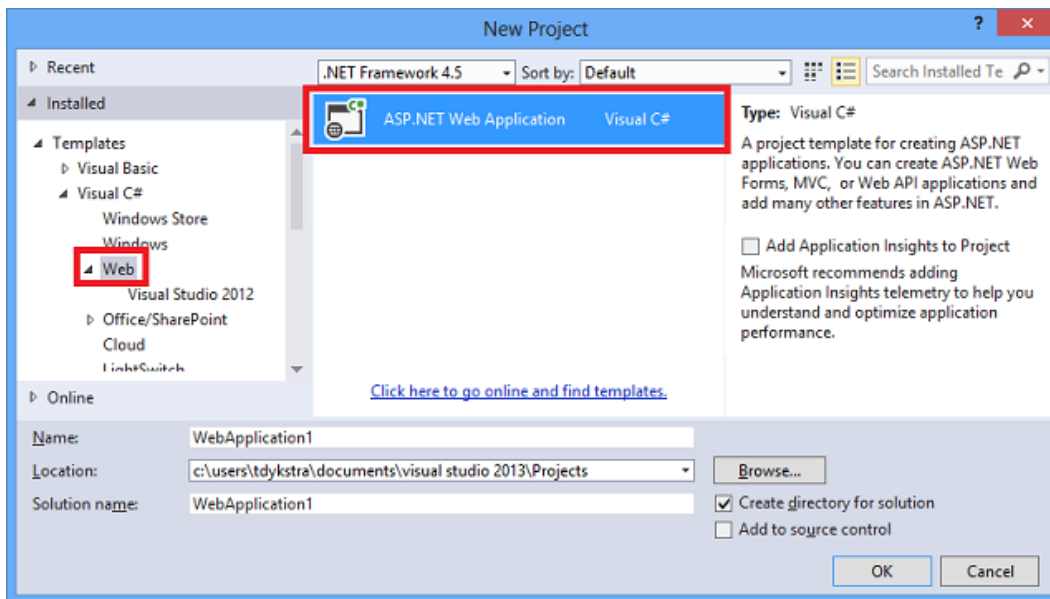


## LAB12: Componentes ASP.NET

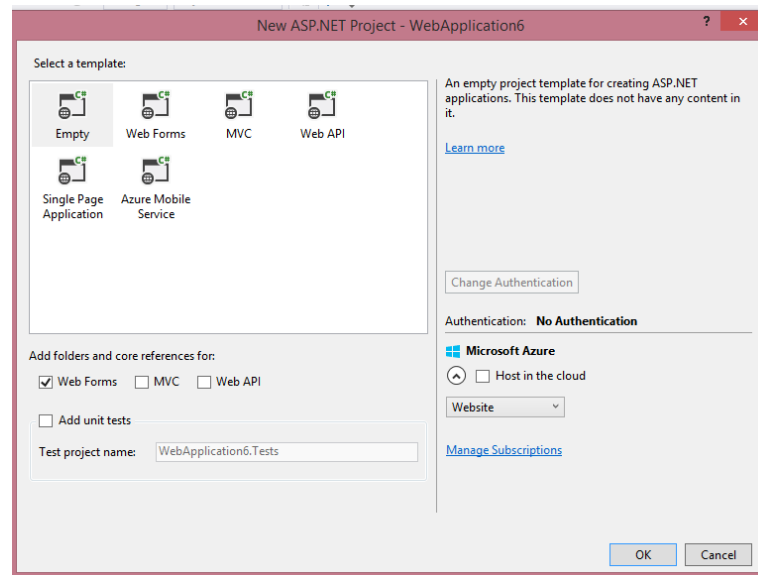
Neste laboratório vamos criar uma aplicação **ASP.NET** que utiliza o componente **DropDownList**, diversas opções de associação de conteúdo ao componente visual, bem como o uso dos componentes básicos de validação.

