

Escola Técnica Estadual “São Paulo” - ETESP

Linguagem C# - Visual Studio 2019

10 - Manipulação de “string”

Podemos afirmar que um bom “sistema de informações” é aquele que seja capaz de garantir uma boa qualidade na entrega das informações, por exemplo:

Uma grande empresa solicita o desenvolvimento de um aplicativo para efetuar o cadastramento de seus clientes. O Analista de Sistemas responsável pelo projeto do aplicativo, após levantamento dos requisitos, identificou como informações importantes para serem cadastradas: CPF, Nome, Sexo, Data_Nascimento, Endereço, CEP, Cidade, Estado e E-mail.

O que se espera é que os dados fornecidos pelos usuários e que serão gravados, sejam confiáveis...

Alguns poderão ter “parametros”, terem um valor mínimo e máximo, por exemplo o ano de nascimento não poderá ser maior que o ano atual.

O CPF possui um dígito verificador. A finalidade deste dígito é justamente para “validar” o CPF informado. Existe uma “fórmula” para o cálculo deste dígito (o site http://www.macoratti.net/alg_cpf.htm apresenta um algoritmo para calcular este dígito). Caso o usuário digite um CPF com um dígito verificador incorreto, o aplicativo poderá exibir uma mensagem de alerta ao usuário;

Os demais campos: sexo, cidade, estado e a data de nascimento também poderão ter regras de validação que poderão ser definidas;

Alguns serviços, disponíveis na Internet, poderão validar o endereço, através de uma consulta feita pelo CEP. O site dos “Correios” fornece esta consulta: (<https://buscacepinter.correios.com.br/>).

E a validação do “nome do cliente”? Quais regras poderemos utilizar: O nome não poderá estar “vazio” ou conter mais de “50 caracteres”? Só isto não garantirá uma boa qualidade... E se o usuário preencher o nome tudo com a barra de espaços?? Fornecer um nome somente com “vogais”, etc, etc, etc...

Quais regras poderemos utilizar para “validar o e-mail”? Verificar se foram digitados o nome e o domínio? Solicitar a confirmação do e-mail, através de nova digitação? Gerar um código de verificação e encaminhar para o e-mail informado e solicitar a digitação?

Para resolvermos parte dos problemas apresentados é que estudaremos este assunto e também os próximos, pois sempre teremos novos desafios para superar.

Vamos começar...

FUNÇÃO	FINALIDADE	EXEMPLO
ToUpper	Converte todo o texto para “maiúsculo” (caixa alta)	string nome = “fulano de tao”; nome = nome.ToUpper(); resultado: Irá converter o texto “fulano de tao” para “FULANO DE TAO”.
ToLower	Converte todo o texto para “minúsculo” (caixa baixa)	string nome = “FULANO DE TAO”; nome = nome.ToLower(); Resultado: Irá converter o texto “FULANO DE TAO” para “fulano de tao”
Length	Armazena a quantidade	string nome = “fulano de tao”;

