



# Criação de Aplicação web

DOS FUNDAMENTOS DA LINGUAGEM C# ATÉ A HOSPEDAGEM

# Sergio Di Fiore

[sergio.fiore@fatec.sp.gov.br](mailto:sergio.fiore@fatec.sp.gov.br)

[sergiodifiore@gmail.com](mailto:sergiodifiore@gmail.com)

<https://www.difiore.com.br/>

<https://github.com/sdifiore/>

[https://youtube.com/c/SergioDiFiore\\_0/](https://youtube.com/c/SergioDiFiore_0/)

<https://www.linkedin.com/in/sergiodifiore/>

<https://dev.to/sdifiore/>

Twitter: @difiores1



# Programação inicial do curso (1)

- ▶ Estudo de diversos ambientes de desenvolvimento e qual escolher;
- ▶ O que é a linguagem C#;
- ▶ O que é .NET;
- ▶ O que é OOP;
- ▶ Programação básica;
- ▶ Desvios e testes condicionais;
- ▶ Vetores e matrizes;
- ▶ Programação orientada a objeto;
- ▶ Listas e coleções;
- ▶ Git e GitHub;
- ▶ MVC;
- ▶ Hospedagem;
- ▶ Criação do projeto de uma loja de álbuns online;
- ▶ HTML/CSS básicos;
- ▶ Bootstrap básico;
- ▶ Desenvolvimento de Views;
- ▶ Views parciais;
- ▶ Roteamento;
- ▶ Bancos de Dados;

# Programação inicial do curso (2)

- ▶ SQL Server;
- ▶ ORM - (Object-relational mapping);
- ▶ Autenticação;
- ▶ Autenticação com 2º fator;
- ▶ Autenticação através do Facebook;
- ▶ Criando formulários sofisticados;
- ▶ Validação;
- ▶ Clear Programming;
- ▶ Tabelas;
- ▶ CRUD;
- ▶ Javascript;
- ▶ Tabelas;
- ▶ Deploy da aplicação pronta;
- ▶ Manutenção;

# Vale a pena conhecer:

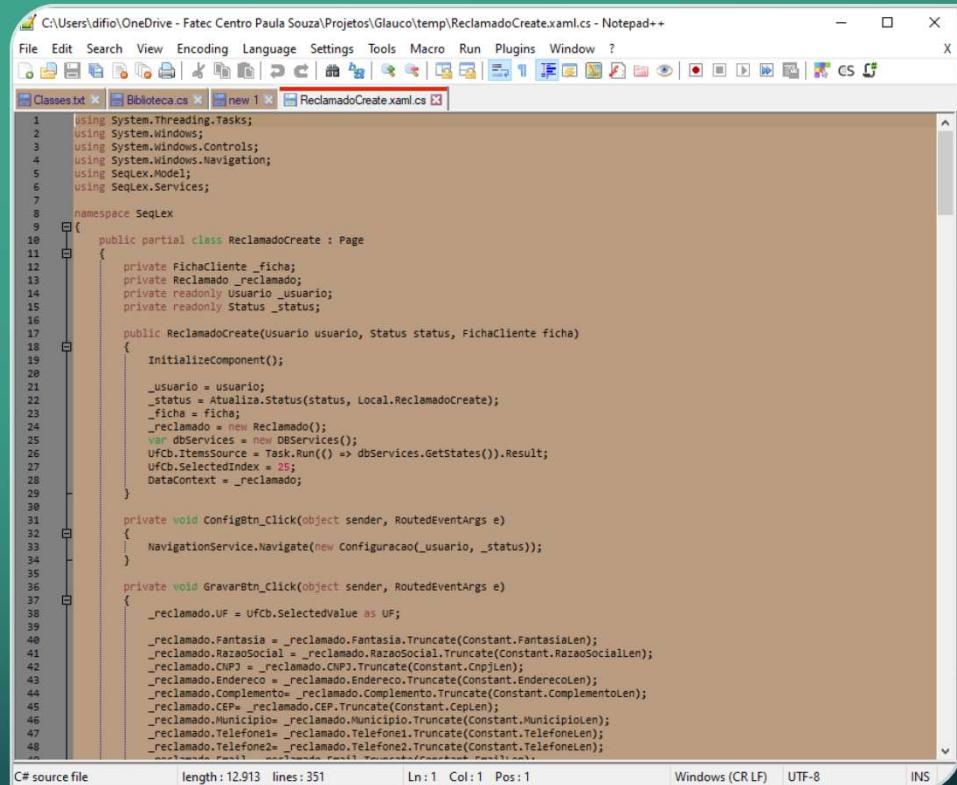
- ▶ <https://channel9.msdn.com/>
- ▶ <https://codepen.io/>
- ▶ <https://docs.microsoft.com/>
- ▶ <https://dotnet.microsoft.com/>
- ▶ **<https://dotnetfiddle.net/>**
- ▶ <https://duckduckgo.com/>
- ▶ <https://www.codegrepper.com/>
- ▶ **<https://stackoverflow.com/>**
- ▶ <https://wakatime.com/>
- ▶ <https://www.codeproject.com/>
- ▶ <https://www.entityframeworktutorial.net/>
- ▶ <https://www.w3schools.com/>

I:

Ferramentas de  
desenvolvimento

# Ambientes de desenvolvimento: Editores

- ▶ Notepad++
- ▶ Atom
- ▶ VI
- ▶ Brackets
- ▶ Brackets
- ▶ Etc.



The screenshot shows a Notepad++ window with the title bar "C:\Users\difio\OneDrive - Fatec Centro Paula Souza\Projetos\Glauco\temp\ReclamadoCreate.xaml.cs - Notepad++". The window contains C# code for a Windows application. The code defines a partial class ReclamadoCreate that inherits from Page. It includes properties for FichaCliente, Reclamado, Usuario, and Status. The constructor initializes these properties and sets up event handlers for Configbtn\_Click and GravarBtn\_Click. The code also includes several Truncate statements to handle string lengths for Fantasia, RazaoSocial, CNPJ, Endereco, Complemento, CEP, Municipio, and Telefone fields.

```
1  using System.Threading.Tasks;
2  using System.Windows;
3  using System.Windows.Controls;
4  using System.Windows.Navigation;
5  using SeqLex.Model;
6  using SeqLex.Services;
7
8  namespace SeqLex
9  {
10     public partial class ReclamadoCreate : Page
11     {
12         private FichaCliente _ficha;
13         private Reclamado _reclamado;
14         private readonly Usuario _usuario;
15         private readonly Status _status;
16
17         public ReclamadoCreate(Usuario usuario, Status status, FichaCliente ficha)
18         {
19             InitializeComponent();
20
21             _usuario = usuario;
22             _status = Atualiza.Status(status, Local.ReclamadoCreate);
23             _ficha = ficha;
24             _reclamado = new Reclamado();
25             var dbServices = new DBServices();
26             UfcB.ItemsSource = Task.Run(() => dbServices.GetStates()).Result;
27             UfcB.SelectedIndex = 25;
28             DataContext = _reclamado;
29         }
30
31         private void Configbtn_Click(object sender, RoutedEventArgs e)
32         {
33             NavigationService.Navigate(new Configuracao(_usuario, _status));
34         }
35
36         private void GravarBtn_Click(object sender, RoutedEventArgs e)
37         {
38             _reclamado.UF = UfcB.SelectedValue as UF;
39
40             _reclamado.Fantasia = _reclamado.Fantasia.Truncate(Constant.FantasiaLen);
41             _reclamado.RazaoSocial = _reclamado.RazaoSocial.Truncate(Constant.RazaoSocialLen);
42             _reclamado.CNPJ = _reclamado.CNPJ.Truncate(Constant.CnpjLen);
43             _reclamado.Endereco = _reclamado.Endereco.Truncate(Constant.EnderecoLen);
44             _reclamado.Complemento = _reclamado.Complemento.Truncate(Constant.ComplementoLen);
45             _reclamado.CEP = _reclamado.CEP.Truncate(Constant.CepLen);
46             _reclamado.Municipio = _reclamado.Municipio.Truncate(Constant.MunicipioLen);
47             _reclamado.Telefone1 = _reclamado.Telefone1.Truncate(Constant.TelefoneLen);
48             _reclamado.Telefone2 = _reclamado.Telefone2.Truncate(Constant.TelefoneLen);
49         }
50     }
51 }
```

C# source file length: 12.913 lines: 351 Ln:1 Col:1 Pos:1 Windows (CR LF) UTF-8 INS

# Notepad++

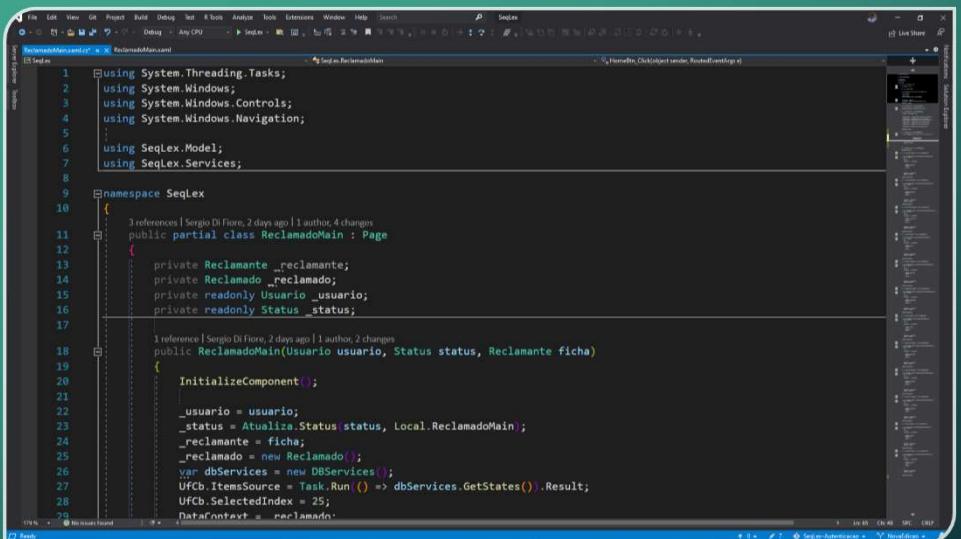
## Extensões recomendáveis:

- ▶ BetterMultiSelectio
- ▶ Bookmarks@Dood
- ▶ Code alignment
- ▶ Compare
- ▶ CS-Script
- ▶ Explorer
- ▶ HTML Tag
- ▶ Language Help
- ▶ Merge files in one
- ▶ Coverter
- ▶ NppFavorites
- ▶ NppGist
- ▶ JSOn Viewer
- ▶ NppSaveAsAdmin
- ▶ Nppp Xml TreView
- ▶ SecurePad
- ▶ Take Notes
- ▶ Wakatime

# Ambientes de desenvolvimento: IDE

# ► Integrated Development Environment

## ► Ambiente de Desenvolvimento Integrado:



# Ambientes de desenvolvimento: IDE

- ▶ Destaca sintaxe
- ▶ Autocompleta o código digitado
- ▶ Refatoramento
- ▶ Controle de versão
- ▶ Depuração
- ▶ Pesquisa de código
- ▶ Suporte a múltiplas linguagens

# VS Code – O intermediário

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** network.py - DeepLearningPython - Visual Studio Code.
- Left Sidebar (Explorer):** Shows files: network.py, Método SGD.py, and Untitled-2.ipynb.
- Editor Area:** Displays the content of the network.py file. The code implements a stochastic gradient descent (SGD) algorithm for training a neural network. It includes a detailed docstring explaining the parameters and the training process. The code uses Python's list comprehensions and the random.shuffle function.
- Bottom Status Bar:** Shows the current branch (master), Python version (3.8.10 64-bit), and other settings like line 80, column 71, and encoding (UTF-8).

```
gradient descent. The ``training_data`` is a list of tuples
``(x, y)`` representing the training inputs and the desired
outputs. The other non-optional parameters are
self-explanatory. If ``test_data`` is provided then the
network will be evaluated against the test data after each
epoch, and partial progress printed out. This is useful for
tracking progress, but slows things down substantially."""

training_data = list(training_data)
n = len(training_data)

if test_data:
    test_data = list(test_data)
    n_test = len(test_data)

for j in range(epochs):
    random.shuffle(training_data)
    mini_batches = [
        training_data[k:k+mini_batch_size]
        for k in range(0, n, mini_batch_size)]
    for mini_batch in mini_batches:
        self.update_mini_batch(mini_batch, eta)
    if test_data:
        print("Epoch {} : {} / {}".format(j, self.evaluate(test_data), n_test))
    else:
        print("Epoch {} complete".format(j))

def update_mini_batch(self, mini_batch, eta):
    """Update the network's weights and biases by applying
    gradient descent using backpropagation to a single mini batch.
    The ``mini_batch`` is a list of tuples ``(x, y)``, and ``eta``
    is the learning rate."""
    nabla_b = [np.zeros(b.shape) for b in self.biases]
    nabla_w = [np.zeros(w.shape) for w in self.weights]
```



II:

# Preparação do ambiente

UNIFICAÇÃO PARA O CURSO

# Alterando o Prompt do PowerShell (1)

- ▶ Abra um terminal PowerShell como administrador;
- ▶ Digite: **new-item -path \$profile -itemtype file -force**
- ▶ Abra o Notepad para editar esse arquivo: **Notepad \$profile**

# Alterando o Prompt do PowerShell (2)

- Adicione o código abaixo (em 4 linhas como está):

```
function prompt
{
    $p = Split-Path -leaf -path (Get-Location)
    "$p> "
}
```

# Alterando o Prompt do PowerShell (3)

- ▶ Grave;
- ▶ Execute o comando:
  - ▶ **Set-ExecutionPolicy RemoteSigned -Scope CurrentUser**
- ▶ O PowerShell agora só irá mostrar o último nível de diretório!

III:

# Visual Studio Code

VS CODE

# Visual Studio Code (1)

## Bibliografia:

Del Sole, A. (2019). Visual Studio Code Distilled Evolved Code Editing for Windows, macOS, and Linux. Apress.

Johnson, B. (2019). Visual Studio Code End-To-End Editing And Debugging Tools For Web Developers. John Wiley & Sons, Inc.

# Visual Studio Code (2)

Extensões mínimas para o curso:

- ▶ .NET Extension Pack;
- ▶ C# for Visual Studio Code;
- ▶ Debugger for Microsoft Edge;
- ▶ HTML CSS Support;
- ▶ JavaScript (ES6) code snippets;
- ▶ Live Share;
- ▶ Razor+;
- ▶ Visual Studio IntelliCode.

