

# INTRODUÇÃO À PROGRAMAÇÃO COM C# [DESAFIOS]

COPYRIGHT © RAFAEL PADILLA

# REGULAMENTO

- É necessário disponibilizar o código (Projeto Visual Studio);
- Não enviar binários (.exe .dll)
- Nos desafios com Windows Forms, utilize os componentes que facilitem a usabilidade – Use o bom senso;
- O código deverá estar bem documentado / comentado;
- Siga as boas práticas de desenvolvimento vistas em aula;

## Os desafios farão parte das notas da disciplina!

Os pontos avaliados em cada projeto serão:

- **Projeto funcional com tratamento de exceções;**
- **Projeto consistente com as instruções;**
- **Código comentado;**
- **Nomenclatura de variáveis, métodos e classes seguindo o padrão C#;**

# DESAFIO # 1

## (SUED)

# DESAFIO #1 (SUED)

No final dos anos 90, um programa de grande sucesso surgiu com o nome de **Einstein**. Sua funcionalidade também foi copiada em soluções similares, a mais famosa foi apresentada com o nome de **SUED**.

Este programa possuía uma interface bastante primitiva: a famosa tela preta do MS-DOS. Apesar da interface pobre, os usuários divertiam-se mostrando para seus amigos que o Einstein era capaz de responder quaisquer perguntas digitadas pelo dono do computador. A parte engraçada era que para o Einstein responder corretamente, era preciso fazer elogios ao programa.

Neste exercício você deverá criar este programa (utilizando C#) para responder as perguntas (e respostas) digitadas pelo usuário. Siga as instruções a seguir:

i) Você deverá definir uma 'frase de elogio' para seu programa.

(Ex: "Máquina dotada de inteligência, capaz de responder todas as minhas perguntas, "

"Programa de grande inteligência, desenvolvido por mentes brilhantes, "

"Minha máquina linda, inteligente, graciosa, superdotada, ")

# DESAFIO #1 (SUED)

**ii)** Ao fazer a pergunta, se o primeiro caractere inserido for um ";", cada próximo caractere digitado deverá mostrar os caracteres da 'frase de elogio' em ordem. Os caracteres que o usuário digitar após o ";" formarão a resposta da pergunta. Neste momento a 'frase de elogio' estará sendo exibida conforme as teclas forem sendo pressionadas. Após digitada a resposta, o usuário deverá apertar o "Enter" e, a partir deste momento, todos caracteres inseridos serão mostrados na tela conforme são digitados.

Caso o primeiro caractere inserido não for um ";", todos os caracteres inseridos serão mostrados na tela conforme são digitados e a resposta não estará sendo capturada

**iii)** Após digitado o "Enter", você deverá mostrar uma 'animação' simulando que o programa está 'pensando' na resposta. (Ex: "Pensando @&%!@#!... ")

**iv)** Depois de 'pensar', a resposta deverá ser mostrada para o usuário. Caso a resposta não tenha sido informada, uma desculpa deverá ser mostrada (Ex: "Esta eu só respondo para meu mestre" ou "Estou cansado agora, não posso te responder")

# DESAFIO #2

## (LIMITES DE VARIÁVEIS)

# DESAFIO #2 (LIMITES DE VARIÁVEIS)

Crie um programa em C# mostrando em uma Windows Form os limites máximos e mínimos dos seguintes tipos de variáveis:

**byte, sbyte, short, ushort, int, uint, long, ulong, float, double e decimal**

**Obs:** Para obter os limites máximos e mínimos, use as constantes de cada classe.

# DESAFIO #3

## (INVERTER CARACTERES)



# DESAFIO #3 (INVERTER CARACTERES)

Crie um programa (Windows Form) com C# capaz de inverter os caracteres de uma frase informada pelo usuário. Desconsidere os caracteres ".", ";", "!", "?", ",", e os espaços " ". A frase de saída deverá ser totalmente em letras minúsculas.

Ex:

Frase de entrada: "Olá, tudo bem com você?"

