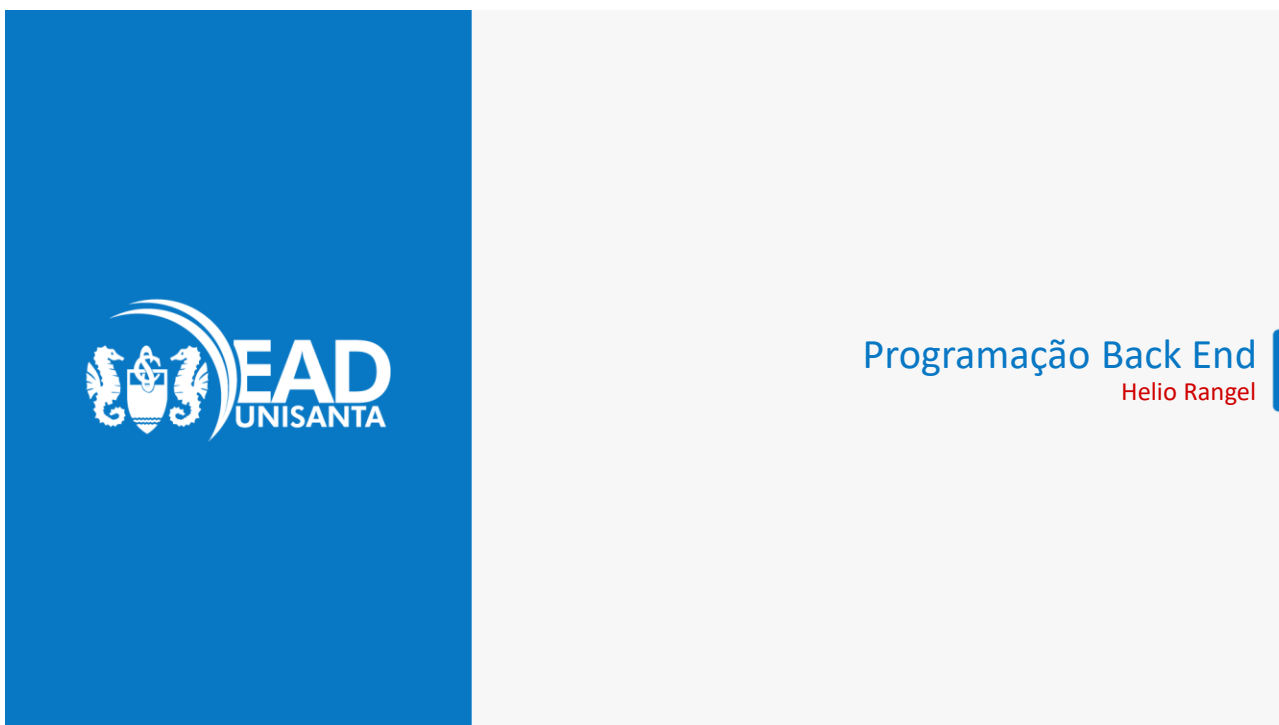




1



2



Login e Menu

- Seguindo a sequência lógica do projeto, vamos primeiramente estudar o código da funcionalidade de login. Assumimos aqui que o aluno possui algum conhecimento prévio da Linguagem C# e de programação orientada a objetos. Neste ponto recomenda-se que, caso o aluno se sinta inseguro com a linguagem, procure não deixar passar nenhuma dúvida indo buscar o conhecimento necessário no material específico de C#. O Código completo do projeto está a disposição do aluno nos anexos. No tópico IV, apresentamos as classes **Usuario**, **Lancamento** e **LivroCaixa** que serão citadas neste e nos próximos tópicos (VI, VII e VIII)