# Visual Studio Code (3)

## Extensões recomendáveis (1/4):

- ▶ .NET Interactive Notebooks
- ▶ .NET Interactive Notebooks;
- ▶ Auto Close Tag
- ▶ Auto Rename Tag;
- ▶ Auto-Using for C#
- ▶ Better Comments
- ▶ Bookmarks
- ▶ Bracket Pair Colorizer
- ▶ C# Extensions
- ▶ C# Helper
- ▶ C# Namespace Autocompletion
- ▶ C# Snippets
- ▶ C# XML Documentation Comments
- ▶ C# for Visual StudioCode
- ▶ C/C++ GNU Global
- ▶ CSS Peek;
- ▶ CSharp to PlantUML Studio Code;
- ▶ Checkpoints
- ▶ Code Runner

# Visual Studio Code (4)

## Extensões recomendáveis (2/4):

- ▶ CodeStream: GitHub, GitLab,...
- ▶ Contentful ModelsCreator Csharp
- ▶ Debugger for Chrome
- ▶ Docker
- ▶ EFCore Snippets
- ▶ EditorConfig for VS Code
- ▶ Explorer Exclude
- ▶ Exuberant CTags
- ▶ FreePascal Toolkit
- ▶ GDB Debugger - Beyond
- ▶ GistPad
- ▶ Git Graph
- ▶ Git History
- ▶ GitHub Pull Requests and Issues
- ▶ GitLens — Git supercharged
- ▶ Hex Editor
- ▶ Hungry Delete
- ▶ Icon Fonts
- ▶ Jupyter

# Visual Studio Code (5)

## Extensões recomendáveis (3/4):

- ▶ Kite
- ▶ Kubernetes
- ▶ Live Preview
- ▶ Live Share Audio
- ▶ Live Share Whiteboard
- ▶ Local History
- ▶ MSBuild project tools
- ▶ Native Debug
- ▶ OmniPascal
- ▶ Pascal magic
- ▶ Path Intellisense
- ▶ Peacock
- ▶ Prettier - Code formatter
- ▶ Project Manager
- ▶ Pylance
- ▶ Regex Previewer
- ▶ Remote - WSL
- ▶ Rich Code Navigation for GitHub Browser
- ▶ Search Editor: Apply Changes

# Visual Studio Code (6)

## Extensões recomendáveis (4/4):

- ▶ Super Sharp (C# extensions)
- ▶ TODO Highlight
- ▶ Text Marker (Highlighter)
- ▶ Text Pastry
- ▶ Visual Studio Keymap
- ▶ Wakatime
- ▶ Word Count
- ▶ XML to JSON
- ▶ Xml Complete
- ▶ change-case
- ▶ csharp-new-class
- ▶ ms-edge-launcher
- ▶ open-in-browser
- ▶ open-in-browser;
- ▶ µTask

# IV: HTML



# Html básico (1)

## Bibliografia:

- Carey, P. (2013). New Perspectives on HTML, CSS, and Dynamic HTML, 5th Edition. Course Technology, Cengage Learning.
- Duckett, J. (2010). Begining HTML, XHTML, CSS, and JavaScript®. Willey Publishing Inc.
- Mueller, J. P. (2016). Security for Web Developers. O'Reilly Media, Inc.
- Myers, M. (2015). A Smarter Way To Learn HTML & CSS Learn it faster. Remember it longer. Mark Myers.
- Powell, T. A. (2010). HTML & CSS:. McGraw-Hill. Powell, T. A. (2010). HTML & CSS: The Complete Reference, Fifth Edition. McGraw-Hill.
- Robbins, J. N. (2018). Learning Web Design, A Beginner'S Guide To HTML, CSS, Javascript, a Web Graphics. O'Reilly Media, Inc.
- Ribeiro, H. (s.d.). Fonte: Site pessoal de Hugo Ribeiro: [hugoribeiro.com.br](http://hugoribeiro.com.br)
- Robbins, J. N. (2018). Learning Web Design, A Beginner'S Guide To HTML, CSS, Javascript, a Web Graphics. O'Reilly Media, Inc.
- Santos, M. d. (s.d.). Fonte: Pensando na web: [pensandonaweb.com.br](http://pensandonaweb.com.br)
- Zea, R. (2015). Mastering Responsive Web Design with HTML5 and CSS3. Packt Publishing.

## Html básico (2)

- ▶ Crie/Escolha um diretório de trabalho com Windows Explorer;
- ▶ Neste crie um diretório WebExemplos;
- ▶ Navegue ao mesmo;
- ▶ Com um clique direito selecione abrir com o VS Code;

## Html básico (3)

- ▶ Confirme a credencial de “Trust” para o autor, você, marcando a opção:
- ▶ “Trust the authors of all files in the parent folder ‘xxxx’”.
- ▶ Através do VS Code crie o arquivo

# Html básico (4)

Digite o seguinte código:

```
<p>Bem-vindo ao mundo do html 2a linha</p>
<p>Não pulei uma linha</p>
<br />
<p>Agora pulei uma linha</p>
<br />
<h1>Primeiro título</h1>
<h2>Segundo título</h2>
<h3>Terceiro título</h3>
<hr />
<p>Nesta linha, <i>isto está itálico</i> <b>e isto está em negrito</b>></p>
```

# Html básico (5)

Execute o código:

```
<p>Bem-vindo ao mundo do html</p>
<p>Não pulei uma linha</p>
<br />
<p>Agora pulei uma linha</p>
<br />
<h1>Primeiro título</h1>
<h3>Terceiro título</h3>
<hr />
<p>Nesta linha, <i>isto está itálico</i>
<b>e isto está em negrito</b>></p>
```

Bem-vindo ao mundo do html

Não pulei uma linha

Agora pulei uma linha

## Primeiro título

### Segundo título

#### Terceiro título

---

Nesta linha, *isto está itálico* e **isto está em negrito**>

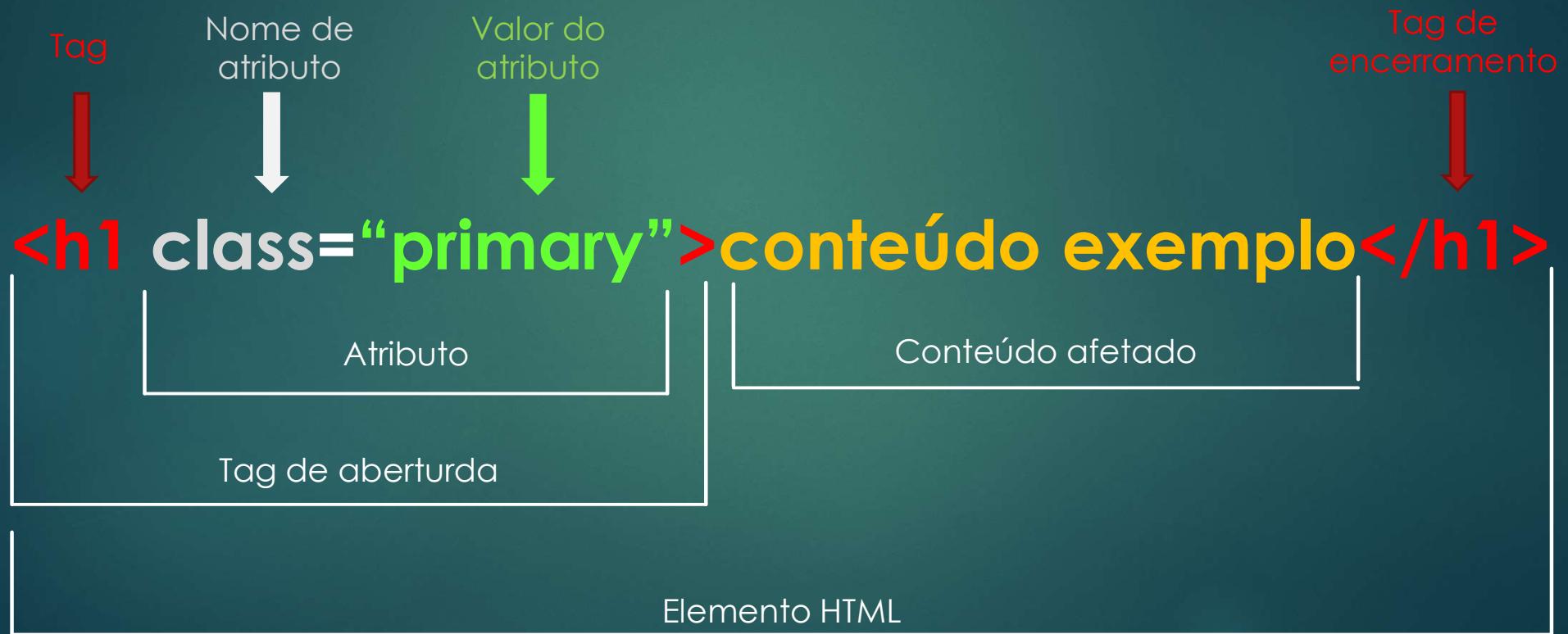
# Html básico (6)

Tags básicas de formatação de texto:

<b>	Negrito
<big>	Grande
<em>	Enfatizado
<i>	ítálico
<small>	Pequeno
<strong>	Forte
<sub>	Subscrito
<sup>	Sobrescrito
<ins>	Inserção de texto
<del>	Eliminação de texto

# Html básico (7)

Estrutura da sintaxe de marcação:



