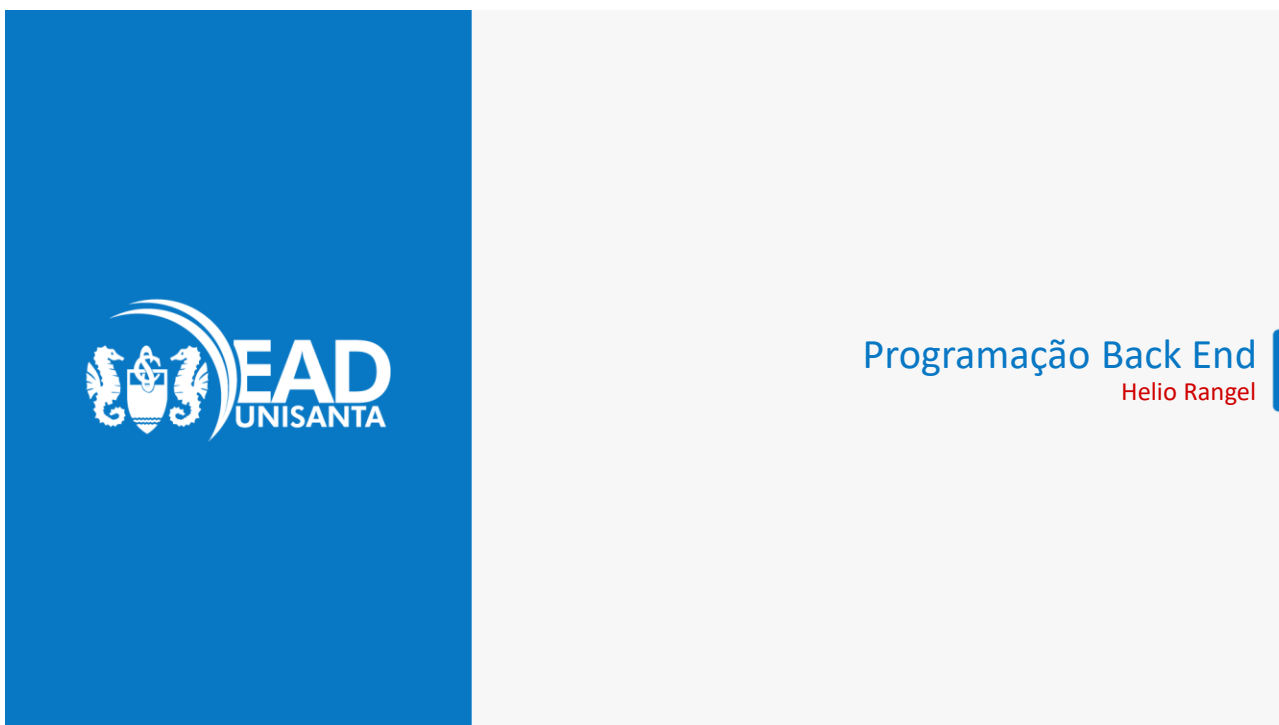




1



2

Cadastro de Usuários - Requisitos

- O Cadastro de Usuários, conforme os requisitos do projeto, deve permitir cadastro de novos usuários, edição de dados de usuários existentes e exclusão.
- No caso de novos usuários, basta digitar os dados do novo usuário e confirmar clicando no botão *Incluir*.
- Para editar ou excluir, é necessário fazer a busca do registro. Quando se procede uma busca, os botões de *Atualizar* e *Excluir* são mostrados.
- Quando o usuário cadastra um novo registro, edita ou exclui um usuário existente, o sistema volta sempre para o modo inicial de inclusão.

Cadastrando novos usuários

- A situação *inicial* da tela é de inserção de registros. O usuário digita os dados obrigatórios e clica no botão de inserir - **btNovo_Click()**.
- Os campos são validados rodando o método **validarCampos()** que se encontrar algum campo inconsistente, mostra uma mensagem na tela e retorna um **false** para o evento **btNovo_Click()** que, neste caso, retorna sem proceder nenhuma alteração.
- Estando tudo ok, instancia um objeto **Usuario**, popula com os dados digitados na tela, cadastra no Banco de Dados e na sequência, mostra no *grid* de usuários cadastrados;
- Observe que o sistema não permite a inserção de mais de um usuário com o mesmo *login* e antes de salvar o CPF, retira os pontos e a barra caso tenham sido digitados pelo usuário.

```
protected void btNovo_Click(object sender, EventArgs e)
{
    if (!validarCampos()) return;
    Usuario usuario = (Usuario)Session["usuario"];
    try
    {
        using (Conexao con = new Conexao(usuario))
        {
            con.open();
            Usuario u = Usuario.busca(con, txLogin.Text);
            if (u != null && u.idUsuario > 0)
            {
                lblMensagem.Text = "Já existe um usuário com este login: " + txLogin.Text;
                return;
            }
            txCPF.Text = txCPF.Text.Replace(".", "").Replace("-", "");
            u = new Usuario(txNome.Text, txLogin.Text, txCPF.Text, rbAdm.Checked ? "ADM" : "USU", txCPF.Text);
            u.inserir(con);
            limpa();
            lblMensagem.Text = "Novo usuário cadastrado com sucesso!";
            relatorioUsuarios.Text = usuario.montaTabela((String)Session["empresa"], con);
        }
    }
    catch (Exception ex)
    {
        if (usuario.perfil == "ADM")
        {
            lblMensagem.Text = ex.Message;
        }
        else
        {
            lblMensagem.Text = "Erro cadastrando usuário.";
        }
    }
}
```

