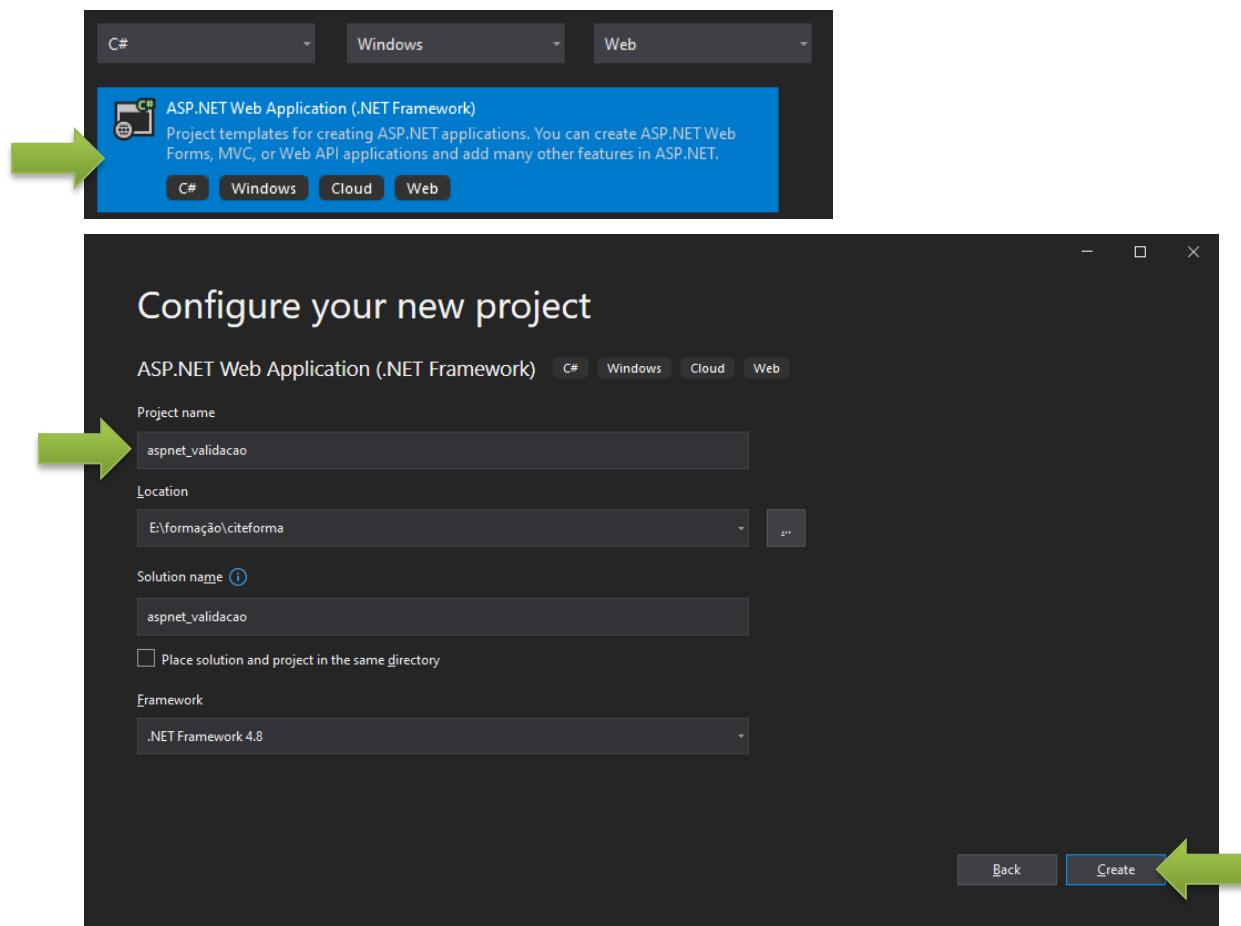


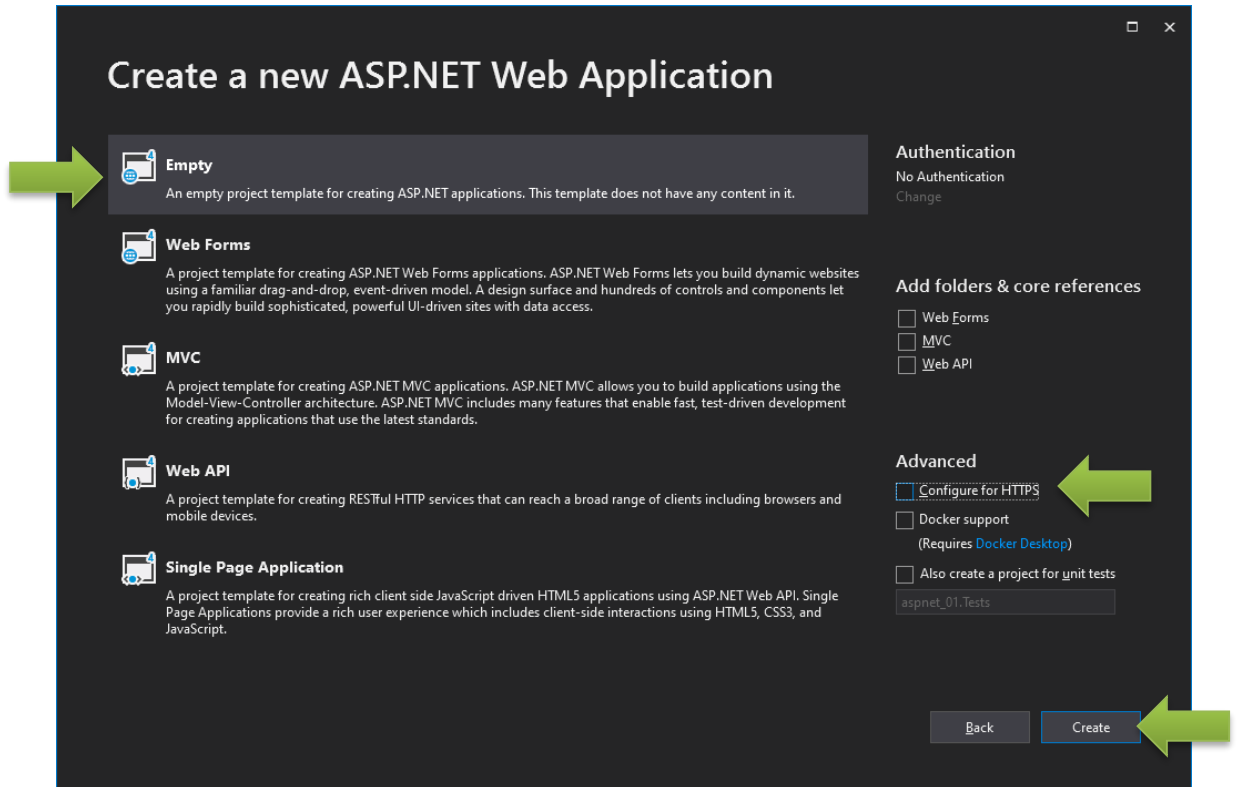
Aplicações Web em ASP.NET

Utilização de controlos de controlos de validação numa aplicação ASP.NET

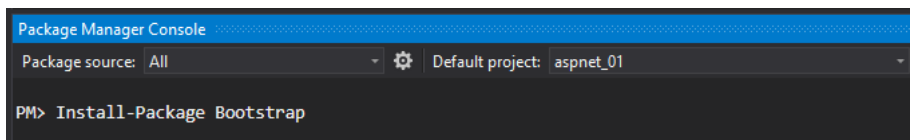
1. Crie uma **ASP.NET Web Application (.NET Framework)**; considere o nome **aspnet_validacao**;