# Html básico (8)

Estrutura da página html:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Título da página</title>
    <meta charset="utf-8">
  </head>
  <body>
    Código html
  </body>
</html>
```

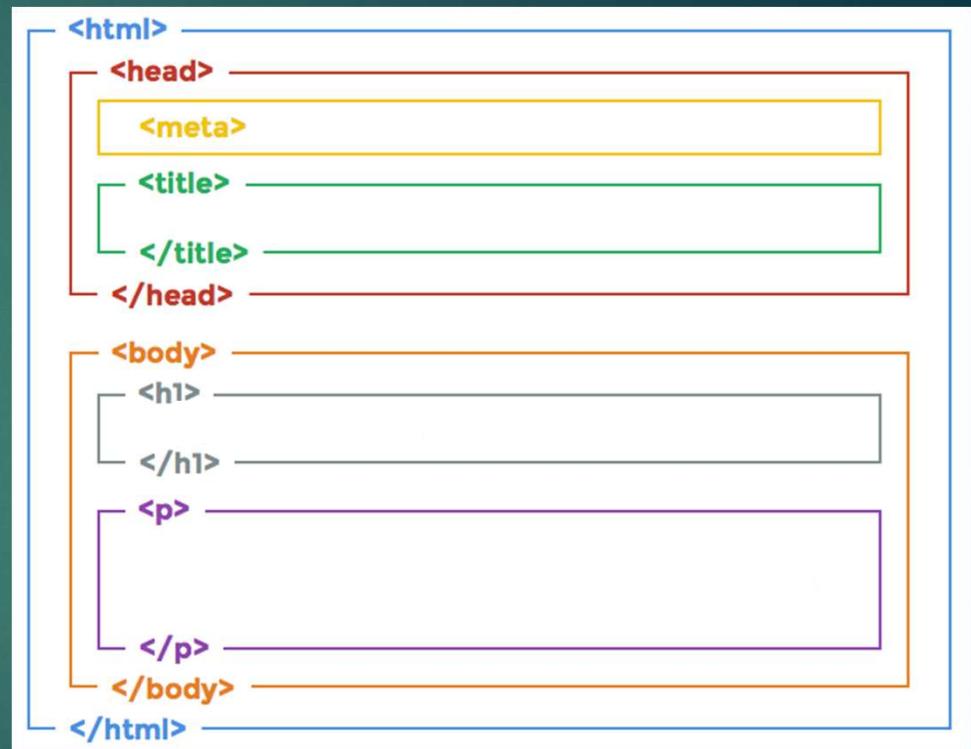
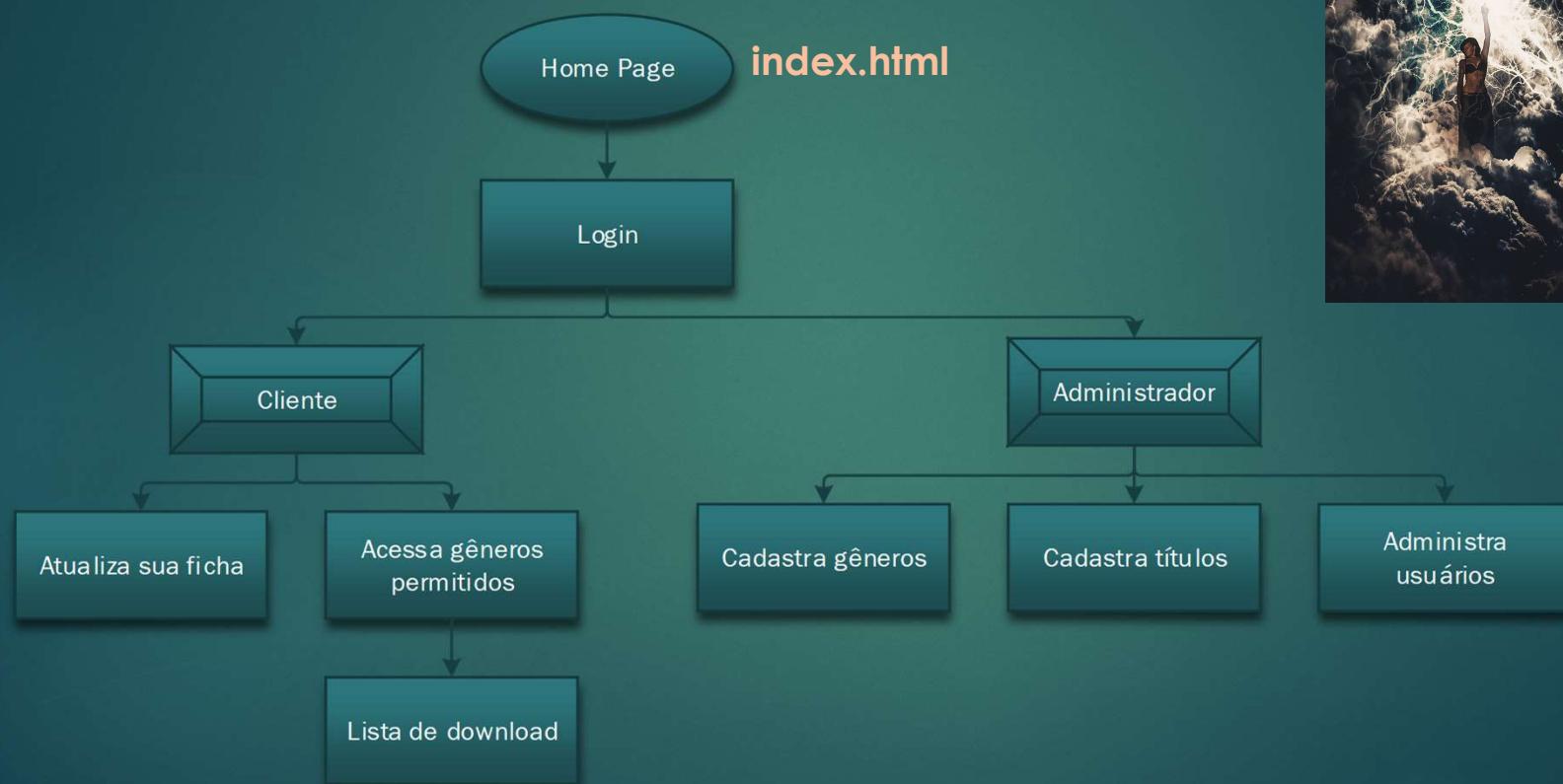


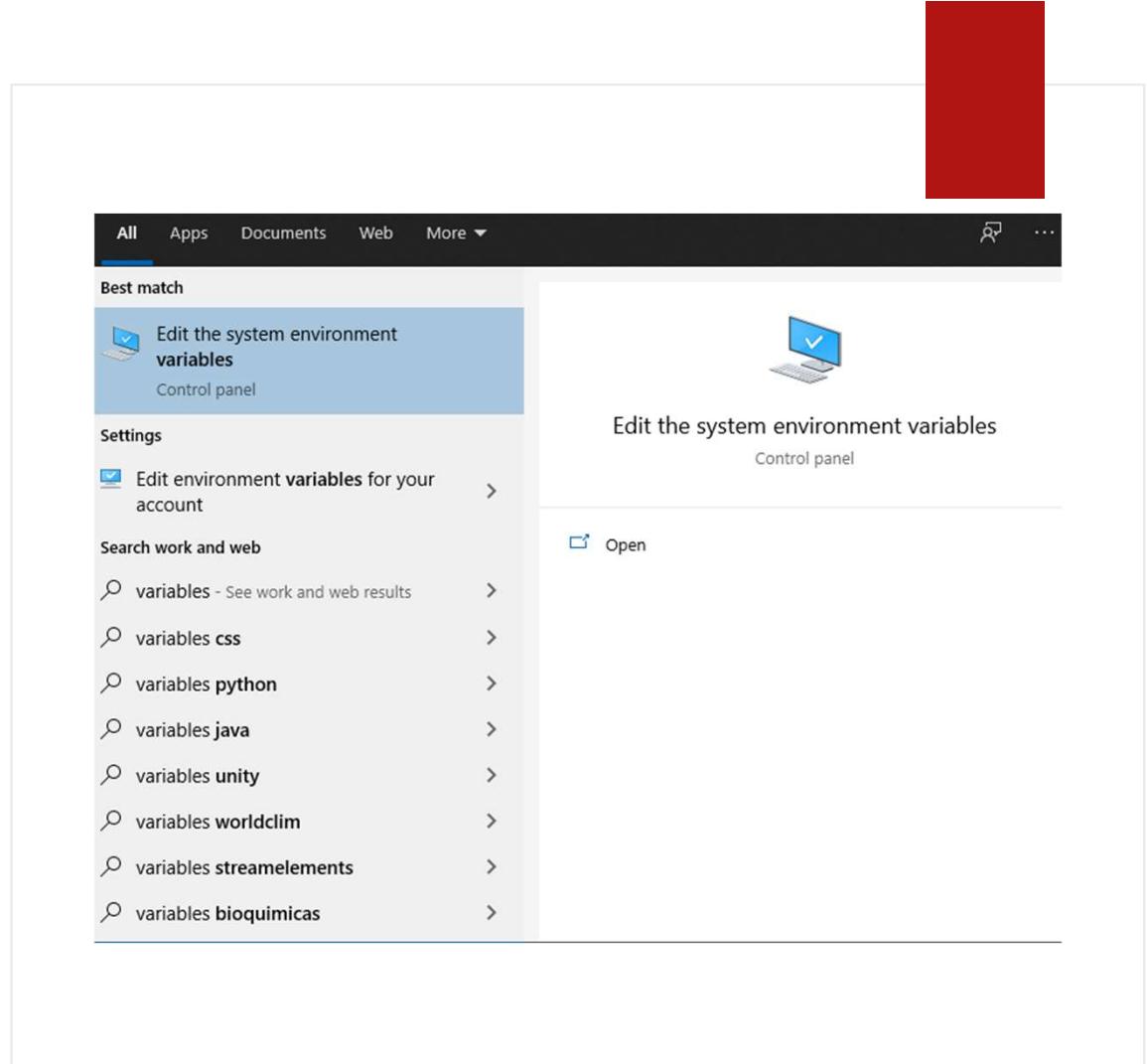
Imagen: pensandonaweb.com.br

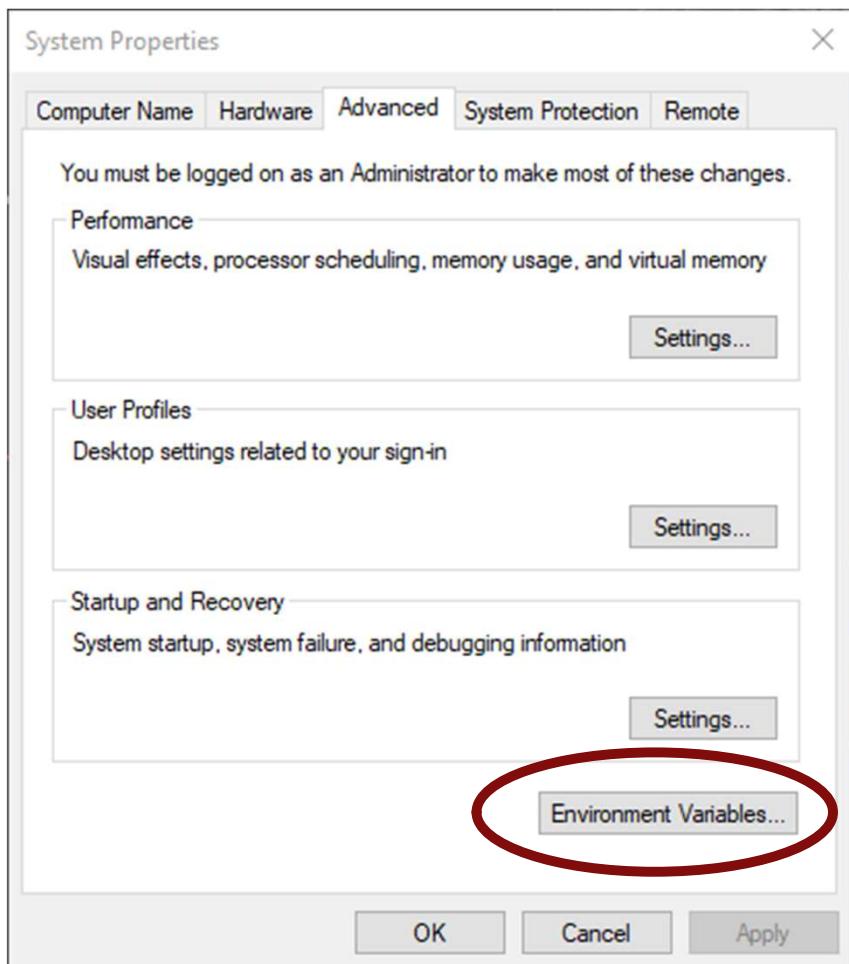
# Fulgora loja de podcasts



# Tunning do PowerShell (1): Chamada no Notepad++ pela linha de comando

- Na linha da Cortana digite “variable”;
- Duplo clique no Editor de variáveis apresentado:



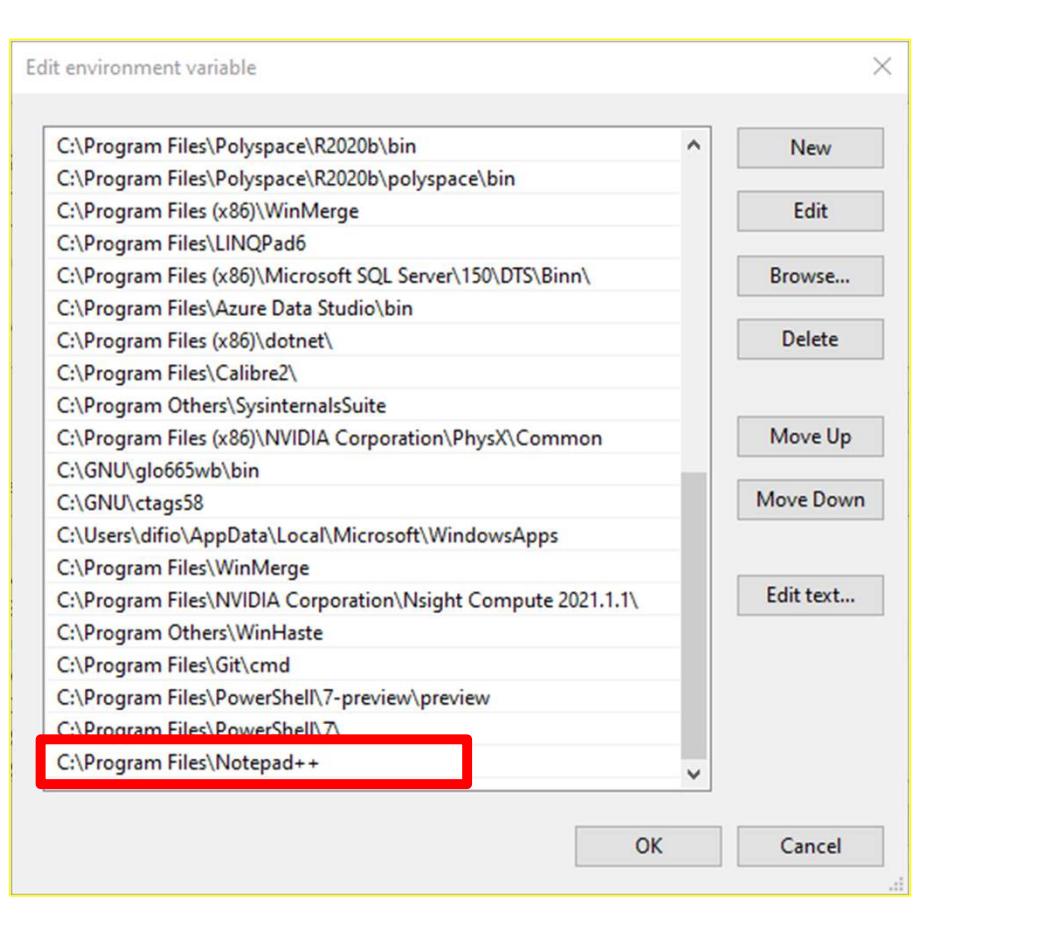


## Tunning do PowerShell (2): Chamada no Notepad++ pela linha de comando

SELECIONE  
VARIÁVEIS DE  
AMBIENTE:

# Tunning do PowerShell (1): Chamada no Notepad++ pela linha de comando

ENCONTRE A VARIÁVEL  
PATH E INCLUA O CAMINHO  
PARA O NOTEPAD++





V:  
Git e Github

# Git básico (1)

Para fixar conceitos  
veremos inicialmente  
através da linha de  
comando do PowerShell.

# Git básico (2)

## Bibliografia:

Abidsov, J. (2020). Pratical Git. Apress.

Belanger, J. A. (2019). Mastering Git. Razeware LLC.

Demaree, D. (2010). Git fot Humans. Wiley Publishing, Inc.

França, D. F. (s.d.). Fonte: Um guia para iniciantes na área de web:  
<https://tableless.github.io/iniciantes/>

Hodson, R. (s.d.). Ry's Git Tutorial. Kindle Edition

Laster, B. (s.d.). Professional Git®. wrox.

Liberty, J. (2021). Git for Programmers - Master Git for effective implementation of version. Packt Publishing.

Narębski, J. (2016). Mastering Git. Packt Publishing.

Overflow, b. t. (s.d.). Git® Notes for Professionals.

Straub, S. C. (2014). Pro Git Second Edition., Apress

Tsitoara, M. (s.d.). Beginning Git and GitHub A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer. Apress.

# Git básico (3)

## Instalação do posh-git

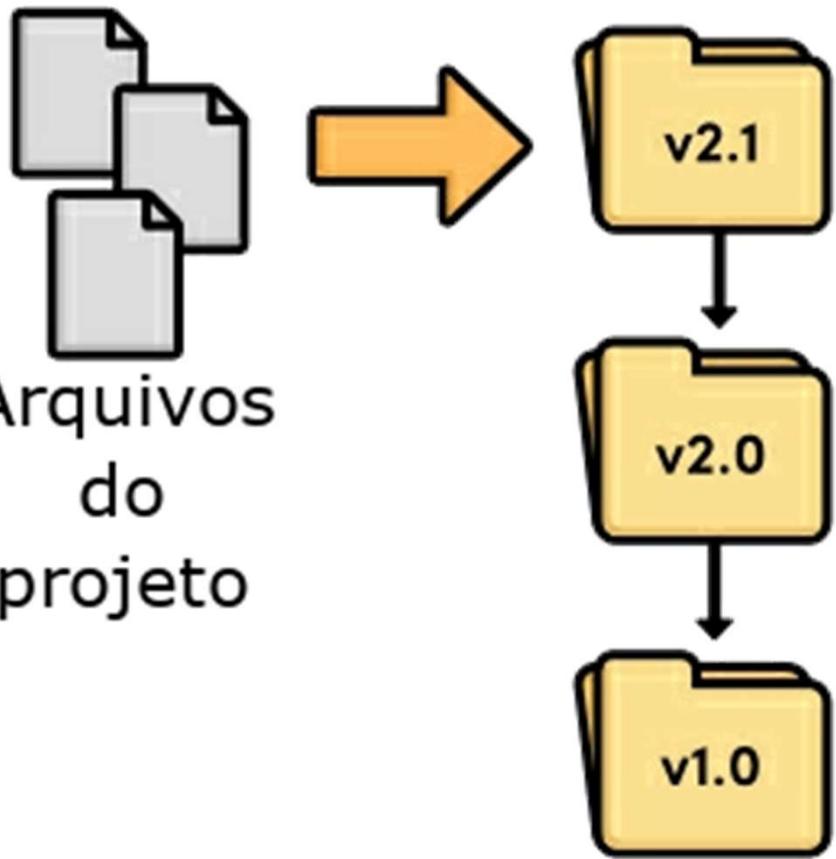
- ▶ Abra um terminal PowerShell em modo Administrador;
- ▶ Digite:  
**PowerShellGet\Install-Module posh-git -Scope CurrentUser**
- ▶ Abra o Notepad++ para editar esse arquivo: **Notepad++ \$profile**
- ▶ Adicione ao final do arquivo a linha:  
**Import-Module posh-git**

# Git básico (4)

Utilidades básicas do Git:

- Controle de versões;
- Backup;
- Útil para qualquer tipo de documento que precise controle de versões e backup – não somente software!

Arquivos  
do  
projeto

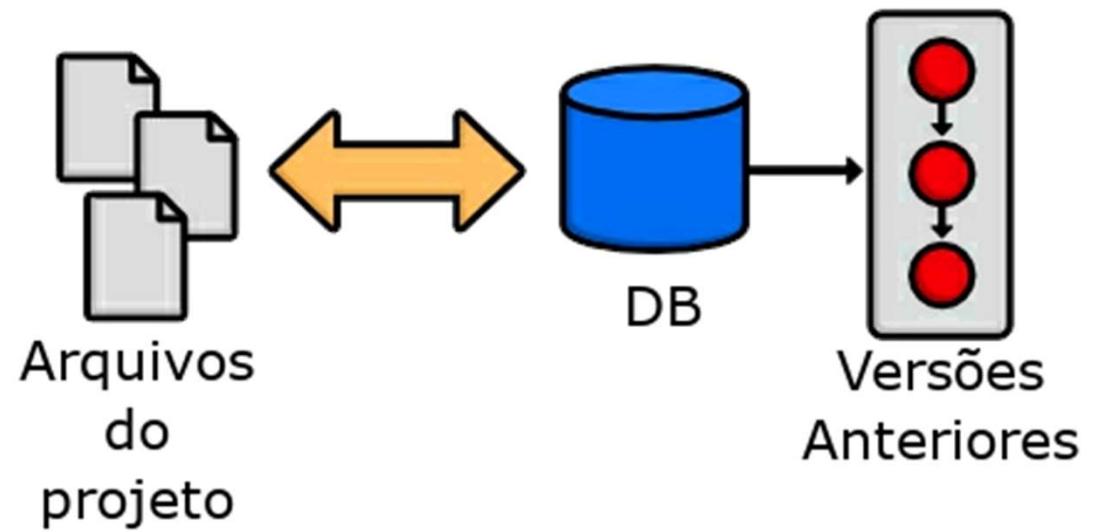


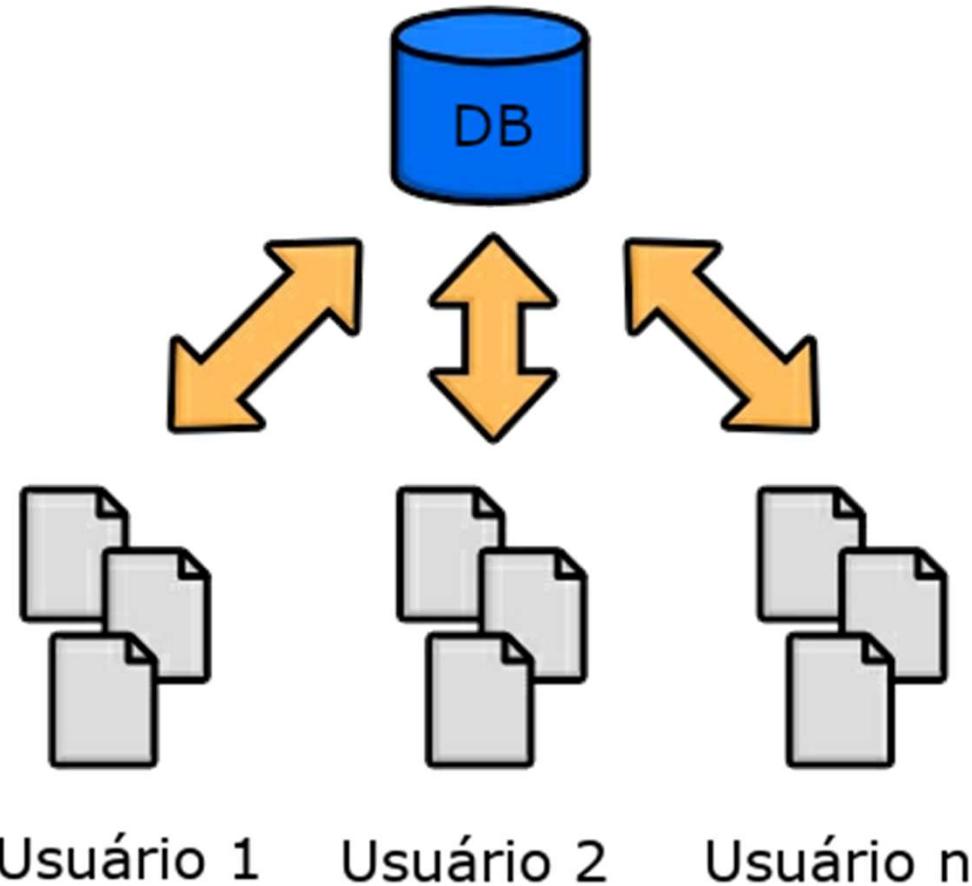
## Git básico (5)

### Revisão por arquivos e pastas

# Git básico (6)

## Controle local de versão

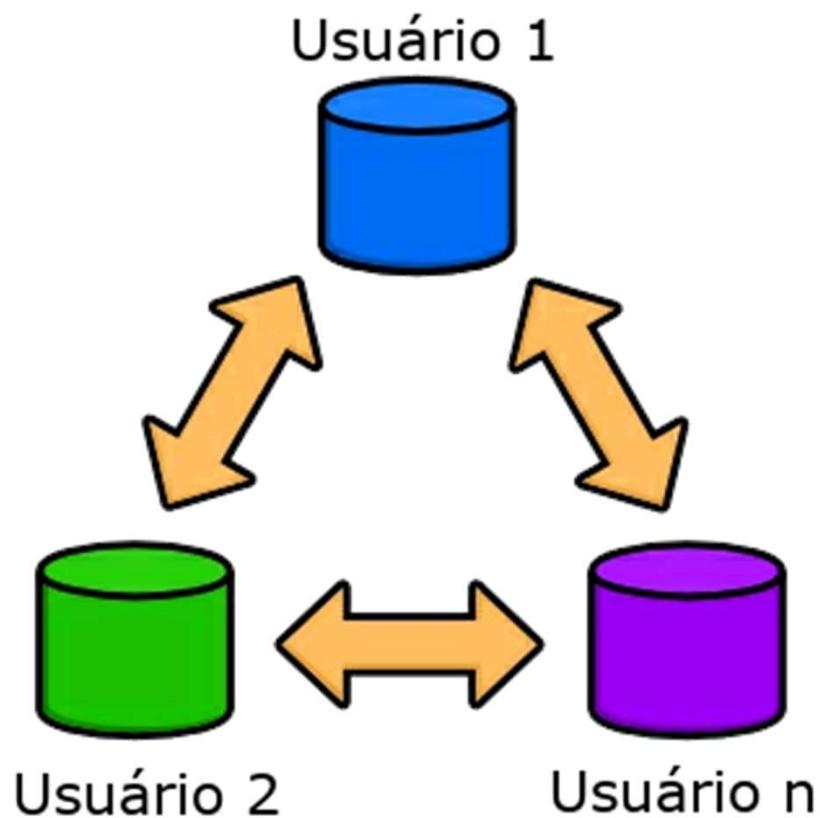




Git básico (7)  
Controle  
centralizado

# Git básico (8)

## Controle distribuído



# Git básico (9)

## Nascimento do Git

- ▶ Criado em 2005 para o desenvolvimento do Linux
- ▶ Vantagens:
  - ▶ Confiável;
  - ▶ Eficiente para grandes projetos;
  - ▶ Suporte ao desenvolvimento distribuído;
  - ▶ Suporte ao desenvolvimento não linear

# Git básico (10)

## Configuração

- ▶ **Atenção:** Use o mesmo usuário e e-mail que cadastrou na conta do GitHub
- ▶ `git config --global user.name "Seu nome"`
- ▶ `git config --global user.email "seu@email.com"`

# Git básico (11)

Criação de um site simples:

- ▶ Crie um diretório para conter o exemplo;
- ▶ Abra o VS Code nesse diretório;
- ▶ Crie o arquivo index.html;
- ▶ Digite e grave o código:

# HTML básico (9)

index.html:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <title>Um Website Colorido</title>
    <meta charset="utf-8" />
</head>
<body>
    <h1 style="color:#07F">Um Website Colorido</h1>
    <p>Este é um website sobre cores!</p>
    <h2 style="color:#C00">Notícias</h2>
    <ul>
        <li>Não aconteceu nada (ainda)</li>
    </ul>
</body>
</html>
```