Método sobrecarregado que
cheça se o login já existe na
base!

Retirando pontos e barra do
CPF



O método validarCampos()

- Este é o método de validação utilizado nos momentos de inserção e edição/atualização de usuários.
- Ao encontrar alguma inconsistência, mostra mensagem da ocorrência e retorna imediatamente.
- Em sistemas profissionais, a validação representa um dos momentos mais importantes e necessários para um sistema livre de erros e robusto.
- Note que não implementamos uma rotina de validação de CPF. Certamente este seria o momento certo de utilizá-la. O teste de validade do CPF é feito aplicando-se um algoritmo conhecido por Módulo 11. Que tal alterar o projeto para viabilizar esta verificação? (tente não utilizar um pronto da internet!)

```
private bool validarCampos()
{
    if (rbAdm.Checked == false && rbUsu.Checked == false)
    {
        lblMensagem.Text = "Selecione o perfil do usuário";
        return false;
    }
    if (txNome.Text.Trim() == String.Empty)
    {
        lblMensagem.Text = "Digite o nome do Usuário";
        return false;
    }
    if (txLogin.Text.Trim() == String.Empty)
    {
        lblMensagem.Text = "Digite o login do usuário";
        return false;
    }
    if (txCPF.Text.Trim() == String.Empty)
    {
        lblMensagem.Text = "Digite o CPF do usuário";
        return false;
    }
    return true;
}
```



Buscando Usuários para editar

- O sistema permite **três tipos de busca**: por nome; login ou pelo número sequencial do usuário.
- Primeiro passo é buscar o usuário logado, inicializar o status da tela.
- Em seguida verificamos se o usuário digitou alguma coisa no campo de busca.
- Depois de aberta a conexão, verificamos se o usuário digitou um número válido, o que é a "deixa" pra decidir que ele está tentando busca por número sequencial. Neste caso, **faz a busca diretamente pelo id e esta feito**.
- No caso de digitar um texto, procuramos na base de dados, utilizando comando like do SQL varrendo os campos de nome e login, se existe uma lista correspondente com **apenas um registro**. Se sim, selecionamos o primeiro e único da lista.
 - Se não**, avisamos que mais de um usuário corresponde a busca e retornamos sem sucesso.
 - Se sim**, preenchemos os campos da tela com os dados do usuário encontrado e habilitamos os botões de editar e excluir, desabilitamos o botão de Incluir e avisamos o sucesso.

```
protected void btnBuscar_Click(object sender, EventArgs e) {
    Usuario usuario = (Usuario)Session["usuario"];
    Usuario sel = null;
    lblMensagem.Text = String.Empty;
    btnExcluir.Visible = btnAtualizar.Visible = false;
    btnIncluir.Visible = true;
    limpa();
    if (txtNomeBusca.Text.Trim() == String.Empty) {
        lblMensagem.Text = "Digite um texto para busca (nome, login ou sequencial)"; return;
    }
    using (Conexao con = new Conexao(usuario)) {
        con.open();
        Int16 id;
        if (Int16.TryParse(txtNomeBusca.Text, out id)) { // Será que o usuário digitou um número?
            sel = Usuario.Busca(id, con);
            if (sel == null) {
                lblMensagem.Text = "Nenhum usuário foi encontrado com o sequencial " + id + "!"; return;
            }
        }
        else { // Não é número, então é substring
            List<Usuario> lista = Usuario.ListaUsuarios(txtNomeBusca.Text, con); // Traz uma lista de usuários
            if (lista.Count == 0) { // lista vazia! não encontrei ninguém!
                lblMensagem.Text = "Nenhum usuário foi encontrado com o nome/login informado!"; return;
            }
            if (lista.Count > 1) { // Mais que um na lista? não vale.. preciso achar só um
                lblMensagem.Text = "Mais de um usuário foi encontrado com o nome/login informado!";
                return;
            }
            sel = lista[0];
        }
        Session["usuarioProcurado"] = sel; // Achei um usuário!
        txtNome.Text = sel.nome; // Colocamos os campos do usuário encontrado na tela
        txtLogin.Text = sel.login;
        txtCPF.Text = sel.cpf;
        if (sel.perfil == "ADM") { // se for ADM, arrumamos os radiobuttons
            rbAdm.Checked = true; rbUsu.Checked = false;
        } else { // se não é USU, arrumamos os radiobuttons
            rbAdm.Checked = false; rbUsu.Checked = true;
        }
        btnExcluir.Visible = btnAtualizar.Visible = true;
        btnIncluir.Visible = false;
        lblMensagem.Text = "Usuário encontrado";
    }
}
```

