

Controle de acesso em .NET

No modelo de autenticação das aplicações ASP.NET, somente usuários autenticados (Nome e Senha) têm acesso a páginas do aplicativo. Usuários não autenticados são redirecionados automaticamente para a página de login especificada no web.config.

REF: [https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-1.1/fkytk30f\(v=vs.71\)](https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-1.1/fkytk30f(v=vs.71))

Abril/2021

FACULDADE DE TECNOLOGIA DE AMERICANA

Prof. Ms Diógenes de Oliveira

Controle de acesso em aplicações ASP.NET

- Controlar o acesso aos arquivos de uma PASTA
 - Regras inseridas no arquivo **web.config** para que somente usuários autenticados tenham acesso a determinados arquivos e/ou pasta;
- Criar o arquivo de autenticação: Login.aspx
 - Identificar e autenticar o usuário;
 - Criar o ticket de autorização para a **sessão do usuário**.
- Definir as regras de acesso no menu

web.config

1. Indicar o formulário que fará a autenticação do usuário.

```
<system.web>  
    <!-- DEFINE O FORM DE LOGIN-->  
    <authentication mode="Forms">  
        <forms loginUrl="~/Login.aspx"></forms>  
    </authentication>  
</system.web>
```

web.config

2. Definir as regras de segurança.

- Segurança de pasta(s) e/ou arquivo(s);
- Liberação de acesso a pasta(s) e/ou arquivo(s).

Exemplo1:

Nega acesso a todos os arquivos da pasta **Admin**, exceto ao usuário que for autorizado explicitamente usando as regras de autenticação inseridas no formulário de Login.

```
<!-- REGRAS DE PROTEÇÃO DA PASTA ADMIN-->
<location path="Admin">
    <system.web>
        <authorization>
            <deny users="?" />
        </authorization>
    </system.web>
</location>
```

Exemplo2.

Esta seção fornece ao usuário não autenticado acesso apenas à página Default1.aspx.

```
<!-- LIBERA O ACESSO AO ARQUIVO Default1.aspx -->
```

```
<location path="Default1.aspx">  
    <system.web>  
        <authorization>  
            <allow users="*" />  
        </authorization>  
    </system.web>  
</location>
```

Exemplo3.

Esta seção fornece ao usuário não autenticado acesso a todos os arquivos armazenados na pasta **Dir1**

```
<!-- LIBERA O ACESSO A PASTA Dir1 -->
```

```
<location path="Dir1">  
    <system.web>  
    <authorization>  
        <allow users="*" />  
    </authorization>  
    </system.web>  
</location>
```

Login.aspx

```
<asp:Content ID="Content" ContentPlaceHolderID="Conteudo"
runat="server">
  <div class="row margin-top-60">
    <div class="col-3">
      <div class="box padding-20">
        <h1>Entrar</h1>
        <br />
        <br />
        <asp:Label ID="Erro" runat="server" ></asp:Label>
        <br />
        <label>NOME DE ACESSO</label>
        <asp:TextBox ID="NomeAcesso" runat="server"></asp:TextBox>
        <label>SENHA</label>
        <asp:TextBox ID="Senha" TextMode="Password"
runat="server"></asp:TextBox>
        <br />
        <br />
        <asp:Button ID="Entrar" OnClick="Entrar_Click"
runat="server" Text="Entrar" />
        <br />
      </div>
    </div>
  </div>
</asp:Content>
```


Login.aspx.cs

```
if (NomeAcesso.Text == "Admin" && Senha.Text == "12345")
{
    // Cria a variavel de sessão para identificar que o usuário está autenticado e
    // permitir a exibição das opções do menu.
    Session["autenticado"] = "";
    // 1. Inicializa a classe de autenticação
    System.Web.Security.FormsAuthentication.Initialize();
    // 2. CRIAR O TICKET
    FormsAuthenticationTicket ticket = new FormsAuthenticationTicket(1, "admin",
    DateTime.Now, DateTime.Now.AddMinutes(20), false,
    FormsAuthentication.FormsCookiePath);

    // 3. CRIPTOGRAFA P TICKET E GRAVAR NO COOKIE DO NAVEGADOR
    Response.Cookies.Add(new HttpCookie(FormsAuthentication.FormsCookieName,
    FormsAuthentication.Encrypt(ticket)));

    // Redireciona para o form que o usuário tentou acessar
    Response.Redirect(FormsAuthentication.GetRedirectUrl("Admin", false));
}
else
{
    Msg.Text = "Dados de acesso invalidos";
}
```