Prof. Roberto de Castro

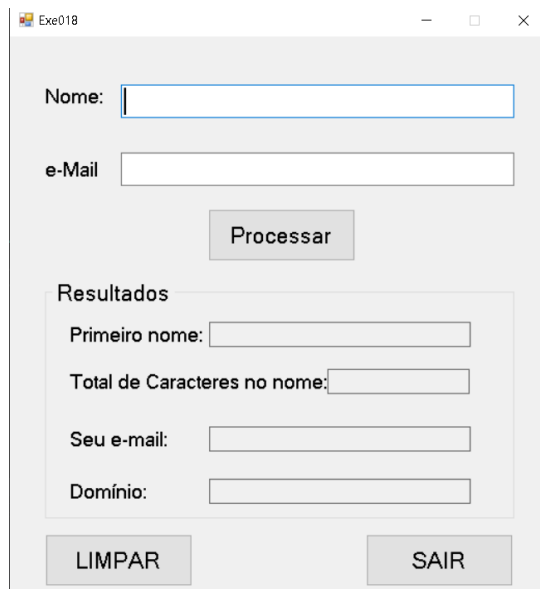
	de caracteres existentes em uma string	<pre>int qtde = 0; qtde = nome.Length;</pre> <p>Resultado → qtde será igual a 13 (total de caracteres existentes na string "nome", incluindo os espaços).</p>
Trim	Remove todos os espaços em branco existentes no início e no final de um texto	<pre>string nome = " fulano de tao "; nome = nome.Trim();</pre> <p>Resultado → Irá remover todos os espaços antes de "fulano" e após "tao". O resultado será: "fulano de tao";</p>
Replace	Substitui um caractere ou texto dentro de uma string.	<pre>string nome = "fulano de tao"; string nomeAlterado = ""; nomeAlterado = nome.Replace("tao", "tchau");</pre> <p>Resultado → nomeAlterado = fulano de tchau</p> <p>OBS: Se não existir, nenhuma alteração será realizada.</p>
IndexOf	Informa a posição de um caractere dentro de um texto	<pre>string email = "fulano@tao.com.br" int posição = email.IndexOf("@");</pre> <p>Resultado → 6 (a primeira posição começa em Zero). Se o caractere não existir, devolve o resultado -1.</p>
Substring	"Retorna" um "pedaço" de uma string.	<pre>string nomeCompleto = "Fulano de Tao"; string primeiroNome = nomeCompleto.Substring(0, 6);</pre> <p>Resultado → Fulano (o número zero indica a posição inicial e o número seis a quantidade de caracteres).</p>
StringBuilder	StringBuilder pode ser usada quando você deseja fazer "grandes" alterações em uma string (incluir, substituir ou excluir partes).	<p><u>Incluir/adicionar caracteres dentro de uma string:</u></p> <pre>StringBuilder meuTexto = new StringBuilder("Olá Mundo"); meuTexto.Insert(4, "Todo");</pre> <p>Resultado → Insere a palavra "todo" a partir da posição 4. A string meuTexto será igual a "Olá Todo Mundo."</p> <p><u>Excluir parte de um texto dentro de uma string:</u></p> <pre>StringBuilder meuTexto = new StringBuilder("Olá Mundo"); meuTexto.Remove(4,5);</pre> <p>Resultado → Exclui 5 caracteres a partir da posição 4 (OBS: A posição inicial é sempre zero). A string meuTexto será igual a "Olá".</p> <p><u>Substituir parte de um texto de uma string:</u></p> <pre>StringBuilder meuTexto = new StringBuilder("Olá</pre>

Prof. Roberto de Castro

		<code>Mundo!");</code> <code>meuTexto.Replace('!', '?');</code> Resultado → Substitui o caractere “!” pelo “?”.
--	--	---