# HTML básico (10)

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <title>Um Website Colorido</title>
5      <meta charset="utf-8" />
6  </head>
7  <body>
8      <h1 style="color: #07F">Um Website Colorido</h1>
9      <p>Este é um website sobre cores!</p>
10     <h2 style="color: #C00">Notícias</h2>
11     <ul>
12         <li>Não aconteceu nada (ainda)</li>
13     </ul>
14     </body>
15 </html>
```

## Um Website Colorido

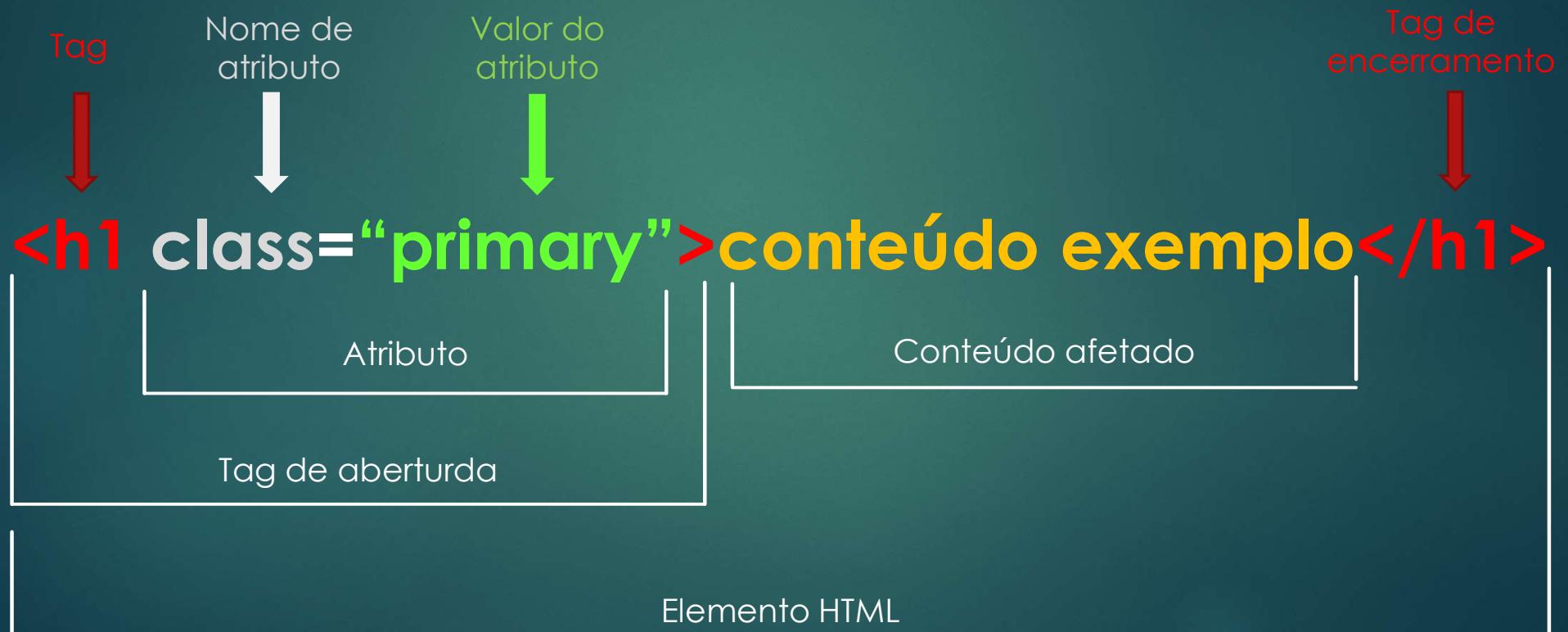
Este é um website sobre cores!

### Notícias

- Não aconteceu nada (ainda)

# Html básico (11)

Relembrando: Estrutura da sintaxe de marcação:



# Html básico (12)

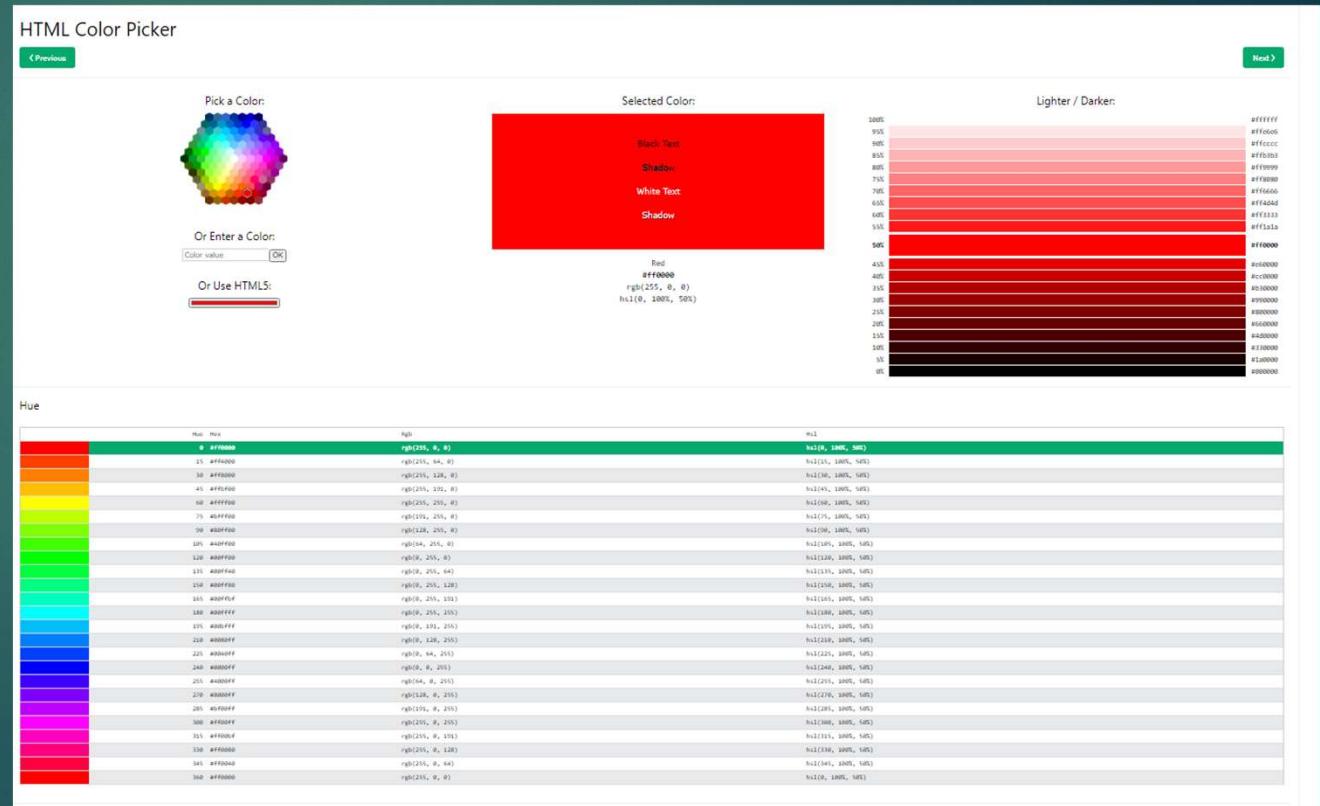
Estrutura da sintaxe de marcação:



# HTML básico (13)

## Color Picker

[https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)



## Git básico (12)

index.html:

- ▶ Abra um terminal PowerShell no diretório que você criou o index.html;
- ▶ No prompt, inicialize o seu repositório:
- ▶ **git init**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Git> pwd

Path
----
C:\Users\difio\OneDrive - Fatec Centro Paula Souza\estudos\Fatec\Turma\App Web de A-Z\Exemplos\Git

Git> git init
Initialized empty Git repository in C:/Users/difio/OneDrive - Fatec Centro Paula Souza/estudos/Fatec/Tur
ma/App Web de A-Z/Exemplos/Git/.git/
Git>
```

# Git básico (13)

## Inicializando o repositório Git local:

```
Git> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

nothing added to commit but untracked files present (use "git add" to track)
Git>
```

## Git básico (14)

Verificar o status do repositório:

**git status**

stage

/steɪdʒ/

US  UK 

## Noun

1. any distinct time period in a sequence of events  
(phase)
  - *we are in a transitional stage in which many former ideas must be revised or rejected*
2. a specific identifiable position in a continuum or series or especially in a process  
(degree; level; point)
  - *a remarkable degree of frankness*
  - *at what stage are the social sciences?*

Git básico  
(15)  
Entendendo  
“stage”:

# Git básico (16)

Faça um primeiro instantâneo, gerando um “stage”:

► **git add index.html**

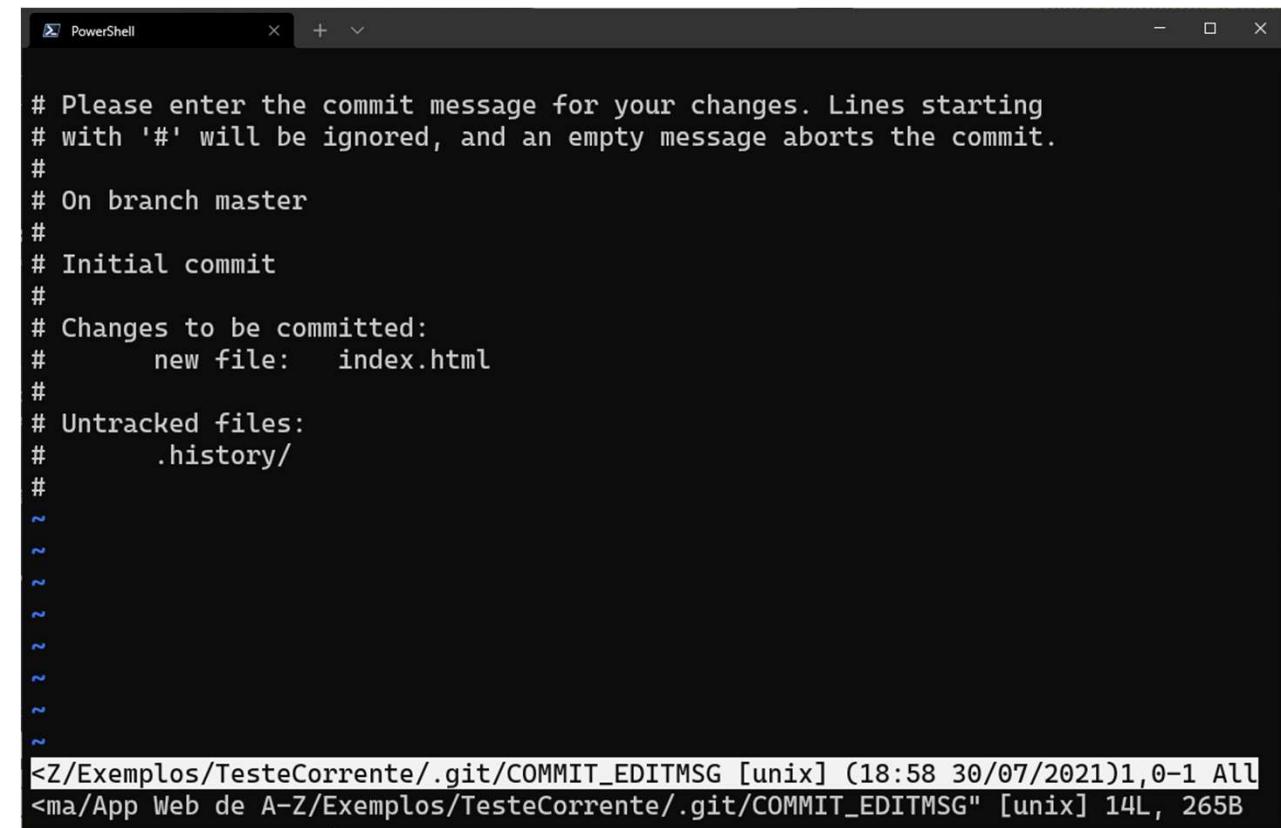
► **git status**

```
TesteCorrente> git add index.html
TesteCorrente> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .history/
```