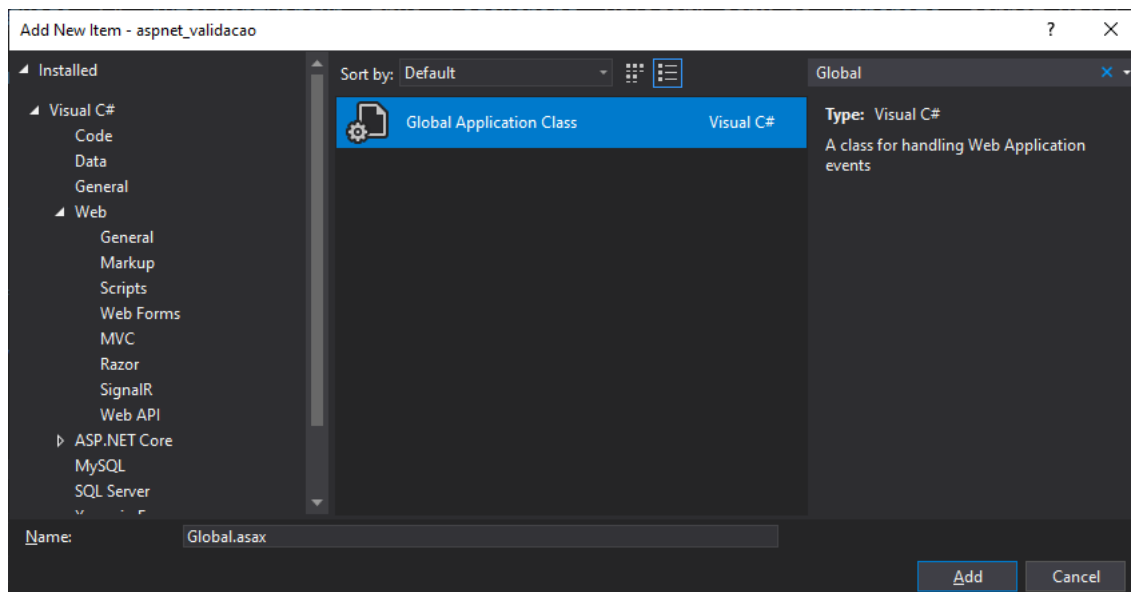
Selecione o tipo **Empty** e desative a opção **Configure for HTTPS**;



2. A partir da opção **Tools – NuGet Package Manager – Package Manager Console**, adicione ao projeto os **packages Bootstrap e jQuery**;



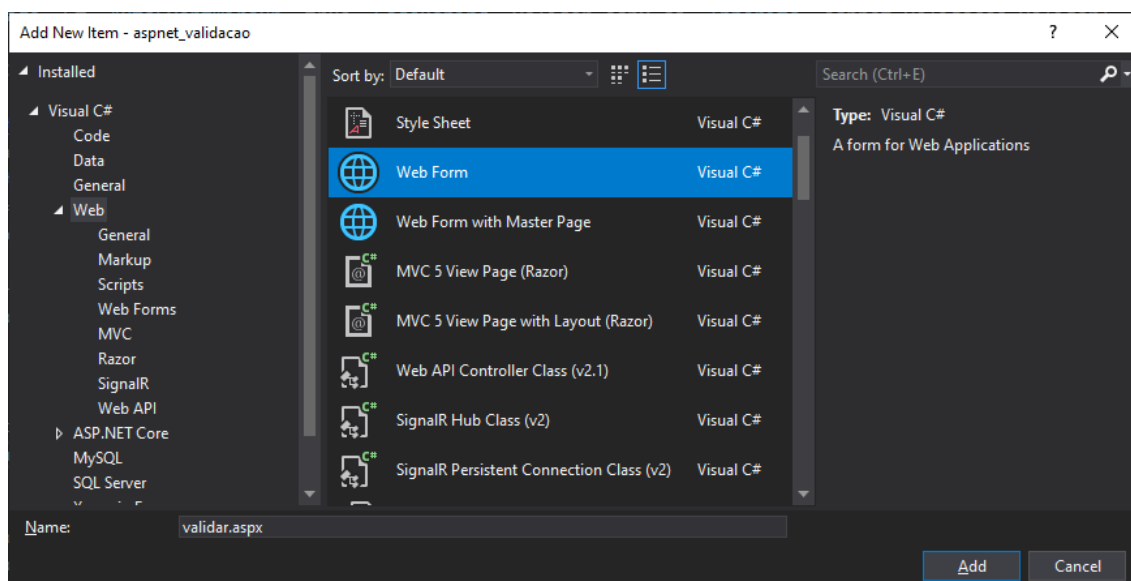
3. A partir da opção **Project – Add new item**, adicione ao projeto o ficheiro **Global Application Class**;