### Exercício 1 – Projeto Web Forms

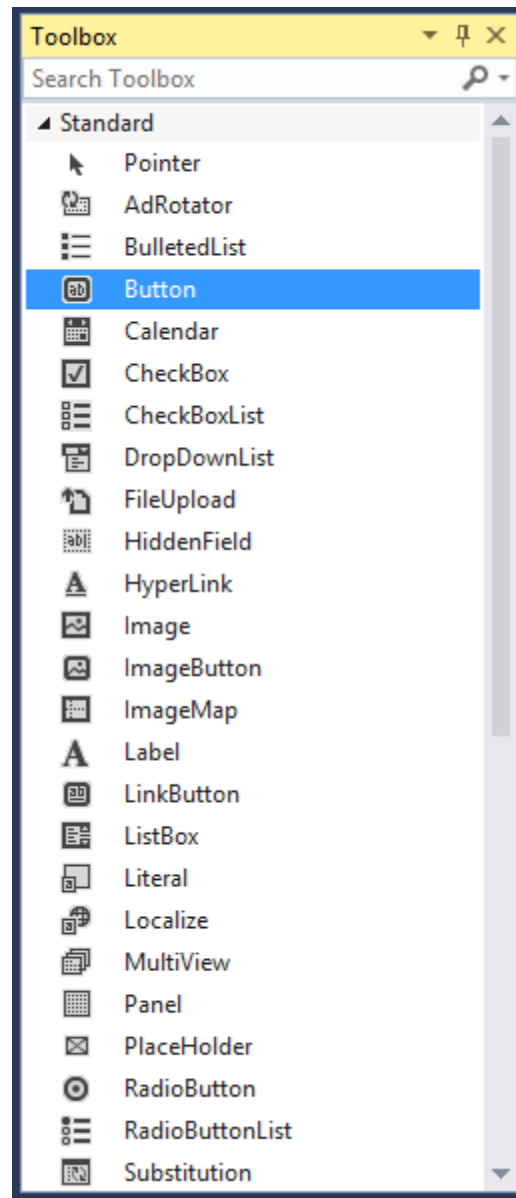
1. Dentro do Visual Studio selecione o menu: **File / New / Project**. Na lista **Installed Templates** (modelos), selecione **ASP.NET Web Application**. Escolha o nome do projeto e a pasta e clique OK.



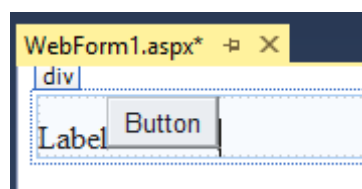
2. Selecione o template **Empty** e marque a opção **WebForms** e clique Ok, conforme figura abaixo:



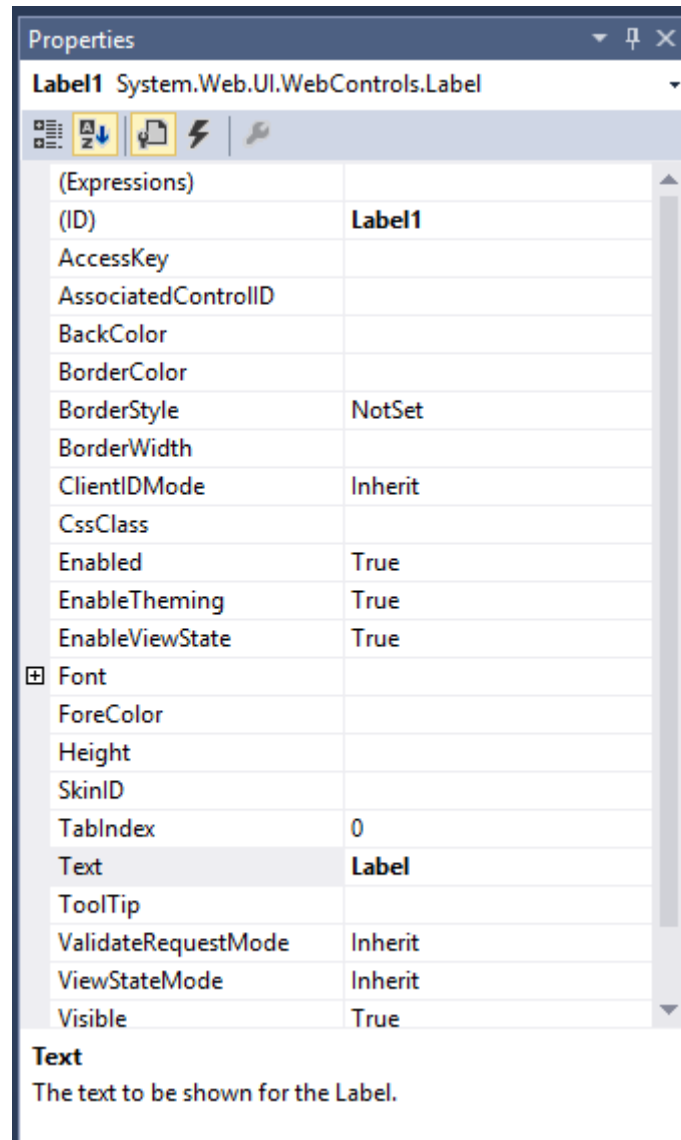
3. No Solution Explorer, clique com o botão direito sobre o nome do projeto que você criou e selecione (**Add / New Item**). Adicione um novo arquivo, do tipo **Web Form**, nomeado "WebForm1.aspx".
4. Selecione o menu **View / Toolbox**. Quando uma página ASP.NET está aberta na interface de design, podem ser vistos os seguintes controles (ou componentes) na Janela de **Toolbox**.