Veja bem: Todo o processo foi especificado nos requisitos do Caso de Uso de cadastro de usuários!



7

Editando um registro

- Como pré requisito, devemos ter o registro já disponível na sessão;
- O mesmo método **validarCampos()** utilizado no cadastro ne novos registros é chamado aqui também para evitar alterações inconsistentes.
- Depois de aberta a conexão, os campos editados da tela são atribuídos as propriedades do objeto e o método **atualiza()** é chamado para finalizar o processo.
- A tabela com os usuários já cadastrados é carregada para auxiliar na administração dos usuários.

```
protected void btnAtualizar_Click(object sender, EventArgs e) {
    Usuario usu = (Usuario)Session["usuarioProcurado"];
    if (usu == null) {
        lblMensagem.Text = "Erro inesperado. Usuário para editar ainda não selecionado!";
        return;
    }
    if (!validarCampos()) return;
    txtCPF.Text = txtCPF.Text.Replace(".", "").Replace("/", "");
    Usuario usuario = (Usuario)Session["usuario"];
    try {
        using (Conexao con = new Conexao(usuario)) {
            con.open();
            usu.nome = txtNome.Text.Trim();
            usu.login = txtLogin.Text.Trim();
            usu.cpf = txtCPF.Text.Trim();
            usu.perfil = rbAdm.Checked ? "ADM" : "USU";
            usu.atualiza(con);
            limpa();
            relatorioUsuarios.Text = usuario.montaTabela((String)Session["empresa"], con);
            lblMensagem.Text = "Usuário atualizado com sucesso!";
        }
    }
    catch (Exception e1) {
        if (usuario.perfil == "ADM") {
            lblMensagem.Text = e1.Message;
        }
        else {
            lblMensagem.Text = "Erro excluindo usuario";
        }
    }
}
```

Cadastro de Usuários

Buscar usuário por Nome>Login ou Sequencial

Nome* Perfil* ADM ☒ USU ☐

Seq.	Nome	Login	Senha	Perfil	CPF
0012	Ana Maria Rosário	ana.maria	21987654321	USU	21987654321
0007	Carlos Nascimento	carlos.nascimento	34567890123	ADM	00000000000
0008	Jose Alberto	jose.alberto	Jã tocada	ADM	32368280778
0009	Maria Aparecida	maria.aparecida	11111111111	ADM	11111111111
0011	Paulo Da Silva	paulo.silva	12345678912	USU	12345678912
0013	Silvia Silva	silvia.silva	21098765432	ADM	21098765432

Usuário encontrado



8

Excluindo um registro

- Como pré requisito, devemos ter o registro já disponível na sessão;
- Basicamente o que é feito é chamar o método de excluir e, em seguida, mostrar a tabela com os registros cadastrados.

```
public int exclui(Conexao con)
{
    try
    {
        String sql = String.Concat("DELETE FROM ", nomeTabela,
                                   " WHERE idUsuario=" , idUsuario);
        return Conexao.executaQuery(con, sql, nomeTabela, null);
    }
    catch (Exception)
    {
        throw;
    }
}
```

```
protected void btExcluir_Click(object sender, EventArgs e)
{
    Usuario usuario = (Usuario)Session["usuario"];
    try
    {
        Usuario usu = (Usuario)Session["usuarioProcurado"];
        if (usu == null)
        {
            lbMensagem.Text = "Erro inesperado. Usuário para excluir ainda não selecionado!";
            return;
        }
        using (Conexao con = new Conexao(usuario))
        {
            con.open();
            usu.exclui(con);
            relatorioUsuarios.Text = usuario.montaTabela((String)Session["empresa"], con);
        }
        limpa();
        lbMensagem.Text = "Usuário excluído com sucesso!";
    }
    catch (Exception e1)
    {
        if (usuario.perfil == "ADM")
        {
            lbMensagem.Text = e1.Message;
        }
        else
        {
            lbMensagem.Text = "Erro excluindo usuário";
        }
    }
}
```