Adicione ao método `Application_Start` a configuração necessária à utilização de jQuery pelos controlos de validação;

```
protected void Application_Start(object sender, EventArgs e)
{
    ScriptManager.ScriptResourceMapping.AddDefinition("jquery",
        new ScriptResourceDefinition
        {
            Path = "~/scripts/jquery-3.7.1.min.js",
            DebugPath = "~/scripts/jquery-3.7.1.js",
            CdnPath = "http://ajax.microsoft.com/ajax/jquery/jquery-3.7.1.min.js",
            CdnDebugPath = "http://ajax.microsoft.com/ajax/jquery/jquery-3.7.1.js"
        });
}
```

4. A partir da opção **Project – Add new item**, adicione ao projeto um novo **Web Form**; considere o nome **validar.aspx**;



5. O formulário **validar.aspx** deverá permitir incluir um conjunto de controlos que permitam efetuar validações sobre valores de tipo numérico; neste sentido, crie o seguinte formulário:



Controlos de validação

Número inteiro > 0

Número real inferior a 25.5

Data anterior a 31/12/2021

Número inteiro entre 50 e 100

Data entre 1/1/2021 e 31/12/2021

Número inteiro

Número real

Data

Código Postal

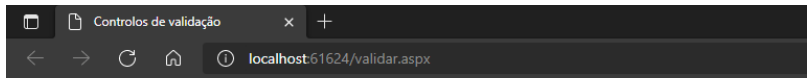
Telefone

Email

Nome (entre 5 e 20 caracteres)

CONTROLO	TIPO DE CONTROLO	ID
Número inteiro > 0	TextBox	textValor1
Número real inferior a 25.5	TextBox	textValor2
Data anterior a 31/12/2021	TextBox	textValor3
Número inteiro entre 50 e 100	TextBox	textValor4
Data entre 1/1/2021 e 31/12/2021	TextBox	textValor5
Número inteiro	TextBox	textValor6
Número real	TextBox	textValor7
Data	TextBox	textValor8
Código postal	TextBox	textValor9
Telefone	TextBox	textValor10
Email	TextBox	textValor11
Nome (entre 5 e 20 caracteres)	TextBox	textValor12
Validar	Button	buttonValidar

Nota: 1) todos os controlos são de preenchimento obrigatório e 2) considere a utilização de uma tabela (como forma de organização dos controlos) e a formatação (tabela e controlos) com as classes Bootstrap;



Controlos de validação

Número inteiro > 0	<input type="text"/>	Obrigatório
Número real inferior a 25.5	<input type="text"/>	Obrigatório
Data anterior a 31/12/2021	<input type="text"/>	Obrigatório
Número inteiro entre 50 e 100	<input type="text"/>	Obrigatório
Data entre 1/1/2021 e 31/12/2021	<input type="text"/>	Obrigatório
Número inteiro	<input type="text"/>	Obrigatório
Número real	<input type="text"/>	Obrigatório
Data	<input type="text"/>	Obrigatório
Código Postal	<input type="text"/>	Obrigatório
Telefone	<input type="text"/>	Obrigatório
Nome (entre 5 e 20 caracteres)	<input type="text"/>	Obrigatório
<input type="button" value="Validar"/>		