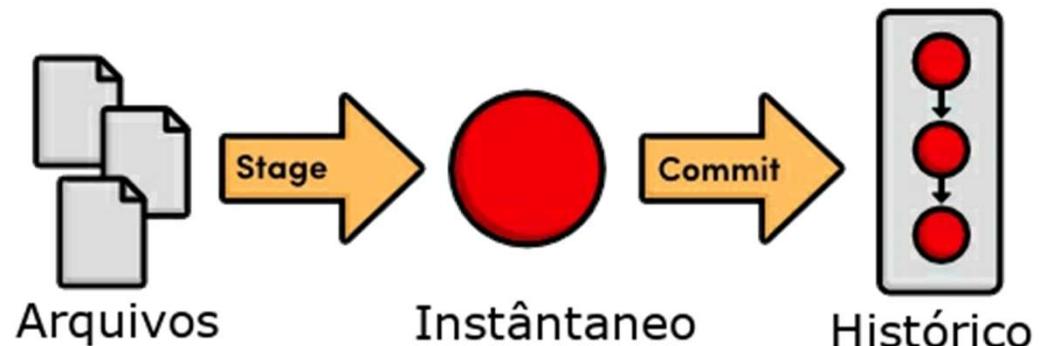


# Git básico

## (17)

### commit e V|:

# Git básico (18) Stage e Commit:



# Git básico (19)

stage e commit

“stage” define quais arquivos incluir no próximo “commit”

“commit” grava os instantâneos em estage com uma descrição”

```
TesteCorrente> git log
commit 81d182d5eec7143e1fca12f96a5ee967730af410 (HEAD → master)
Author: Sergio Di Fiore <difiore@outlook.com>
Date:   Fri Jul 30 19:31:29 2021 -0300

    index.html
TesteCorrente>
```

## Git básico (20)

História do repositório:

git log

# Git básico (21)

Criando novos arquivos HTML:

Abra novos documentos no

VS Code, e digite os dois arquivos:

“laranja.html” e “azul.html”, não

esquecendo de gravar

# Git básico (21)

laranja.html:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <title>A página laranja</title>
5      <meta charset="utf-8" />
6  </head>
7  <body>
8      <h1 style="color: #F90">A página laranja</h1>
9      <p>Laranja é tção legal que tem até
10     <span style="color: #F90">frutas</span> com esse nome.</p>
11  </body>
12 </html>
```

You, 3 weeks ago • laranja e azul

# Git básico (22)

azul.html:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <title>A página azul</title>
5      <meta charset="utf-8" />
6  </head>
7  <body>
8      <h1 style="color: #00F">A página azul</h1>
9      <p>Azul é a cor do céu.</p>          You, 3 weeks
10 </body>
11 </html>
```

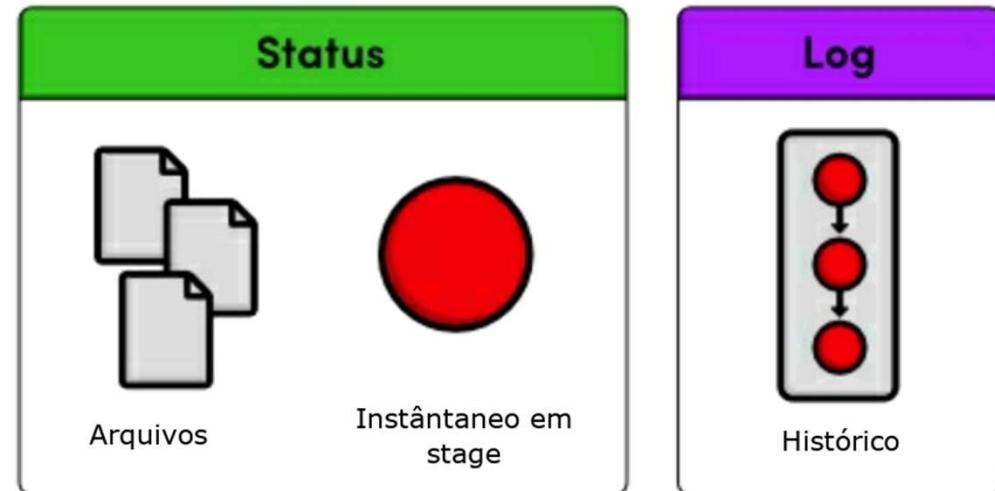
# Git básico (24)

Coloque-os em stage:

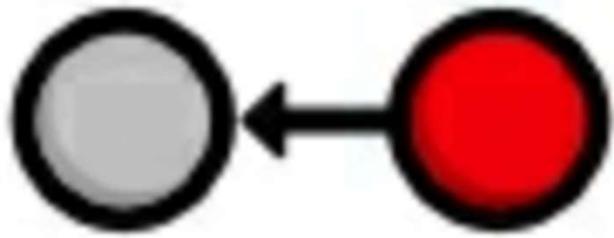
► `git add laranja.html azul.html`

► `git status`

► `git log`



Git básico (25)  
Status x Log:



Git básico (26)  
commit:

**git commit**

# Git básico (27)

## VI:

# Git básico (28)

Modificando as páginas HTML:

- ▶ Modifique a página index.html conforme indicado e
- ▶ Adicione um link à home page nas duas outras páginas

# Git básico (29)

index.html:

```
1  <!DOCTYPE html>          You, 3 weeks ago • laranja e azul
2  <html lang="pt-br">
3  <head>
4  |   <title>Um Website Colorido</title>
5  |   <meta charset="utf-8" />
6  </head>
7  <body>
8  |   <h1 style="color: #07F">Um Website Colorido</h1>
9  |   <p>Este é um website sobre cores!</p>
10
11  |   <h2 style="color: #C00">Notícias</h2>
12  |   <ul>
13  |   |   <li>Não aconteceu nada (ainda)</li>
14  |   </ul>
15  |   <h2>Navigation</h2>
16  <ul>
17  |   <li style="color: #F90">
18  |   |   <a href="orange.html">A página laranja</a>
19  |   </li>
20  |   <li style="color: #00F">
21  |   |   <a href="blue.html">A página azul</a>
22  |   </li>
23  </ul>
24  </body>
25  </html>
```

# Git básico (30)

laranja.html:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4  |  <title>A página laranja</title>
5  |  <meta charset="utf-8" />
6  </head>
7  <body>
8  |  <h1 style="color: #F90">A página laranja</h1>
9  |  <p>Laranja é tção legal que tem até
10 |    <span style="color: #F90">frutas</span> com esse nome.</p>
11 |  <p><a href="index.html">Voltar à Home Page</a>
12 </body>
13 </html>
```

You, 3 weeks ago • laranja e azul

# Git básico (31)

azul.html:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <title>A página azul</title>
5      <meta charset="utf-8" />
6  </head>
7  <body>
8      <h1 style="color: #00F">A página azul</h1>
9      <p>Azul é a cor do céu.</p>
10     <p><a href="index.html">Voltar à Home Page</a>
11  </body>
12 </html>
```

You, 3 weeks ago • laranja e azul

# Git básico (32)

Stage e commit dos instantâneos:

```
git status
```

```
git add index.html laranja.html  
azul.html
```

```
git status
```

```
git commit -m "Adicionada navegação"
```

# Git básico (33)

git log --oneline:

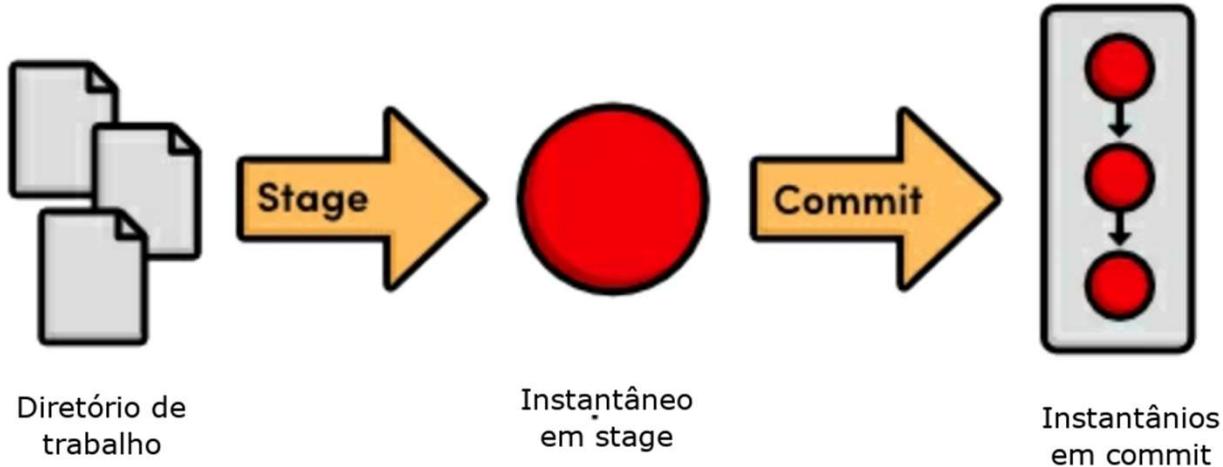
► `git log`

► `git log -2`

► `git log --oneline`

► `git status`

► `git log --oneline azul.html`



# Git básico (34)

Conclusão: O fluxo fundamental do Git:

# Git básico (35)

Criando uma chave criptográfica para interagir com o GitHub:

- ▶ Inicie uma sessão PowerShell no diretório do exemplo;
- ▶ **ssh-keygen -t rsa -b 4096 -C seuEmail@aqui.com**
- ▶ Ao prompt informe uma senha e confirme;
- ▶ **Start-Process -filepath powershell.exe -Verb Runas -ArgumentList @('Set-Service -StartupType Manual ssh-agent')**

# Git básico (36)

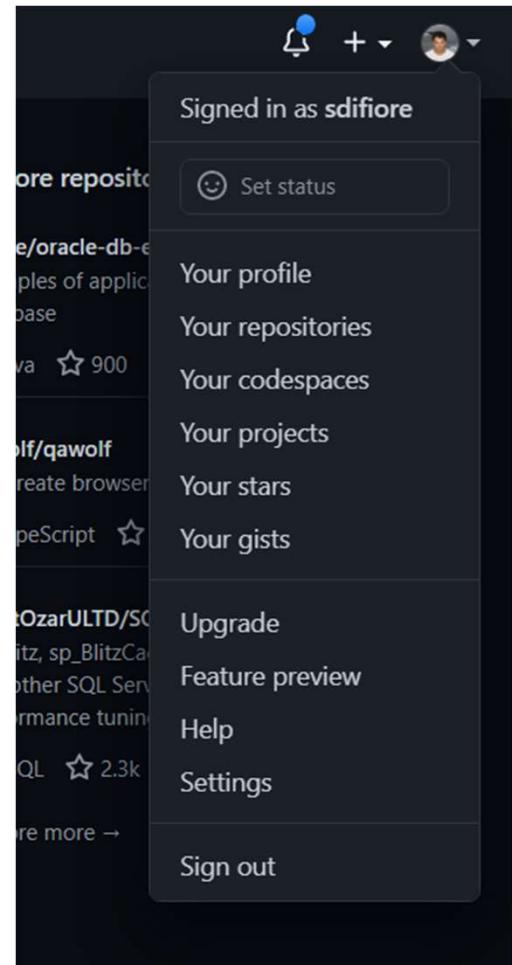
Criando uma chave criptográfica para interagir com o GitHub:

- ▶ `Get-Content $HOME\.ssh\id_rsa.pub | clip.exe`
- ▶ A linha anterior escreverá na área de transferência a chave criptográfica. **Não apague**;

# Git básico (37)

Gravando a chave no GitHub:

- Acesse sua conta no GitHub";
- Acesse a página "settings";



# Git básico (38)

Gravando a chave no GitHub:

- ▶ No menu à esquerda, acesse: SSH and GPC;
- ▶ Crie uma nova chave;
- ▶ Copie a que está na sua área de transferência
  - ▶ CTRL + V
- ▶ Grave

The screenshot shows the 'SSH keys' section of a GitHub account. At the top right is a green button labeled 'New SSH key'. Below it, a message reads: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' A single SSH key is listed:

- galileo** (key icon)
- SHA256: j+uXYwgpyt5g5czvrx1NmERqS3B1HDzFLA+3UILiwfI
- Added on 11 Jul 2021
- Never used — Read/write

At the bottom, there's a link to 'generating SSH keys' and 'troubleshoot common SSH problems'.

# Git básico (39)

Criando novos repositório no GitHub:

- ▶ No GitHub acesse:

<https://github.com/sdifiore/cores>;

- ▶ Faça o “fork” para a sua conta;
- ▶ Recomendação: Deixe-o público;

# Git básico (32)

Download dos arquivos:

- ▶ Escolha um diretório para trabalhar e abra o PowerShell;
- ▶ `git clone https://github.com/<seu usuário>/cores.git`
- ▶ `cd cores`
- ▶ `git log --oneline`

# Git básico (40)

## Após o download:

```
Git Alterações> git clone https://github.com/sdifiore/cores.git
Cloning into 'cores' ...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 14 (delta 5), reused 14 (delta 5), pack-reused 0
Receiving objects: 100% (14/14), done.
Resolving deltas: 100% (5/5), done.
Git Alterações> cd cores
cores> git log --oneline
a9bc51d (HEAD → main, origin/main, origin/HEAD) Adicionada navegação
7f619a7 Textos em português
85312e7 laranja e azul
cores>
```



The screenshot shows a GitHub commit history for the 'main' branch. It displays three commits made by 'sdifiore' on July 10, 2021:

- Adicionada navegação** (Unverified, a9bc51d) - committed 14 days ago
- Textos em português** (Unverified, 7f619a7) - committed 14 days ago
- laranja e azul** (Unverified, 85312e7) - committed 14 days ago

# Git básico (41)

## Compare com o GitHub:

# Git básico (42)

Verifique uma das revisões anteriores

► **git checkout 7f619a7**

► Atenção para colocar o valor que aparece para você no lugar de 7f619a7, uma vez que pode diferir do exemplo!

# Git básico (43)

Verifique uma das revisões anteriores

- A saída do comando apresenta um grande conjunto de informações com relação ao estado do cabeçalho (HEAD), que no momento vamos desconsiderar.

```
cores> git log --oneline
a9bc51d (HEAD → main, origin/main, origin/HEAD) Adicionada navegação
7f619a7 Textos em português
85312e7 laranja e azul
cores> git checkout 7f619a7
Note: switching to '7f619a7'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

```
HEAD is now at 7f619a7 Textos em português
cores>
```

# Git básico (44)

Verifique uma das revisões anteriores

- Abrindo o arquivo index.html no visual estúdio, verifica-se que o arquivo existente é realmente daquela versão anterior:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4  | <title>Um Website Colorido</title>
5  | <meta charset="utf-8" />
6  </head>
7  <body>
8  | <h1 style="color: #07F">Um Website Colorido</h1>
9  | <p>Este é um website sobre cores!</p>
10 
11 <h2 style="color: #C00">Notícias</h2>
12 <ul>
13 | <li>Não aconteceu nada (ainda)</li>
14 </ul>
15 </body>
16 </html>
```

# Git básico (45)

Verifique no GitHub essa versão:

As mudanças efetuadas

Showing 2 changed files with 5 additions and 5 deletions.