5. Arraste sobre a interface de design do um formulário criado, o controle **Label** na aba **Standard** da **Toolbox**. Após arraste um controle **Button**:



6. Selecione o menu **View / Properties Window**. Nesta janela você pode visualizar e alterar as propriedades dos controles do formulário:




**Figura 7** - Janela Properties

7. Clique sobre o rótulo (**Label**) de seu formulário altere as seguintes configurações (dentro da janela **Properties**):

<b>ID</b>	<b>lblMensagem</b>
<b>Text</b>	Primeira Aula, S2B

- Da mesma forma, altere as configurações do botão (**Button**)

<b>ID</b>	<b>btnExibir</b>
<b>Text</b>	Exibir

8. Execute e teste a aplicação. Note que não há “evento” associado ao clique do botão na aplicação.
9. Adicione ao formulário um novo **Label** altere a propriedade ID para *lblData*.
10. Selecione o componente **btnExibir**, clique em **Eventos** (botão ) da **Janela de Propriedades**. Nesta nova tela você pode acionar os eventos referentes ao **btnExibir**.
11. Escolha o evento **Click**. Dê duplo clique com o **botão esquerdo** do mouse sobre ele (o mesmo efeito é obtido com um duplo clique sobre o componente na janela de design). Adicione o código abaixo ao evento “**btnExibir\_Click**”:

```

publicpartialclassWebForm1 : System.Web.UI.Page
{
protectedvoid Page_Load(object sender, EventArgs e)
{
}

protectedvoid btnExibir_Click(object sender, EventArgs e)
{
    lblHora.Text = DateTime.Now.ToLongTimeString();
}
}

```

12. Execute e teste.

## Exercício 2 – Trabalhando com DropDownList

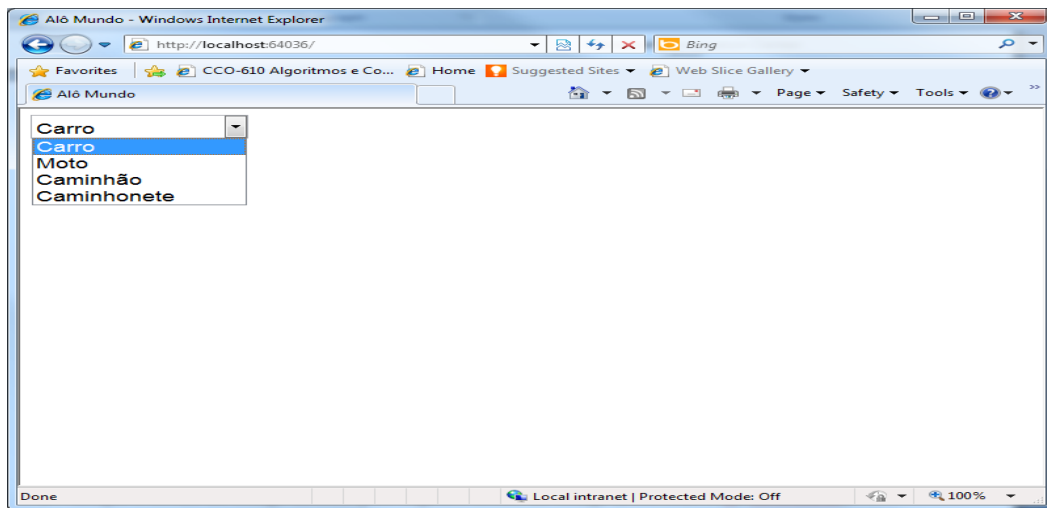
1. Selecione o projeto e adicione um novo formulário (**Add |Web Form**).
2. Arraste para o novo formulário um componente do tipo **Dropdown list** da **Toolbox**, na aba **Standard**, para o formulário criado.
3. Clique com o botão direito no arquivo e acesse **View Code**. Com isso deve ser exibido o **code-behind** do **Webform**.
4. No método **Page\_Load** adicione o seguinte trecho de código:

```

if (!Page.IsPostBack)
{
    string[] tiposAutomovel = newstring[4] { "Carro", "Moto",
    "Caminhão", "Caminhonete" };
    DropDownList1.DataSource = tiposAutomovel;
    DropDownList1.DataBind();
}