Exercício de fixação → Exe018

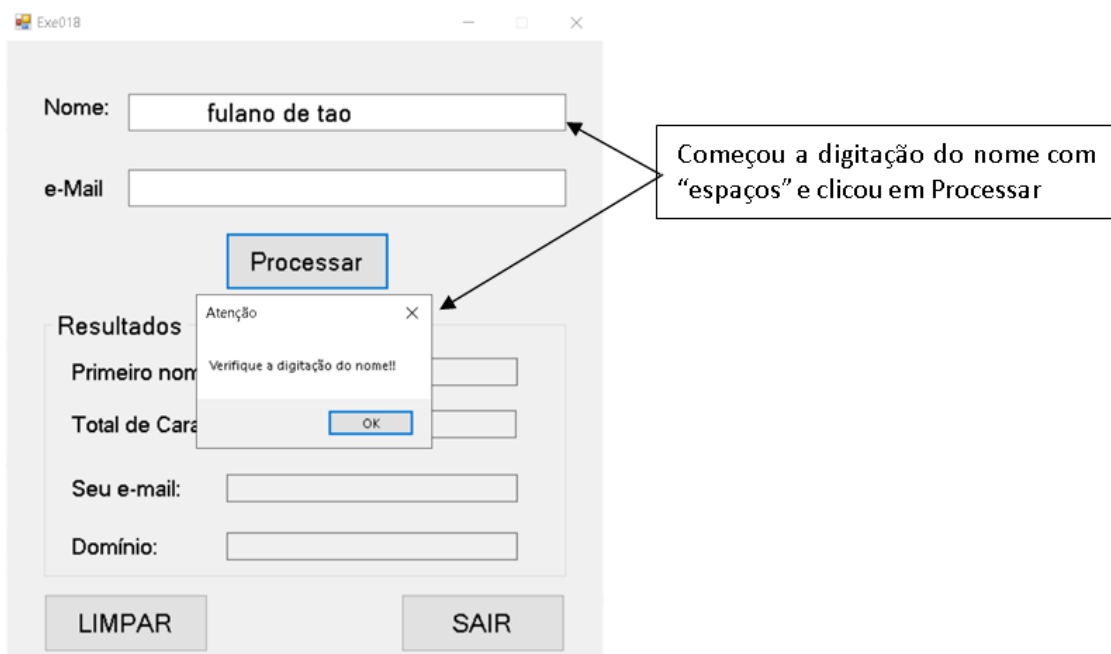
INTERFACE



Considerações:

- O nome é obrigatório e não pode iniciar com “espaços” e deverá ter entre 10 e 60 caracteres;
- O endereço de e-mail é obrigatório

Observe algumas situações:



Prof. Roberto de Castro

Nome: fulano de tao

e-Mail: fulano

Processar

Atenção

Verifique o e-mail!!

OK

Resultados

Primeiro nome:

Total de Caracteres no nome: 13

Seu e-mail:

Domínio:

LIMPAR SAIR

Nome: FULANO DE TAO

e-Mail: fulano@paulasouza.com.br

Processar

Resultados

Primeiro nome: FULANO

Total de Caracteres no nome: 13

Seu e-mail: fulano

Domínio: paulasouza.com.br

LIMPAR SAIR

SOLUÇÃO

```
using System;
using System.Windows.Forms;

namespace Exe018
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void BtnProcessar_Click(object sender, EventArgs e)
```

```
{
    string nome = TxtNome.Text;
    string testeNome = nome.Trim();
    int qtdeCaracteres = testeNome.Length;

    if (nome != testeNome)
    {
        MessageBox.Show("Verifique a digitação do nome!!", "Atenção");
        TxtNome.SelectionStart = 0;
        TxtNome.SelectionLength = TxtNome.Text.Length;
        TxtNome.Focus();
    }
    else if (qtdeCaracteres < 10 || qtdeCaracteres > 60)
    {
        MessageBox.Show("Tamanho do nome está fora do padrão!!", "Atenção");
        TxtNome.SelectionStart = 0;
        TxtNome.SelectionLength = TxtNome.Text.Length;
        TxtNome.Focus();
    }
    else
    {
        int finalPrimeiroNome = testeNome.IndexOf(" ");
        LblNome.Text = nome.Substring(0, finalPrimeiroNome);
        LblTotCaractere.Text = qtdeCaracteres.ToString();

        string eMail = TxtEmail.Text;
        int finalNomeEmail = eMail.IndexOf("@");
        int totTamanhoEmail = eMail.Length;
        int totTamanhoDominio = totTamanhoEmail - finalNomeEmail;

        if (finalNomeEmail < 1) //indica que o "@" não foi inserido
        {
            MessageBox.Show("Verifique o e-mail!!", "Atenção");
            TxtEmail.SelectionStart = 0;
            TxtEmail.SelectionLength = TxtEmail.Text.Length;
            TxtEmail.Focus();
        }
        else
        {
            LblEmail.Text = eMail.Substring(0, finalNomeEmail);
            LblDominio.Text = eMail.Substring(finalNomeEmail + 1, totTamanhoDominio-1);
        }
    }
}

private void BtnSair_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Proposta de melhorias: A solução apresentada poderá gerar erros caso o usuário digite “apenas” o primeiro nome.