicativo e LIVRE"

<http://www.truquesemacetes.com>