```

5. Aperte **Ctrl+F5** e veja o resultado. O **dropdown** deve conter os itens que estavam no array. O resultado deve ser similar à figura a seguir.



6. Com isso nós declaramos um array e inicializamos com os valores na própria declaração. Na sequência vamos fazer o mesmo porém inicializar o array após a declaração.
7. Troque o trecho de código anterior pelo seguinte trecho de código (mantendo **o if(!isPostBack)**):

```

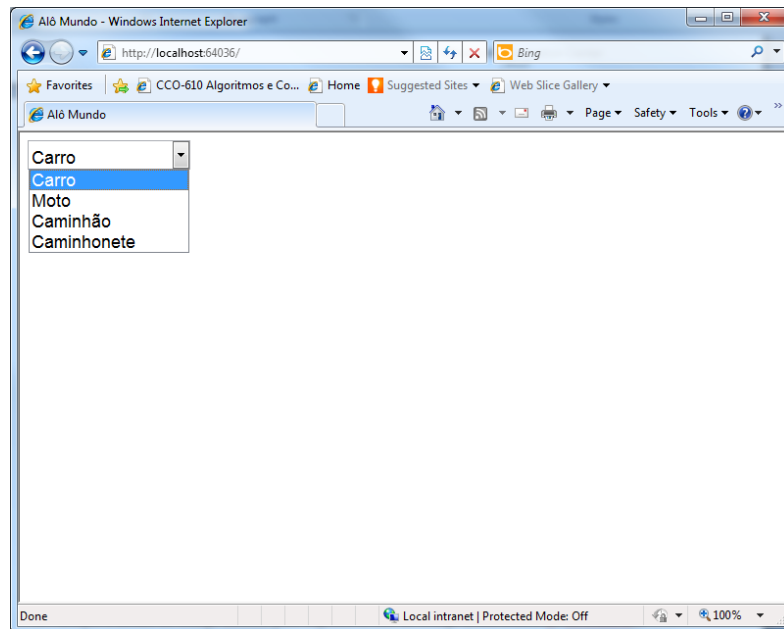
string[] tiposAutomovel = newstring[4];

tiposAutomovel[0] = "Carro";
tiposAutomovel[1] = "Moto";
tiposAutomovel[2] = "Caminhão";
tiposAutomovel[3] = "Caminhonete";

DropDownList1.DataSource = tiposAutomovel;
DropDownList1.DataBind();

```

8. Aperte **Ctrl+F5** e veja o resultado. O **dropdown** deve conter os itens que estavam no array. O resultado deve ser similar à figura a seguir.