Entendendo o pnLogin

- Este é um trecho do código do index.aspx, ele está visível no momento que o form é carregado.

```
<asp:Panel runat="server" ID="pnLogin" Visible="true">
  <table style="width: 400px">
    <tr>
      <td>&nbsp;&nbsp;&nbsp;&nbsp;<asp:Label ID="lbUsuario" runat="server" Text="Usuário"></asp:Label>
      </td>
      <td>&nbsp;&nbsp;&nbsp;&nbsp;<asp:TextBox ID="txUsuario" runat="server" Width="120px"></asp:TextBox>
      </td>
    </tr>
    <tr>
      <td>&nbsp;&nbsp;&nbsp;&nbsp;<asp:Label ID="lbSenha" runat="server" Text="Senha"></asp:Label>
      </td>
      <td>&nbsp;&nbsp;&nbsp;&nbsp;<asp:TextBox ID="txSenha" runat="server" TextMode="Password" Width="120px"></asp:TextBox>
      </td>
    </tr>
    <tr>
      <td>&nbsp;&nbsp;&nbsp;&nbsp;<td>&nbsp;&nbsp;&nbsp;&nbsp;<asp:Button ID="btOk" runat="server" Text="Ok" OnClick="btOk_Click" Width="120px" />
      </td>
    </tr>
  </table>
</asp:Panel>
```

Visual representation of the login form. It shows a table with two columns. The first column contains labels 'Usuário' and 'Senha'. The second column contains input fields: a text box with 'jose.alberto' and a password box with '*****'. Below the password box is an 'Ok' button. Red lines connect the code snippets in the previous block to these UI elements.



5

Ao Clicar no Botão OK

```
protected void btOk_Click(object sender, EventArgs e) {
    pnTrocaSenha.Visible = false;
    lbMensagem.Text = String.Empty;
    using (Conexao con = new Conexao(null)) {
        con.open();
        Usuario usuario = Usuario.busca(txUsuario.Text.Trim(), txSenha.Text.Trim(), con);
        if (usuario == null) {
            lbMensagem.Text = "Usuário / senha não cadastrado!";
            return;
        }
        Session["usuario"] = usuario;
        if (usuario.senha == usuario.cpf) { // Primeiro acesso
            lbMensagem.Text = "Este é o seu primeiro acesso, favor definir" +
                " uma nova senha com no mínimo 6 e no máximo 10 caracteres";
            pnTrocaSenha.Visible = true;
            return;
        }
        if (usuario != null && usuario.senha != usuario.cpf) {
            HttpCookie cookie = Request.Cookies["empresa"];
            if (cookie == null || cookie.Value.ToString() == String.Empty) {
                boxOkCancelaTextoToJavaScript("Digite o nome da instituição", "digiteNome");
            }
            else {
                Session["empresa"] = empresa = cookie.Value;
                pnLogin.Visible = false;
                if (usuario.perfil == "ADM") {
                    pnMenu.Visible = true;
                }
                else {
                    Response.Redirect("fluxoDeCaixa.aspx", false);
                    return;
                }
            }
        }
    }
}
```

Este é o código que será executado no momento que
O usuário clicar no botão OK

Depois de abrir a conexão para acessar o banco de dados, o sistema tenta instanciar um usuário passando o usuário e senhas digitados. Se o retorno for null, significa que o usuário não conseguiu se autenticar e retorna;

Uma vez o usuário autenticado, verifica se a senha e o CPF do usuário são idênticos. Caso positivo, significa que este foi o primeiro acesso, mostra o painel de troca de senha e retorna sem prosseguir.

Neste ponto, com o usuário já validado, verifica se o cookie com nome da empresa já está carregado. Caso não esteja, solicita que o usuário, seja quem for, digite o nome da instituição. Para isso, a função digitaNome() será chamada:

```
String evento = (this.Request["__EVENTTARGET"] == null) ? String.Empty : t
String resposta = (this.Request["__EVENTARGUMENT"] == null) ? String.Empty
```

```
if (evento == "digiteNome")
{
    if (resposta == String.Empty)
    {
        lbMensagem.Text = "Por favor, digite o nome da Instituição!";
    }
    else
    {
        HttpCookie cookie = new HttpCookie("empresa");
        Session["empresa"] = empresa = cookie.Value = resposta;
        DateTime dtNow = DateTime.Now;
        TimeSpan tsMinute = new TimeSpan(1000, 1, 0, 0);
        cookie.Expires = dtNow + tsMinute;
        Response.Charset = "UTF-8";
        Response.Cookies.Add(cookie);
    }
}
```