6. Para a obrigatoriedade em termos de preenchimento, adicione um controlo do tipo

RequiredFieldValidator junto do controlo **textValor1** e considere a seguinte configuração:

PROPRIEDADE	VALOR
ControlToValidate	textValor1
ErrorMessage	Obrigatório
Display	Dynamic
ForeColor	#CC0000

```
<asp:TextBox runat="server" ID="textValor1" class="form-control me-2"
    Style="width: 100px; display: inline;" />
<asp:RequiredFieldValidator ErrorMessage="Obrigatório"
    ControlToValidate="textValor1" runat="server" Display="Dynamic"
    ForeColor="#CC0000" />
```

7. Para a validação do controlo **textValor1** (Número inteiro > 0) adicione ao formulário um controlo do tipo **CompareValidator**; considere a seguinte configuração para este controlo:

PROPRIEDADE	VALOR
ControlToValidate	textValor1
ErrorMessage	Valor inteiro maior que zero.
Display	Dynamic
ForeColor	#CC0000
Type	Integer
Operator	GreaterThan
ValueToCompare	0

Validação - comparação e tipos de dados

Número inteiro > 0 Valor inteiro maior que zero.

8. Para a validação do controlo **textValor4** (Número inteiro entre 50 e 100) adicione ao formulário um controlo do tipo **RangeValidator**; considere a seguinte configuração para este controlo:

PROPRIEDADE	VALOR
ControlToValidate	textValor4
ErrorMessage	Número inteiro entre 50 e 100.
Display	Dynamic
ForeColor	#CC0000
Type	Integer
MinimumValue	50
MaximumValue	100

Número inteiro entre 50 e 100 Número entre 50 e 100.

9. Para a validação do controlo **textValor6** (Número inteiro) adicione ao formulário um controlo do tipo **CompareValidator**; considere a seguinte configuração para este controlo:

PROPRIEDADE	VALOR
ControlToValidate	textValor8
ErrorMessage	Defina um número inteiro.
Display	Dynamic
ForeColor	#CC0000
Type	Integer
Operator	DataTypeCheck

Número inteiro Defina um número inteiro.

10. Para a validação do controlo **textValor10** (Telefone) adicione ao formulário um controlo do tipo **RegularExpressionValidator**; considere a seguinte configuração para este controlo:

PROPRIEDADE	VALOR
ControlToValidate	textValor10
ErrorMessage	Número inválido.
Display	Dynamic
ForeColor	#CC0000
ValidationExpression	[2][0-9]{8}
Operator	DataTypeCheck

11. Para a validação do controlo **textValor11** (Nome entre 5 e 20 caracteres), comece por definir a seguinte função JavaScript:

```
<script>
function Validar(source, arguments)
{
    if ((arguments.Value.length >= 5) && (arguments.Value.length <= 20))
        arguments.IsValid = true;
    else
        arguments.IsValid = false;
}
</script>
```

Adicione ao formulário um controlo do tipo **CustomValidator**; considere a seguinte configuração para este controlo:

PROPRIEDADE	VALOR
ControlToValidate	textValor11
ErrorMessage	O nome deve ter entre 5 e 20 caracteres.
Display	Dynamic
ForeColor	#CC0000
ClientValidationFunction	Validar

A partir das propriedades defina o **Event handler** para o evento **Server Validate**; repita no servidor o processo de validação efetuado na função **JavaScript**;

```
protected void CustomValidatorNome_ServerValidate
(object source, ServerValidateEventArgs args)
{
    if ((args.Value.Length >= 5) && (args.Value.Length <= 20))
        args.IsValid = true;
    else
        args.IsValid = false;
}
```