Unified Split

4 azul.html

```
@@ -1,11 +1,11 @@
1   1   <!DOCTYPE html>
2   2   <html lang="pt-br">
3   3     <head>
4   4       - <title>The Blue Page</title>
5   5       + <title>A página azul</title>
6   6         <meta charset="utf-8" />
7   7     </head>
8   8     <body>
9   9       - <p>Blue is the color of the sky.</p>
9   9       + <p>Azul é a cor do céu.</p>
10 10     </body>
11 11   </html>
```

6 laranja.html

```
@@ -1,12 +1,12 @@
1   1   <!DOCTYPE html>
2   2   <html lang="pt-br">
3   3     <head>
4   4       - <title>The Orange Page</title>
4   4       + <title>A página laranja</title>
5   5         <meta charset="utf-8" />
6   6     </head>
7   7     <body>
8   8       - <h1 style="color: #00F">The Orange Page</h1>
8   8       + <h1 style="color: #F90">A página laranja</h1>
9   9       <p>Orange is so great it has a
10 10      - <span style="color: #F90">fruit</span> named after it.</p>
10 10      + <span style="color: #F90">frutas</span> com nome por isso.</p>
11 11     </body>
12 12   </html>
```

16 lines (15 sloc) | 333 Bytes

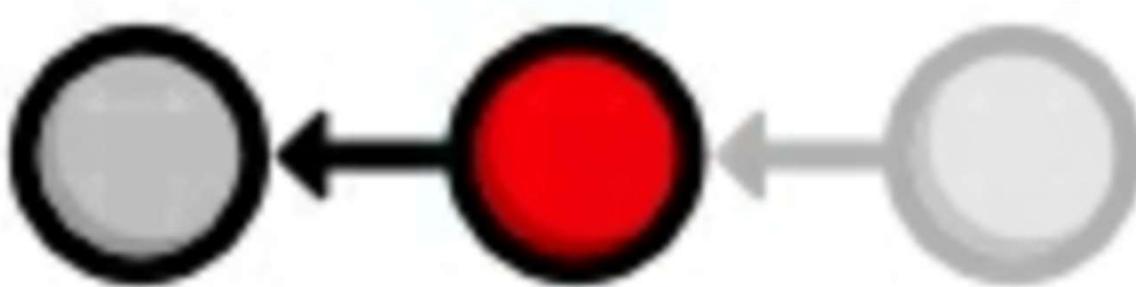
Raw Blame   

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <title>Um Website Colorido</title>
5      <meta charset="utf-8" />
6  </head>
7  <body>
8      <h1 style="color: #07F">Um Website Colorido</h1>
9      <p>Este é um website sobre cores!</p>
10
11     <h2 style="color: #C00">Notícias</h2>
12     <ul>
13         <li>Não aconteceu nada (ainda)</li>
14     </ul>
15 </body>
16 </html>
```

# Git básico (46)

Verifique no GitHub essa versão:

O MESMO CÓDIGO QUE TEMOS NO VS CODE NO MOMENTO



# Git básico (47)

Verifique uma das revisões anteriores

O que se está vendo é uma volta ao passado, o 2º commit

# Git básico (48)

Verifique uma das revisões ainda mais anteriores (1/3):

► **git checkout 85312e7**

► Lembrando novamente que você deve usar o valor que aparece para você no lugar de 7f619a7, uma vez que pode diferir do exemplo!

# Git básico (49)

Verifique uma das revisões ainda mais anteriores (2/3):

- É possível que você tenha uma mensagem informando que você deve efetuar um commit. Isso pode acontecer por qualquer alteração, mínima que seja, quando aberto o arquivo do VS Code. Se for o caso, faça o commit e em seguida o checkout.

## Git básico (50)

Verifique uma das revisões ainda mais anteriores (3/3):

- ▶ Temos agora a situação do 1º commit



# Git básico (51)

Retornando à situação atual:

► **git checkout main**

► Fique de olho: É muito comum, além do termo “main” empregar-se o “master”.

# Git básico (52)

Rotule a versão:

```
► git tag -a v1.0 -m "Versão estável do Website"
```

- ▶ Tags são formas eficientes de marcar pontos importantes no projeto;
- ▶ Pode-se usar, no exemplo, a tag v1.0 ao invés de sua identidade hexadecimal;
- ▶ A chave **-a** indica que será criada uma “tag anotada”, o que faz com que sejam gravados também o nome do usuário, a data e uma mensagem de descrição, esta última vindo em seguida à uma chave **-m**

# Html básico (14)

Estrutura da página html:

- ▶ No diretório cores, crie um diretório “Images”
- ▶ Grave a imagem de:  
<https://www.difiore.com.br/images/Cherokee.jpg>
- ▶ Pode ser feito direto do navegador ou pelo PowerShell como abaixo:

```
mkdir Images
```

```
$url = "https://www.difiore.com.br/images/Cherokee.jpg"
```

```
Invoke-WebRequest $url -OutFile .\Images\Cherokee.jpg
```

# Html básico (15)

Página aviacao.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <title>A página da aviação</title>
    <meta charset="utf-8" />
</head>
<body>
    <h1 style="color: #00F">Este é o Piper Cherokee PA-28</h1>
    
    <p>Aeronave de treinamento.</p>
    <p><a href="index.html">Voltar à Home Page</a>
</body>
</html>
```

# Html básico (16)

tag image:

```

```

```

```

# Html básico (17)

Atualizando index.html:

```
<li style="color: #1176ee">  
  <a href="aviacao.html">A página da aviação</a>  
</li>
```

# Git básico (53)

*Desfazendo alterações que já passara por um commit (1/13):*

- ▶ Crie a página aviação.html e grave;
- ▶ Crie, em baixo do diretório corrente um “Images” e lá coloque a imagem;
- ▶ Altere index.html

**git status**

```
cores> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
            modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .history/
        Images/
        aviacao.html

no changes added to commit (use "git add" and/or "git commit -a")
cores>
```

```
Your branch is up to date with 'origin/main'.
```

```
Changes not staged for commit:  
(use "git add <file> ..." to update what will be committed)  
(use "git restore <file> ..." to discard changes in working directory)  
 modified: index.html
```

```
Untracked files:  
(use "git add <file> ..." to include in what will be committed)  
.history/  
Images/  
aviacao.html
```

```
no changes added to commit (use "git add" and/or "git commit -a")  
cores> git add index.html aviacao.html Images  
cores> git status  
On branch main  
Your branch is up to date with 'origin/main'.
```

```
Changes to be committed:  
(use "git restore --staged <file> ..." to unstage)  
 new file: Images/Cherokee.jpg  
 new file: aviacao.html  
 modified: index.html
```

```
Untracked files:  
(use "git add <file> ..." to include in what will be committed)  
.history/
```

```
cores> git commit -m "Adicionada pagina Aviação"  
[main 1998e48] Adicionada pagina Aviação  
 3 files changed, 15 insertions(+)
```

## Git básico (54)

Desfazendo alterações que já passara por um commit (2/13):

```
create mode 100644 Images/Cherokee.jpg
create mode 100644 aviacao.html
cores> git log
commit 1998e483ead10bb0e5f13c0be73914ff864d31d (HEAD → main)
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Tue Aug 3 10:09:35 2021 -0300
```

Adicionada pagina Aviação

```
commit a9bc51d43dc2cebe609c57ea0bf80d03d276500b (tag: v1.0, origin/main, origin/
HEAD)
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Sat Jul 10 20:13:37 2021 -0300
```

Adicionada navegação

```
commit 7f619a76593b396b3576ed683e9186544f658cf8
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Sat Jul 10 20:05:41 2021 -0300
```

Textos em português

```
commit 85312e72ed91ca987b439bd208f2176368289d2e
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Sat Jul 10 19:57:13 2021 -0300
```

laranja e azul

```
cores>
```

## Git básico (55)

Desfazendo alterações que já passara por um commit (3/13):

```
create mode 100644 Images/Cherokee.jpg
create mode 100644 aviacao.html
cores> git log
commit 1998e483ead10bb0e5f13c0be73914ff864d31d (HEAD → main)
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Tue Aug 3 10:09:35 2021 -0300
```

Adicionada pagina Aviação

```
commit a9bc51d43dc2cebe609c57ea0bf80d03d276500b (tag: v1.0, origin/main, origin/
HEAD)
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Sat Jul 10 20:13:37 2021 -0300
```

Adicionada navegação

```
commit 7f619a76593b396b3576ed683e9186544f658cf8
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Sat Jul 10 20:05:41 2021 -0300
```

Textos em português

```
commit 85312e72ed91ca987b439bd208f2176368289d2e
Author: Sergio Di Fiore <difiores@outlook.com>
Date:   Sat Jul 10 19:57:13 2021 -0300
```

laranja e azul

```
cores>
```

## Git básico (56)

Desfazendo alterações que já passara por um commit (4/13):

```
1998e48 (HEAD → main) Adicionada pagina Aviação
a9bc51d (tag: v1.0, origin/main, origin/HEAD) Adicionada navegação
7f619a7 Textos em português
85312e7 laranja e azul
cores> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file> ..." to include in what will be committed)
    .history/

nothing added to commit but untracked files present (use "git add" to track)
```

## Git básico (57)

Desfazendo alterações que já passara por um commit (5/13):

```
cores> git log --oneline
1998e48 (HEAD → main) Adicionada pagina Aviação
a9bc51d (tag: v1.0, origin/main, origin/HEAD) Adicionada navegação
7f619a7 Textos em português
85312e7 laranja e azul
cores> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file> ..." to include in what will be committed)
    .history/

nothing added to commit but untracked files present (use "git add" to track)
cores>
```

## Git básico (58)

Desfazendo alterações que já passara por um commit (6/13):

# Git básico (59)

Desfazendo alterações que já passara por um commit (7/13):  
Voltando à versão estável (**v1.0**)

As versões que possuímos:

```
1998e48 (HEAD → main) Adicionada pagina Aviação
a9bc51d (tag: v1.0, origin/main, origin/HEAD) Adicionada navegação
7f619a7 Textos em português
85312e7 laranja e azul
```

## git checkout v1.0

```
cores> git checkout v1.0
HEAD is now at a9bc51d Adicionada navegação
cores>
```

```
cores> dir
```

```
Directory: C:\Users\difio\OneDrive - Fatec Centro Paula  
Souza\estudos\Fatec\Turma\App Web de A-Z\Exemplos\WebExemplos\cores
```

Mode	LastWriteTime	Length	Name
la---	03/08/2021	10:00	.history
la---	03/08/2021	09:17	270 azul.html
-a---	03/08/2021	10:23	551 index.html
la---	03/08/2021	09:17	344 laranja.html

```
cores>
```

## Git básico (60)

Desfazendo alterações que já passara por um commit (8/13):  
Voltando à versão estável (v1.0): **aviação.html não está presente**

# Git básico (61)

Desfazendo alterações que já passara por um commit (9/13):

Retornando à atualização *mais recente*

```
git checkout main(1)
git log --oneline
```

```
cores> git checkout main
Previous HEAD position was a9bc51d Adicionada navegação
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
cores> git log --oneline
1998e48 (HEAD → main) Adicionada pagina Aviação
a9bc51d (tag: v1.0, origin/main, origin/HEAD) Adicionada navegação
7f619a7 Textos em português
85312e7 laranja e azul
cores>
```

<sup>(1)</sup> Alternativamente “master”

# Git básico (62)

Desfazendo alterações que já passara por um commit (10/13):

Efetivamente desfazendo!

► Lembrando que:

```
1998e48 (HEAD → main) Adicionada pagina Aviação
a9bc51d (tag: v1.0, origin/main, origin/HEAD) Adicionada navegação
7f619a7 Textos em português
85312e7 laranja e azul
```

git revert 1998e48

# Git básico (63)

Desfazendo alterações que já passara por um commit (11/13):

Efetivamente desfazendo!

O VI é aberto para digitar-se uma mensagem para a reversão, e já traz uma mensagem default, que pode-se aceitar.

Para gravar e sair do VI, lembre-se **[ESC]:wq**

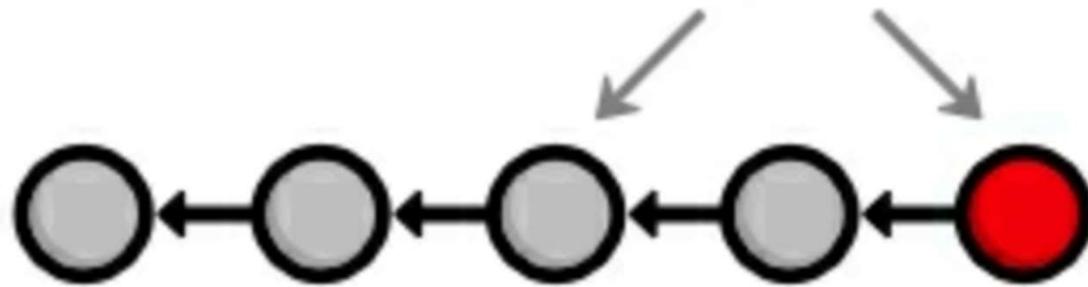
**Note:** Foi criado um quinto commit que desfaz o quarto e esse não é perdido!

Dessa forma, o quinto commit e o terceiro são idênticos!

```
[main c786105] Revert "Adicionada pagina Aviação"  
 3 files changed, 15 deletions(-)  
 delete mode 100644 Images/Cherokee.jpg  
 delete mode 100644 aviacao.html  
cores> git log --oneline  
c786105 (HEAD → main) Revert "Adicionada pagina Aviação"  
1998e48 Adicionada pagina Aviação  
a9bc51d (tag: v1.0, origin/main, origin/HEAD) Adicionada navegação  
7f619a7 Textos em português  
85312e7 laranja e azul  
cores>
```



## Instantâneos Equivalentes



# Git básico (64)

Desfazendo alterações que já passara por um commit (12/13):

# Git básico (65)

Desfazendo alterações que já passara por um commit (13/13):

- ▶ Lembre-se: Especifique o commit que se quer desfazer –não o estável para o qual se deseja voltar!
- ▶ Pense como se estivesse dizendo: “Desfaça este commit” ao invés de “restaure esta versão”

# Git básico (66)

Desfazendo alterações que não sofreram “commit”:

- ▶ Crie um arquivo nada.html e deixe-o vazio
- ▶ À index.html acrescente um mais item de lista apontando para nada.html:

<li>

    <a href="nada.html">Uma página vazia</a>

</li>

# Git básico (67)

Desfazendo alterações que não sofreram “commit”:

## git status

Um arquivo rastreado e um não, que precisa ser tratado.

Primeiramente, cuidamos do index.html

```
cores> git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file> ..." to update what will be committed)
  (use "git restore <file> ..." to discard changes in working directory)
    modified:   index.html

Untracked files:
  (use "git add <file> ..." to include in what will be committed)
    .history/
    nada.html

no changes added to commit (use "git add" and/or "git commit -a")
cores>
```

# Git básico (68)

Desfazendo alterações que não sofreram “commit”:

## **git reset --hard**

Faz com que todos os arquivos rastreados estejam de acordo com o commit mais recente

Note que a flag **--hard** é o que realmente atualiza os arquivos

**git reset** sem essa flag simplesmente retira de stage o index.html

Em ambos os casos **git reset** opera somente no diretório corrente e área de stage.

O histórico não é afetado!