9. O resultado esperado é o mesmo, porém os valores do array foram inicializados de outra maneira.
10. Volte ao modo **Design** do *webform*.
11. Clique com o botão esquerdo do mouse ao lado do **DropDown** e aperte enter para criar uma nova linha. Arraste um componente **Label** da **Toolbox** na aba **Standard** para a linha criada.
12. Volte para o **code-behind** do *webform* e adicione o seguinte trecho de código logo após a linha **DropDownList1.DataBind()**:

```
Label1.Text = "Total de itens: " + tiposAutomovel.Length.ToString();
```

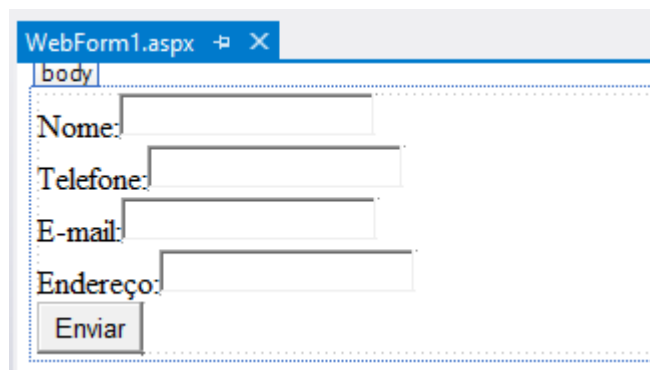
13. Execute e teste.
14. Adicione um novo componente **Label**.
15. Coloque o texto "Seleção do usuário:" no *label* criado.
16. Adicione um novo **label** ao lado do anterior. Coloque o nome **lblSelecao** e o texto vazio.
17. Selecione o componente **DropDownList** e habilite o mecanismo de **postback**. Faça isso clicando sobre a seta de **atividades rápidas** e marcando **Enable AutoPostBack**.
18. Dê um duplo clique sobre o **DropDownList** e acrescente o seguinte tratador de evento:

```
protectedvoid DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    lblSelecao.Text = DropDownList1.SelectedValue;
}
```

19. Aperte **Ctrl+F5** e veja o resultado.

### Exercício 3 – Componentes básicos de validação

1. Adicione um novo formulário ao projeto.
2. Adicione ao formulário quatro componentes do tipo TextBox (ID→ “txtNome”, ID → “txtTelefone”; ID→ “txtEmail”, ID→ “txtEndereco”).
3. Adicione quatro Labels (ID→ “lblNome”, Text→ “Nome”; ID→ “lblTelefone” Text → “Telefone”; ID→ “lblEmail”, Text→ “E-mail”; ID→ “lblEndereco”, Text→ “Endereço”).
4. Adicione um botão (ID→ “btnEnviar”, Text→ “Enviar”).
5. Arranje a página para se parecer com a seguinte figura:



#### Acrescentar componentes de validação

6. Adicione um *RequiredFieldValidator* da Toolbox na aba *Validation*, ao lado do *txtNome*. Mude suas propriedades da seguinte maneira:

```
ErrorMessage -> "Campo nome obrigatório"
ControlToValidate ->txtNome
ID ->rfvNome
Display ->None
Text ->*
```



7. Adicione um *RegularExpressionValidator* da *Toolbox*, ao lado do *txtTelefone*. Mude suas propriedades da seguinte maneira:

```
ErrorMessage -> "Telefone inválido"
ControlToValidate ->txtTelefone
ValidationExpression -> Aperte nos três pontinhos e selecione P.R.C. Phone Number
ID -> revTelefone
Display -> None
Text -> *
```

8. Adicione um *RegularExpressionValidator* da *Toolbox*, ao lado do *txtEmail*. Mude suas propriedades da seguinte maneira:

```
ErrorMessage -> "E-mail inválido"
ControlToValidate ->txtEmail
ValidationExpression -> Aperte nos três pontinhos e selecione Internet E-mail Address
ID -> revEmail
Display -> None
Text -> *
```

9. Adicione um *RequiredFieldValidator* da *Toolbox*, ao lado do *txtEndereco*. Mude suas propriedades da seguinte maneira:

```
ErrorMessage -> "Campo endereço obrigatório"
ControlToValidate ->txtEndereco
ID ->rfvEndereco
Display ->None
Text ->*
```

10. Adicione um *ValidationSummary* ao final da página.

11. Compile e execute o programa.

Exercício adicional:

- 1) Adicione um *Label* que será preenchido com as informações "Processamento ok" ou "Erro" caso o evento de click do botão tenha processado os dados corretamente. Utilize a propriedade *Page.IsValid* para garantir que a página foi validada antes de tentar utilizar os dados inseridos pelo usuário.