6

Trocando a senha inicial

Ao clicar no botão de confirmação de troca de senha inicial, o sistema compara as duas senhas digitadas para evitar engano... Verifica de o número mínimo e máximo de dígitos está correto. Estando tudo ok, o usuário é carregado na sessão, a nova senha é atribuída e o método de atualização do usuário é chamado. Na sequência, o painel pnTrocaSenha é escondido e a mensagem de sucesso é mostrada.

Note que, sempre que o sistema faz consulta/atualização no banco de dados, se utiliza um bloco try/catch para tratar as eventuais exceções.

Redefinição de Senha

Nova senha
Confirmação

Este é o seu primeiro acesso, favor definir uma nova senha com no mínimo 6 e no máximo 10 caracteres

```
protected void btTrocaSenha_Click(object sender, EventArgs e)
{
    lbMensagem.Text = String.Empty;
    try
    {
        if (txSenha1.Text.Trim().CompareTo(txSenha2.Text.Trim()) != 0)
        {
            lbMensagem.Text = "Senha de confirmação não confere!";
            return;
        }
        if (txSenha1.Text.Trim().Length < 6 || txSenha1.Text.Trim().Length > 10)
        {
            lbMensagem.Text = "A nova senha digitada deve ter no mínimo 6 e no máximo 10 dígitos!";
            return;
        }
        Usuario usuario = (Usuario)Session["usuario"];
        usuario.senha = txSenha1.Text.Trim();
        using (Conexao con = new Conexao(usuario))
        {
            con.open();
            usuario.atualiza(con);
            pnTrocaSenha.Visible = false;
            lbMensagem.Text = "Senha reiniciada com sucesso!";
        }
    }
    catch (Exception)
    {
        lbMensagem.Text = "Erro inesperado redefinindo senha!";
    }
}
```



7

Como funciona o método de busca?

```
public static Usuario busca(String login, String senha, Conexao con)
{
    try
    {
        string []args = {login, senha};
        String sql = String.Concat("SELECT ", campos, ",idUsuario FROM ", nomeTabela,
            " WHERE login=@1 AND senha=@2");
        DataTable dt = Conexao.executaSelect(con, sql,args);
        if (dt.Rows.Count == 0) return null;
        DataRow[] r = dt.Select();
        Usuario item = new Usuario(r[0][0].ToString(), r[0][1].ToString(), r[0][2].ToString(),
            r[0][3].ToString(), r[0][4].ToString());
        item.idUsuario = Int16.Parse(r[0][5].ToString());
        return item;
    }
    catch (Exception)
    {
        throw;
    }
}
```

- O método busca acima, utiliza construção de *queries* para evitar *sql injection*, recebendo os strings login e senha e repassando para o *executaSelect()* (um método genérico da classe *Conexao*), cujo código é mostrado a direita.
- O *executaSelect()* recebe a conexão aberta, a *query* ou *array* de strings (args) que irão substituir os símbolos @1 e @2, retornando um *DataTable* de onde obtemos, caso a consulta retorne algo, o *array* de *DataRow* que finalmente nos fornece os strings resultantes da consulta.
- No caso de sucesso, o usuário é instanciado e retornado para o chamador. Caso contrário o método retorna *null* sinalizando que não existe um usuário com o conjunto usuário/senha cadastrado.

```
public static DataTable executaSelect(Conexao con, string sql, String[] args) {
    try
    {
        DataTable dt = new DataTable();
        SqlCommand comando = new SqlCommand(sql, con.con());
        comando.Transaction = con.getTransaction();
        comando.CommandType = CommandType.Text;
        comando.Prepare();
        SqlDataAdapter da = new SqlDataAdapter();
        da.SelectCommand = comando;
        if (args != null) {
            int chave = 1;
            foreach (string s in args) {
                string seq = "@" + chave++;
                comando.Parameters.AddWithValue(seq, s);
            }
        }
        try {
            da.Fill(dt);
        }
        catch (Exception) {
            throw;
        }
        finally {
            dt.Dispose();
            da.Dispose();
            comando.Dispose();
        }
        return dt;
    }
    catch (Exception) {
        throw;
    }
}
```

Construção de Queries

Evitamos sempre que possível utilizar concatenação de strings obtidos diretamente da digitação dos usuários.