# Git básico (68)

Desfazendo alterações que não sofreram “commit”:

**git reset --hard**

```
Changes not staged for commit:  
  (use "git add <file>..." to update w  
  (use "git restore <file>..." to disc  
    modified:   index.html
```

```
Untracked files:  
  (use "git add <file>..." to include  
    .history/  
    nada.html
```

```
On branch main  
Your branch is ahead of 'origin/main'  
  (use "git push" to publish your loca
```

```
Untracked files:  
  (use "git add <file>..." to include  
    .history/  
    nada.html
```

# Git básico (69)

Desfazendo alterações que não sofreram “commit”:

- ▶ Eliminando o arquivo dummy.html
  - ▶ Obviamente poderia ser uma remoção manual, mas desta forma evitam-se erros humanos

## git clean -f

- ▶ Remove arquivos não rastreados

```
cores> git clean -f
Removing nada.html
cores> git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file> ..." to include in what will be committed)
    .history/

nothing added to commit but untracked files present (use "git add" to track)
cores>
```

# Git básico (70)

Desfazendo alterações que não sofreram “commit”:

- **Cuidado:** Tanto `git reset` quanto `git clean` operam no diretório de trabalho e não nos instantâneos que sofreram commit.
- Diferente do `git revert` eles operam mudanças permanentes, então tenha certeza que você quer jogar fora o conteúdo que está sendo afetado.

# Git básico (71)

Revisão:

- ▶ A maioria dos comandos Git operam em um dos três componentes:
  - ▶ O diretório de trabalho;
  - ▶ Os instantâneos em stage;
  - ▶ Os instantâneos que sofreram commit.

# Git básico (72)

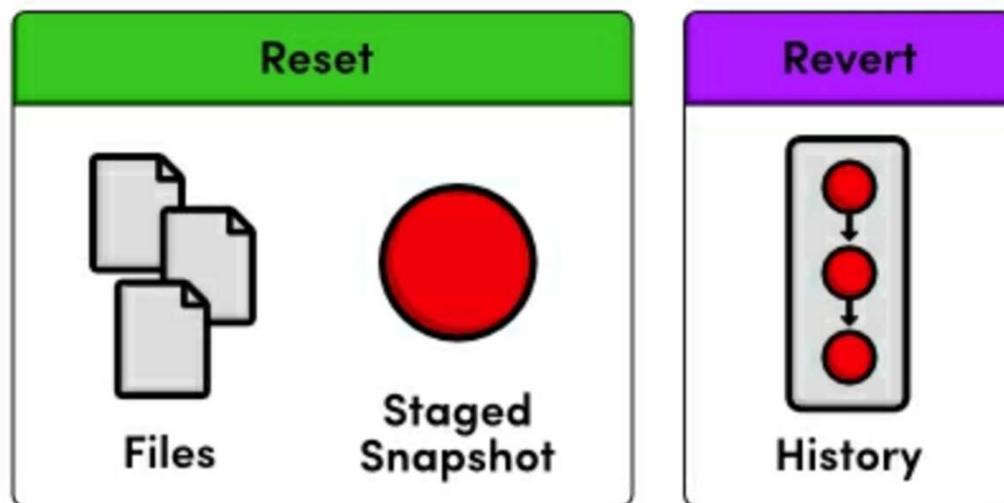
Revisão:

- ▶ git reset desfaz mudanças no diretório de trabalho e no instantâneo em stage;
- ▶ git revert desfaz as modificações contidas nos instantâneos que sofreram commit;
- ▶ git revert, ao invés de remover o commit, grava o commit para o caso que se queira ele de volta em outro momento.
- ▶ Essa é a única razão para preservar instantâneos que já sofreram commit

# Git básico (73)

Revisão:

**reset x revert**



 [Code](#)  [Issues](#)  [Pull requests](#)  [Actions](#)  [Projects](#)  [Wiki](#)  [Security](#)  [Insights](#)  [Settings](#)

 [main](#) 

 Commits on Jul 10, 2021

**Adicionada navegação**

 [sdifiore](#) committed 28 days ago

Unverified

 a9bc

**Textos em português**

 [sdifiore](#) committed 28 days ago

Unverified

 7f61

**laranja e azul**

 [sdifiore](#) committed 28 days ago

Unverified

 8531

# Git básico (74)

Quando e como usar o GitHub: Revisar commits é mais rápido e não interfere no código que está sendo entrado!

icio -> GoBack()

master (#1)

sdifiore committed on 26 Jun Unverified

1 parent bf5edbc commit 4e9812b17b

Showing 40 changed files with 399 additions and 296 deletions.

14 14 SeqLex/ClienteInfoSelect.xaml.cs

```
11 11 {
12 12     public partial class ClienteInfoSelect : Page
13 13 {
14 -    private ObservableCollection<Reclamante> _fichas;
14 +    private ObservableCollection<Reclamante> _reclamantes;
15 15     private readonly Status _status;
16 16     private readonly Usuario _usuario;
17 17
18 18     @@ -22,7 +22,7 @@ public ClienteInfoSelect(Usuario usuario, Status status)
19 19         var dbServices = new DBServices();
20 20         _usuario = usuario;
21 21         _status = Atualiza.Status(status, Local.ClienteInfoSelect);
22 -        //_fichas = dbServices.GetReclamantes();
22 +        //_reclamantes = dbServices.GetReclamantes();
23 23
24 24
25 25     private void AnteriorBtn_Click(object sender, RoutedEventArgs e)
26 26
27 27
28 28     private void nomeText_TextChanged(object sender, TextChangedEventArgs e)
29 29
30 30     private void nomeText_TextChanged(object sender, TextChangedEventArgs e)
31 31
32 32
33 33     private void nomeText_TextChanged(object sender, TextChangedEventArgs e)
34 34 {
35 -        nomeComboBox.ItemsSource = _fichas.Where(f => f.Nome.StartsWith(nomeText.Text));
35 +        nomeComboBox.ItemsSource = _reclamantes.Where(f => f.Nome.StartsWith(nomeText.Text));
36 36 }
37 37
38 38     private void btNomeSelect_Click(object sender, RoutedEventArgs e)
39 39
40 40     @@ -44,7 +44,7 @@ private void btNomeSelect_Click(object sender, RoutedEventArgs e)
```

# Git básico (75)

Quando e como usar o GitHub: Executar diff entre arquivos é imediato!

# Git básico (76)

Quando e como usar o GitHub: Modos **código** e **raw** facilitam compreender e copiar e colar trechos, sem precisar reversão!

## código

```
1  using System.Collections.Generic;
2  using System.Collections.ObjectModel;
3  using System.Linq;
4  using System.Threading.Tasks;
5
6  using Microsoft.EntityFrameworkCore;
7
8  using SeqLex.Model;
9
10 namespace SeqLex.Services
11 {
12     public class DBServices
13     {
14         private readonly DbContextOptions<DataContext> _dBService;
15
16         public string Truncate { get; private set; }
17
18         public DBServices()
19         {
20             _dBService = new DbContextOptionsBuilder<DataContext>()
21                 .UseSqlServer(Constant.dbString)
22                 .Options;
23         }
24
25         public ObservableCollection<Role> ListRoles()
26         {
27             using (var context = new DataContext(_dBService))
28             {
29                 var rolesEnum = context.Roles.Select(r => r);
30                 var roles = new ObservableCollection<Role>(rolesEnum);
31
32                 return roles;
33             }
34         }
35     }
36 }
```

## raw

```
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

using SeqLex.Model;

namespace SeqLex.Services
{
    public class DBServices
    {
        private readonly DbContextOptions<DataContext> _dBService;

        public string Truncate { get; private set; }

        public DBServices()
        {
            _dBService = new DbContextOptionsBuilder<DataContext>()
                .UseSqlServer(Constant.dbString)
                .Options;
        }

        public ObservableCollection<Role> ListRoles()
        {
            using (var context = new DataContext(_dBService))
            {
                var rolesEnum = context.Roles.Select(r => r);
                var roles = new ObservableCollection<Role>(rolesEnum);

                return roles;
            }
        }
    }
}
```

# Git básico (77)

Quando efetuar um commit?

1. Sempre que estiver encerrando um dia de trabalho
  - a. (Junto com um arquivo em algum lugar deixando exatamente claro o que estava fazendo quando terminou e quais os próximos passos – muito importante)
2. Sempre que encerrar uma tarefa específica, por exemplo a criação de uma classe ou um método
3. Sempre que iniciar o trabalho em alguma tarefa específica nova, se não efetuou o passo anterior

# VI: C# e .NET



# C# e .NET(1)

Este curso elegeu a linguagem C# a ser ministrada.

Desde seu surgimento vem constantemente sofrendo alterações que, a sua compreensão ajuda o desenvolvedor se manter atualizado.

# C# e .NET(2)

## Bibliografia (1/2):

- Albahari, J. (2021). C# 9.0 in a Nutshell - The Definite Reference. O'Reilly Media.
- Esposito, A. S. (2009). Microsoft .NET Architecting Applications for the Enterprise. Microsoft Press.
- Hu, Y. (2020). Fácil Aprendizagem Estruturas De Dados e Algoritmos C#.
- Metzgar, S. b. (2017). Exploring .NET Core with Microservices, ASP.NET Core, and Entity Framework Core. Manning Publications.
- Price, M. J. (2020). C# 9 and .NET 5 – Modern Cross-Platform Development. Kindle Edition. Packt Publishing.
- Ravindra Akella, et al. (2021). Enterprise Application Development with C# 9 and .NET 5. Packt Publishing.

# C# e .NET(3)

## Bibliografia (2/2):

- Shanu, S. (s.d.). Beginning C# Object Oriented Programming. C# Corner.
- Sharp, D. (2019). Mastering C#: The Ultimate Beginner's And Intermediate's Guide to Learn C# Programming.
- Sharp, J. (2018). Microsoft Visual C# Step by Step - Ninth Edition. Published with the authorization of Microsoft Corporation by: Pearson.
- Skeet, J. (2019). C# in Depth, Fouth Edition. M A NManning Publications.
- White, M. B. (2019). Mastering C# (C Sharp Programming). Newstone.
- Zeeshan, A. A. (2016). OOPs Principle and Theory. C# Corner.

# C# e .NET(4)

Diferença entre C# e .NET (1/4):

- ▶ .NET é ao mesmo tempo uma biblioteca e um ambiente de execução para os sistemas operacionais: Windows, macOS, Linux, Android e iOS em arquiteturas x86, x64, ARM32 e ARM64.
- ▶ Um programa escrito em C# roda dentro de um ambiente .NET, que provê serviços como o gerenciamento de memória ou bibliotecas de funções.

# C# e .NET(5)

Diferença entre C# e .NET (2/4):

- ▶ O código C# é compilado, para a chamada IL (Linguagem intermediária);
- ▶ A IL independe de plataforma, o que torna possível um programa C# em diferentes hardware e sistemas operacionais;
- ▶ Para isso é necessário o CLR onde o aplicativo C# é executado, compilando o IL no código da máquina nativa.
- ▶ Também chamado de Compilação Just-in-time (JIT).

# C# e .NET(6)

Diferença entre C# e .NET (3/4):

- ▶ O conhecimento de C# implica no conhecimento de .NET pois que, o modelo de objetos de do C# corresponde ao modelo de objetos do .NET e
- ▶ Muito do que se pode fazer com C# está no uso de bibliotecas de funções .NET

# C# e .NET(7)

Diferença entre C# e .NET (4/4):

- ▶ O oposto não é necessário, isto é:
- ▶ Sistemas e linguagens como o F# e o VB que usam o ambiente e bibliotecas .NET.

# C# e .NET(8)

NET Framework , .NET Core e Xamarin(1/3):

► Plataformas  
relacionadas e  
sobrepostas

# C# e .NET(9)

NET Framework , .NET Core e Xamarin(2/3):

## ►.NET Framework:

- É a plataforma original para Windows que está se encerrando na atual versão 7.82
- Exclusiva para rodar em sistema operacional Windows
- Só deve ser usada para dar manutenção em sistemas já existentes

# C# e .NET(10)

NET Framework , .NET Core e Xamarin(2/3):

## ►.NET Core:

- ▶ Recente, é multiplataforma, isto é, roda em Windows, Linux e MacOS
- ▶ Requer só a sua instalação na plataforma desejada
- ▶ Deve ser usada para o desenvolvimento de novos sistemas

# C# e .NET(11)

NET Framework , .NET Core e Xamarin(3/3):

## ►Xamarin:

- Originalmente um desenvolvimento de terceiros em um projeto equivalente ao .NET Framework, denominado Mono
- O Mono foi empregado na plataforma de jogos Unity e na Mobile Xamarin
- Em 2016 foi adquirida pela Microsoft onde passou a ser a plataforma oficial de desenvolvimento mobile

# C# e .NET(12)

O que é possível desenvolver com cada uma delas:

- ▶ Windows:
  - ▶ WindowsForm
  - ▶ WPF: Windows Presentation Foundation
    - ▶ Desktop e IoT
  - ▶ UWP: Universal Windows Platform
    - ▶ Desktop e IoT
- ▶ Windows, Linux Mac OS
  - ▶ Microserviços
  - ▶ Console
  - ▶ ASP.NET
  - ▶ WEB
  - ▶ Xamarin

# C# e .NET(13)

Resumo:

Implementação .NET	Tipos de aplicações
.NET Framework	WinForms, WPF, Windows e ASP.NET Framework
.NET Core e .NET 6	WPF, e ASP.NET, Windows, ASP.NET Core, Web, IoT, REST-API
Mono / Xamarin	Android e iOS nativos, macOS, tvOS e watchOS
UWP	Windows em PC's, tablets e Xbox

# C# e .NET(14)

Tipos de aplicações:

Tipo de aplicação	Descrição
Web	Baseados em Web, REST-API
Microserviços	Independentes de deploy, altamente escaláveis, usam Docker
Nuvem	Funções Servlets e nativas
Mobile	Nativos para iOS, Android e Windows
Desktop	WinForms e WPF para Windows, Console para Lixu, MacOS e Windows
IoT	Raspberry Pi e HummingBoard
Games	2D e 3D para Windows, Android e iOS, Unity;
Machine Learning	Modelos do usuário e integráveis com .NET; ML.NET

# C# e .NET(15)

Versões e suporte ao .NET (1/3):

- LTS - Long Term Support:
  - As releases são estáveis e requerem pouca manutenção;
  - O suporte será mantido por cerca de três anos após a disponibilidade geral.

# C# e .NET(16)

Versões e suporte ao .NET (2/3):

## ►Corrente:

- As releases terão características mutáveis em função de sugestões;
- O suporte às versões menores será mantido por até de três meses.

# C# e .NET(17)

Versões e suporte ao .NET (3/3):

## ► LTS ou Corrente:

- A corrente é ideal para desenvolvimento de novos projetos, pois sempre se terá as últimas características possíveis;
- O software entregue em produção é recomendado ser uma versão LTS o que diminui as manutenções.

## Frameworks (1):

► **ASP.NET Core** : Esta é uma estrutura de desenvolvimento de aplicativos de código aberto e multiplataforma que permite construir aplicativos modernos, baseados na nuvem, conectados à Internet, como aplicativos web, IoT e API. ASP.NET Core é construído em cima do .NET Core, portanto, você pode construir e executar em plataformas como Linux, macOS e Windows.

## Frameworks (2):

### ► **WPF:**

Esta é uma estrutura de interface do usuário que permite criar aplicativos de desktop para Windows, que usa o *Extensible Application Markup Language (XAML)*, um modelo declarativo para o desenvolvimento de aplicativos.

## Frameworks (3):

- ***Entity Framework*** (EF) **Core**: Estrutura de mapeamento (código aberto, multiplataforma), de objetos relacionais (ORM) para trabalhar com bancos de dados usando objetos .NET.
- Suporta consultas LINQ, rastreamento de alterações e migrações de esquema;
- Bancos de dados populares como SQL Server, SQL Azure, SQLite, Azure Cosmos DB, MySQL e outros; mais.