Frase de saída: "êcovmocmebodutálo"

# DESAFIO #4

## (FREQUÊNCIA PALAVRAS)

# DESAFIO #4 (FREQUÊNCIA PALAVRAS)

Crie um programa (Windows Form) com C# capaz de informar a frequência que cada palavra aparece em um texto de forma ordenada por quantidade de ocorrências.

# DESAFIO #5

## (API PIADAS)

# DESAFIO #5 (API PIADAS)

Neste desafio, você deverá criar um **Crawler** de uma página HTML, capaz de mostrar em uma `List<T>`, os objetos capturados a partir do código HTML.

Para este desafio, não é necessário criar um projeto Windows Forms. A interface com o usuário é opcional.

É necessário obter as 15 primeiras piadas do site

<http://www.osvigaristas.com.br/piadas/>  
em forma de objetos (`List<Piada>`) da seguinte classe:

```
public class Piada
{
    public string Nome { get; set; }
    public string Url { get; set; }
    public string Corpo { get; set; }
    public string Categoria { get; set; }
    public DateTime DataPublicacao { get; set; }
    public int VotosPositivos { get; set; }
    public int VotosNegativos { get; set; }
    public int VotosMedios { get; set; }
}
```

# DESAFIO #6

## (XML DE CARROS)

# DESAFIO #6 (XML DE CARROS)

Neste desafio, você deverá criar um pequeno sistema de uma loja de carros. Com este sistema o usuário poderá:

1) Selecionar um arquivo XML e *deserializar* as informações contidas neste arquivo

1.1) Este arquivo deverá ter a seguinte estrutura:

```
<?xml version="1.0" encoding="utf-8"?>
<ElementoRaiz>
  <DataCriacao>15 de maio de 2014</DataCriacao>
  <Carros>
    <Carro>
      <Codigo>0001</Codigo>
      <Marca>Nissan</Marca>
      <Modelo>Sentra</Modelo>
      <Ano>2015</Ano>
      <Preço>50000</Preço>
    </Carro>
    <Carro>
      <Codigo>0002</Codigo>
      <Marca>Toyota</Marca>
      <Modelo>Corolla</Modelo>
      <Ano>2014</Ano>
      <Preço>58000</Preço>
    </Carro>
    <Carro>
      <Codigo>0003</Codigo>
      <Marca>Chevrolet</Marca>
      <Modelo>Chevette</Modelo>
      <Ano>1993</Ano>
      <Preço>12000</Preço>
    </Carro>
  </Carros>
</ElementoRaiz>
```

Lista de carros  
(mín 10 cadastros)

1.2) Este arquivo deverá ter no mínimo 10 carros cadastrados.

# DESAFIO #6 (XML DE CARROS)

2) Visualizar em uma UI (Windows Form) as informações contidas no XML

2.1) Esta UI deverá ter 4 ComboBoxes contendo todas as Marcas, Modelos, Anos e Preços disponíveis no XML. Não poderão ter marcas, modelos, anos e preços repetidos.

2.2) As ComboBoxes serão filtros para que as informações do XML sejam apresentadas em uma ListView.

2.3) Um item obrigatório em todas as ComboBoxes é uma opção que quando selecionada traga todos os itens desta categoria sem filtros. Esta opção pode ser “(Todas Marcas)”, “(Todos Modelos)”, “(Todos Anos)” e “(Todos Preços)”.

3) Utilize expressões Lambda ou LINQ para realizar a filtragem dos itens.

4) Utilize o exemplo de UI a seguir



# DESAFIO #6 (XML DE CARROS)

Auto Fácil

Arquivo:

Marca:  Ano:

Modelo:  Preço:

Id	Marca	Modelo	Ano	Preço
----	-------	--------	-----	-------

Diagram labels:

- Caminho do XML (points to the Arquivo field)
- Botão para escolha do XML (points to the ... button)
- ComboBoxes de filtragem (points to the Marca, Ano, Modelo, and Preço fields)
- ListView mostrando os resultados (points to the table)

Exemplo de UI Windows Form