O recurso disponível para isso é a Construção de Queries que pode ser vista em uso aqui utilizando a classe de construção *SqlCommand* em conjunto com *SqlDataAdapter*.



8

Trabalhando com Cookies

- Uma vez registrado o usuário, o sistema verifica se existe um *cookie* cadastrado chamado **"empresa"**. Caso o *cookie* ainda não esteja indefinido, abre uma caixa de diálogo do navegador solicitando a definição do nome da instituição que opera o sistema, caso contrário coloca na sessão o nome da instituição para ser usado nas demais telas e prossegue escondendo o pnLogin;
- `boxOkCancelaTextoJavaScript()` é um método utilitário utilizado para montar uma caixa de diálogo ok/cancela com digitação de texto sempre que necessário.

```
public void boxOkCancelaTextoJavaScript(string mens, string funcao)
{
    StringBuilder script = new StringBuilder(String.Empty);
    script.Append("<script type='text/javascript' language='javascript'> ");
    script.Append("var retorno = prompt('" + mens.Replace("<br />", "\\n") + "', '');");
    script.Append("_doPostBack('" + funcao + "', retorno);");
    script.Append("</script>");
    ScriptManager.RegisterStartupScript(Page, Page.GetType(), "msg", script.ToString(),
        false);
}
```

```
HttpCookie cookie = Request.Cookies["empresa"];
if (cookie == null || cookie.Value.ToString() == String.Empty)
{
    boxOkCancelaTextoJavaScript("Digite o nome da instituição", "digiteNome");
}
else
{
    Session["empresa"] = empresa = cookie.Value;
    pnLogin.Visible = false;
    if (usuario.perfil == "ADM")
    {
        pnMenu.Visible = true;
    }
    else
    {
        Response.Redirect("fluxoDeCaixa.aspx", false);
        return;
    }
}
```

- Caso o *cookie* já esteja definido, o sistema verifica se o usuário é perfil **ADM**, se for, mostra o painel do menu, caso contrário redireciona diretamente para o form **fluxoDeCaixa.aspx**



9

O Menu

- Mantendo a ideia de um **Front End** minimalista, adotamos um menu com três opções, não dinâmicas, escritas como *link* diretamente no código aspx.
- Existem disponíveis no mercado muitos menus de vários tipos, fáceis de configurar, dinâmicos e poderosos.



```
<asp:Panel runat="server" ID="pnMenu" Visible="false">
    <table style="width: 300px">
        <tr>
            <td style="text-align: center">
                <h2>Menu</h2>
            </td>
        </tr>
        <tr>
            <td style="text-align: center">
                <a href='cadastroUsuarios.aspx' title="clique aqui para cadastro de usuários.">
                    Cadastro de Usuários</a>
            </td>
        </tr>
        <tr>
            <td style="text-align: center">
                <a href='fluxoDeCaixa.aspx' title="clique aqui lançamentos do Fluxo de Caixa.">
                    Fluxo de Caixa</a>
            </td>
        </tr>
        <tr>
            <td style="text-align: center">
                <a href='index.aspx?sair=sair' title="Sair">Sair</a>
            </td>
        </tr>
    </table>
</asp:Panel>
```

```
String sair = Request.QueryString["sair"];
if (sair != null && sair == "sair")
{
    Session["usuario"] = null;
    pnTrocaSenha.Visible = false;
    pnLogin.Visible = true;
    txUsuario.Text = txSenha.Text = String.Empty;
    Response.Redirect("index.aspx", false);
}
```

- Para cada opção do menu, devemos capturar o *postback* no método `FormLoad()` do Form como mostra o exemplo da opção "sair"



10