

Página de Erro

Abra o Visual Studio.NET

Clique em File > New Web Site

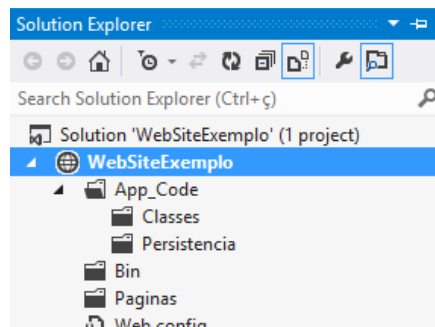
Selecione

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

Clique em OK

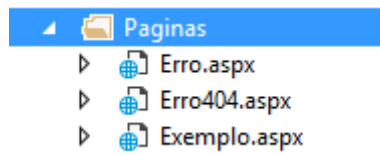
Adicione as pastas na Solution Explorer:

- 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)



Adicione 3 web forms na pasta Paginas:

- Exemplo.aspx (página com conteúdo)
- Erro.aspx (página para erro genérico)
- Erro404.aspx (página caso uma URL não seja encontrada)



Identifique cada página alterando o conteúdo delas.

No web.config, configure-o para tratar os erros:

```
<system.web>
  <compilation debug="true" targetFramework="4.5" />
  <httpRuntime targetFramework="4.5" />
  <customErrors mode="On" defaultRedirect="Paginas/Erro.aspx">
    <error statusCode="404" redirect="Paginas/Erro404.aspx"/>
  </customErrors>
</system.web>
```

Foi configurada uma página genérica de erro (Erro.aspx) e uma página de erro específica (Erro404.aspx). Outras páginas de erros específicos poderiam ser configuradas.

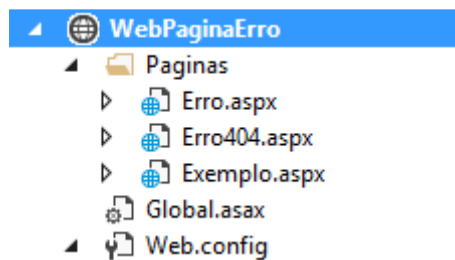
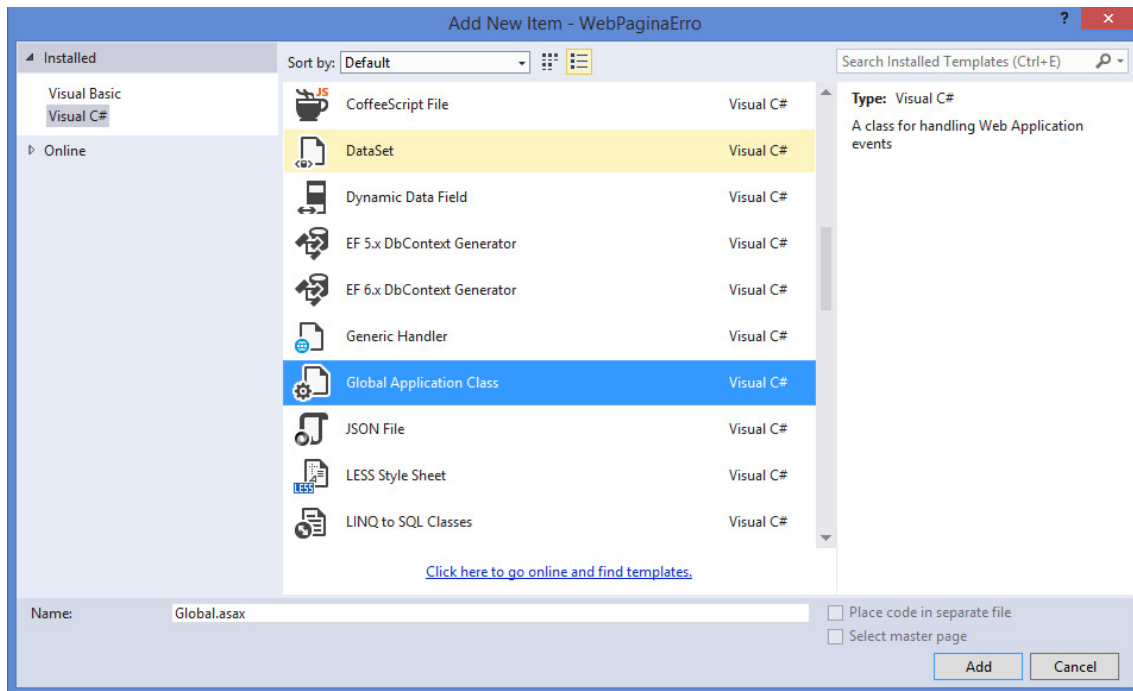
Coloque a página Exemplo.aspx para ser a primeira a ser executada.

Execute a aplicação

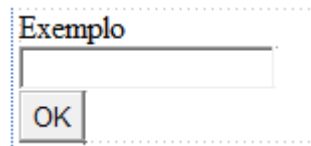
- Acesse a página Exemplo.aspx
- Acesse a página Exemplo1.aspx (não existente – será redirecionado para Erro404.aspx)
- A outra página de erro será chamada caso aconteça qualquer outro tipo de erro.

Capturando qualquer erro da aplicação

Adicione na Solution Explorer, um novo item.



Na página Exemplo.aspx, adicione um TextBox e um Button para podermos simular outro tipo de erro.



No Click do botão, coloque:

```
Response.Write(Convert.ToInt32(TextBox1.Text));
```

No método `Application_Error`, no arquivo `Global.asax`, coloque:

```
void Application_Error(object sender, EventArgs e)
{
    Exception exc = Server.GetLastError();

    // Handle HTTP errors
    if (exc.GetType() == typeof(HttpException))
    {
        if (exc.Message.Contains("NoCatch") ||
exc.Message.Contains("maxLength"))
            return;
        //Server.Transfer("HttpErrorPage.aspx");
    }
    else
    {
        Response.Write("<h2>Mensagem de erro</h2>\n");
        Response.Write(
            "<p>" + exc.Message + "</p>\n");
        Response.Write(
            "<p>" + exc.InnerException + "</p>\n");
        Response.Write("<a href='~Exemplo.aspx'>" +
            "Default Page</a>\n");
        Server.ClearError();
    }
}
}
```

Execute a aplicação

- Acesse a página `Exemplo.aspx` (acesso normal)
- Acesse a página `Exemplo1.aspx` (não existente – erro 404)
- Volte a página `Exemplo.aspx`, preencha o `TextBox` com um número e clique no botão (funcionamento normal)
- Na página `Exemplo.aspx`, preencha o `TextBox` com um texto e clique no botão (vai dar erro na página – continue a execução para que o método `Application_Error` possa ser executado)

Lembre-se de usar `Try Catch` sempre que possível para tratar as exceções no escopo local.

Outra opção interessante é salvar as exceções capturadas num banco de dados para que possam ser monitoradas posteriormente.

Exercício

- 1) Implemente na sua aplicação o tratamento de exceção com páginas de erro personalizadas