## Frameworks (4):

### ► *Linguagem por Consulta Integrada (LINQ):*

Recursos de consulta às linguagens de programação.NET.

- Permite consultar dados de um banco de dados, XML, matrizes na memória e coleções com a mesma API.

# Programação C# básica (1)

- ▶ Crie um diretório de trabalho e abra o Visual Studio Code nele;
- ▶ Abra o terminal PowerShell no VS Code ou uma janela aparte, mas sempre apontando para o mesmo diretório
- ▶ O que faz o do VS Code ser mais conveniente

# Programação C# básica (2)

- ▶ Crie o esqueleto da solução:

**dotnet new console**

- ▶ Esse comando cria toda a estrutura básica para uma aplicação de console:
- ▶ Dois arquivos: **<nome da pasta>.csproj** e **Program.cs**
- ▶ No exemplo, como a pasta se chama “Basico” o .csproj chama-se **Basico.csproj**

The screenshot shows the Microsoft Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Area:** Shows the `Basico.csproj` file content:

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <OutputType>Exe</OutputType>
5     <TargetFramework>net5.0</TargetFramework>
6   </PropertyGroup>
7
8 </Project>
```
- Sidebar:** Explorer, Open Editors, BASICO, Program.cs.
- Bottom Status Bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, COMMENTS, atec\Turma\App Web de A-Z\Exemplos\C#\Basico\Basico.csproj..., Determining projects to restore..., Restored C:\Users\difio\OneDrive - Fatec Centro Paula Souza\basico\Turma\App Web de A-Z\Exemplos\C#\Basico\Basico.csproj (in 352 ms). Restore succeeded., Basico>, [Aa] Ln 1, Col 1 Spaces: 2 UTF-8 CRL

# Programação C# básica (3)

Basico.csproj

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net5.0</TargetFramework>
  </PropertyGroup>
</Project>
```

Arquivo XML que informa ao compilador (Roselyn) que o projeto usa:

- SDK Microsoft.Net
- O arquivo resultado da compilação deverá ser um .Exe
- A versão .NET utilizada é a 5.0
  - Lembre-se que a última Mainframe era a 4.8.2, portanto essa já é Core!

# C# e .NET(18)

Versões anteriores de sdk e runtime:

- Listagem das versões instaladas:

**dotnet --list-sdks**

**dotnet --list-runtimes**

# Programação C# básica (4)

Program.cs

```
using System;

namespace Basico
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

# Programação C# básica (5)

Note que cada abrir e fechar de chaves delimita uma região:

```
using System;

namespace Basico
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

# Programação C# básica (6)

Em um primeiro momento basta entender que a primeira coisa que um programa, para console em C# executa é o que está dentro do “método” Main() que esta dentro da classe Program:

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Qualquer programa inicia executando o que está aqui  
        }  
    }  
}
```

# Programação C# básica (7)

Executando o programa:

► No Powershell do VS Code digite:

**dotnet run**

```
Basico> dotnet run
Hello World!
Basico> █
```

# Programação C# básica (7)

Executando o programa:

```
{  
    Console.WriteLine("Hello World!");  
}
```

```
Basico> dotnet run  
Hello World!  
Basico> █
```

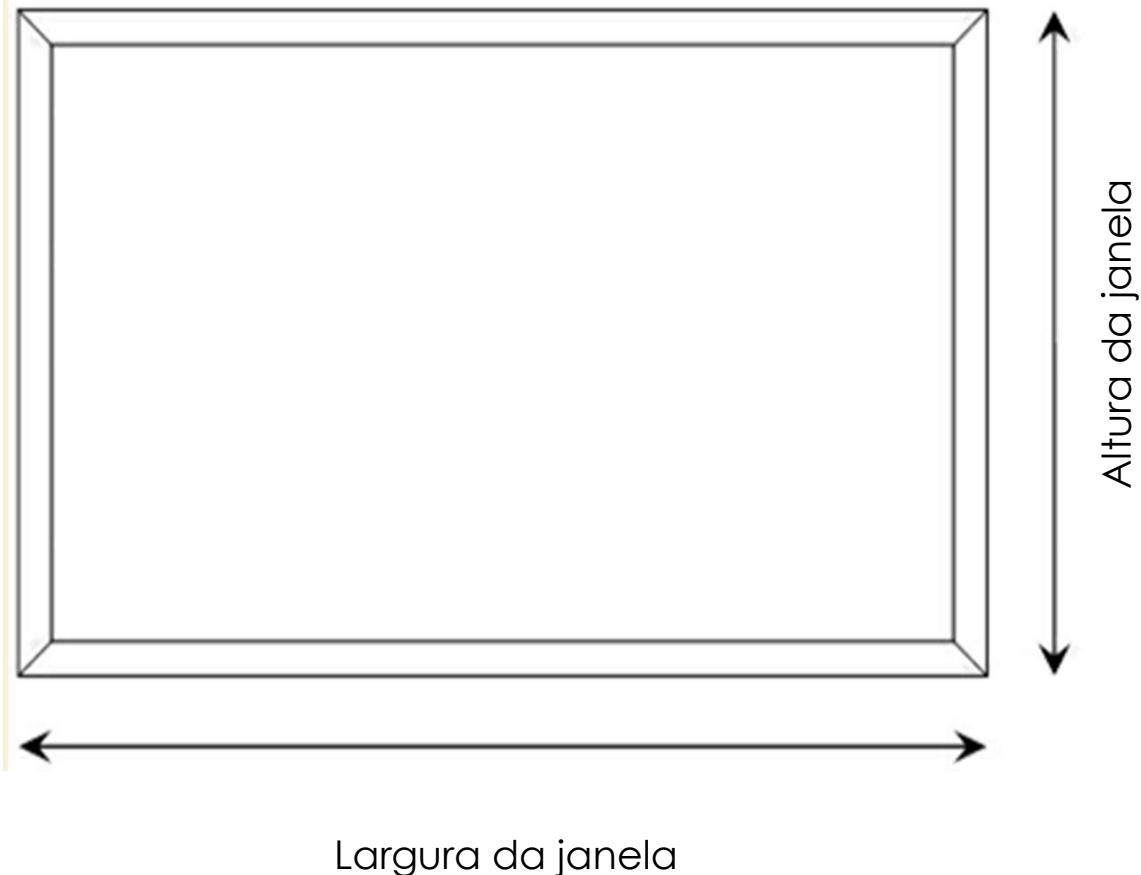
# Programação C# básica (8)

Analisando Program.cs:

```
using System;

namespace Revisao
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

- A execução será iniciada pelo que estiver contido em `Program`. `Main()`
- Neste primeiro momento podemos ignorar os modificadores `static void`
- `Console.WriteLine("Hello World!")` escreve na console o texto – entre aspas – Hello Word!
- `string[] args` serve para a passagem de argumentos, raramente usado
- Preste atenção no `namespace Revisao`



# Programação C# básica (9)

1º Programa área  
envidraçada e  
comprimento de batentes

# Programação C# básica (10)

1º Programa área  
envidraçada e  
comprimento de  
batentes

Dirija-se ao diretório onde quer residir a nova aplicação e  
abra uma console PowerShell;

Digite: **dotnet new console --name VitroCalc**

```
C#> dotnet new console --name VitroCalc
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on VitroCalc\VitroCalc.csproj...
Determining projects to restore...
Restored C:\Users\difio\OneDrive - Fatec Centro Paula Souza\estudos\Fatec\Turma\App Web de A-Z\Exemplos\C#\VitroCalc\VitroCalc.csproj (in 235 ms).
Restore succeeded.

C#> dir

Directory: C:\Users\difio\OneDrive - Fatec Centro Paula Souza\estudos\Fatec\Turma\App Web de A-Z\Exemplos\C#

Mode          LastWriteTime        Length Name
----          -----          ----
la---        08/08/2021    14:54
d---        14/08/2021    07:28
                           Basico
                           VitroCalc

C#>
```

Abra o VS Code em VitroCalc

## Programação C# básica (11)

1º Programa  
área  
envidraçada e  
comprimento  
de batentes

```
1  using System;
2
3  namespace VitroCalc
4  {
5      class Program
6      {
7          static void Main()
8          {
9              double largura, altura, comprimentoMadeira, areaVidro;
10             string larguraString, alturaString;
11
12             Console.WriteLine("Digite a largura da janela em metros:\n");
13             larguraString = Console.ReadLine();
14             largura = double.Parse(larguraString);
15
16             Console.WriteLine("Digite a altura da janela em metros:\n");
17             alturaString = Console.ReadLine();
18             altura = double.Parse(alturaString);
19
20             comprimentoMadeira = 2 * (largura + altura) * 3.25;
21
22             areaVidro = 2 * (largura * altura);
23
24             Console.WriteLine($"O comprimento da madeira é { comprimentoMadeira }.\n"
25             | | | | + $"A área envidraçada é { areaVidro } pés quadrados");
26         }
27     }
28 }
```

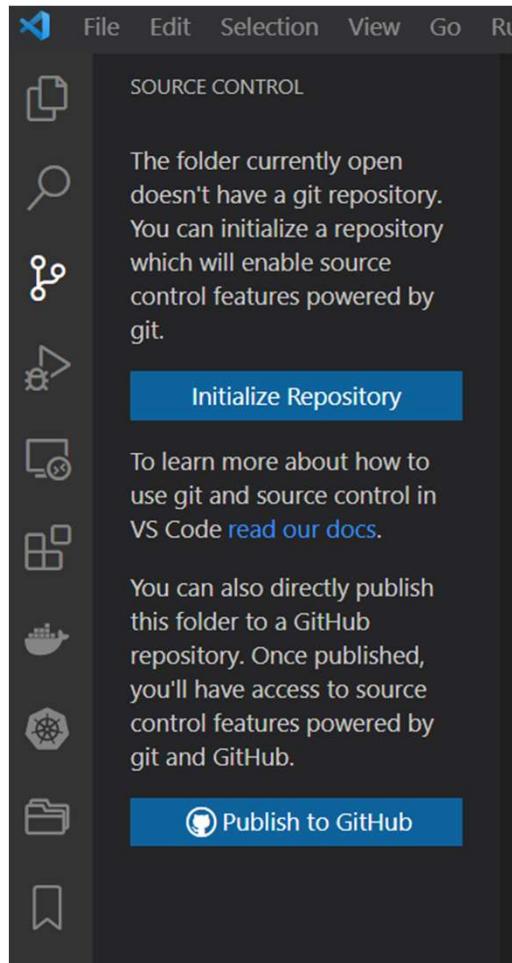
# Programação C# básica (12)

Alguns tipos de dados

<b>Tipo</b>	<b>Tamanho</b>	<b>Alcance</b>
sbyte	8 bits	-128 a 127
byte	8 bits	0 a 255
short	16 bits	-32.768 a 32.767
ushort	16 bits	0 a 65.535
int	32 bits	-2.147.483.648 a 2.147.483.647
uint	32 bits	0 a 4.294.967.295
long	64 bits	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
ulong	64 bits	0 a 18.446.744.073.709.551.615
Char	16 bits	0 a 65.535

# Git básico (78)

## Git e GitHub direto do Visual Studio Code



# Git básico (79)

Git e GitHub direto do Visual Studio Code



## Authorize Visual Studio Code to access GitHub

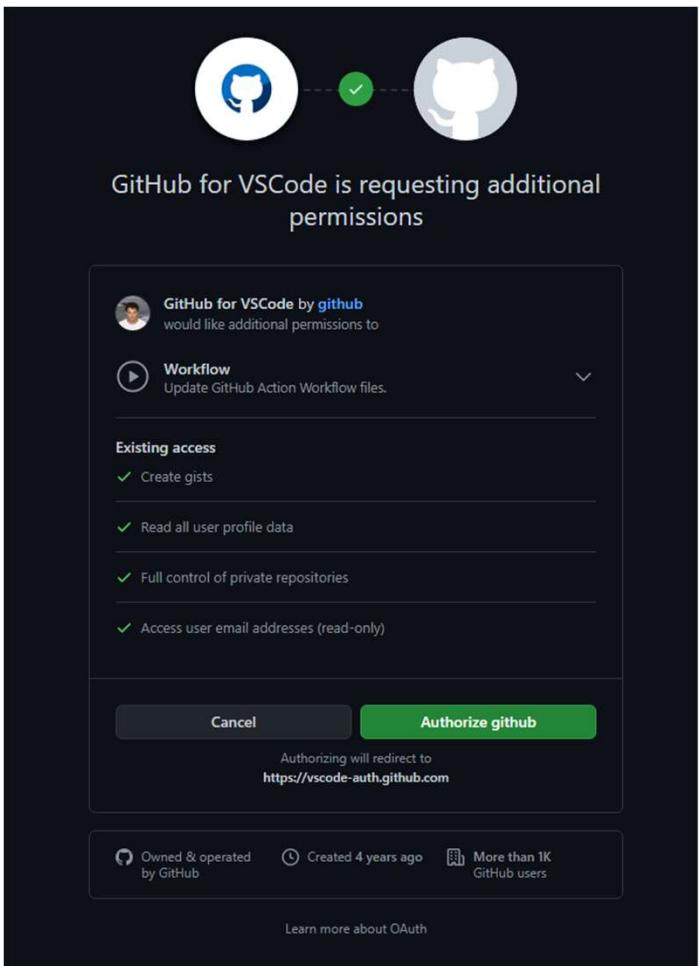
If you initiated this authorization from Visual Studio Code, click 'Continue' to authorize access to GitHub

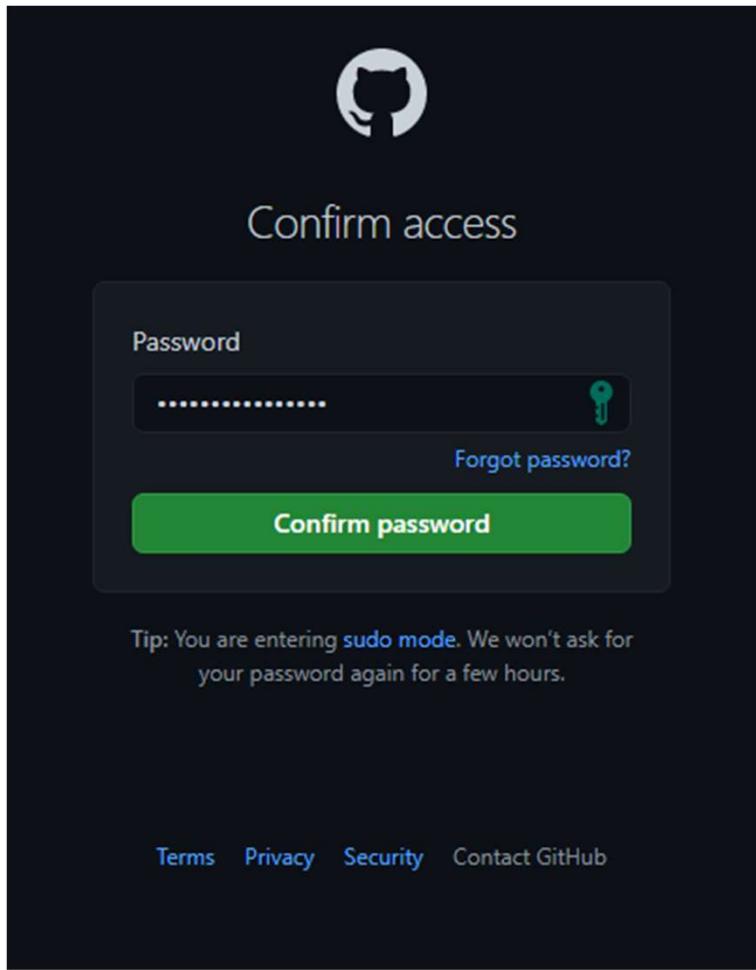
[Continue](#)

[Do not authorize](#)

# Git básico (80)

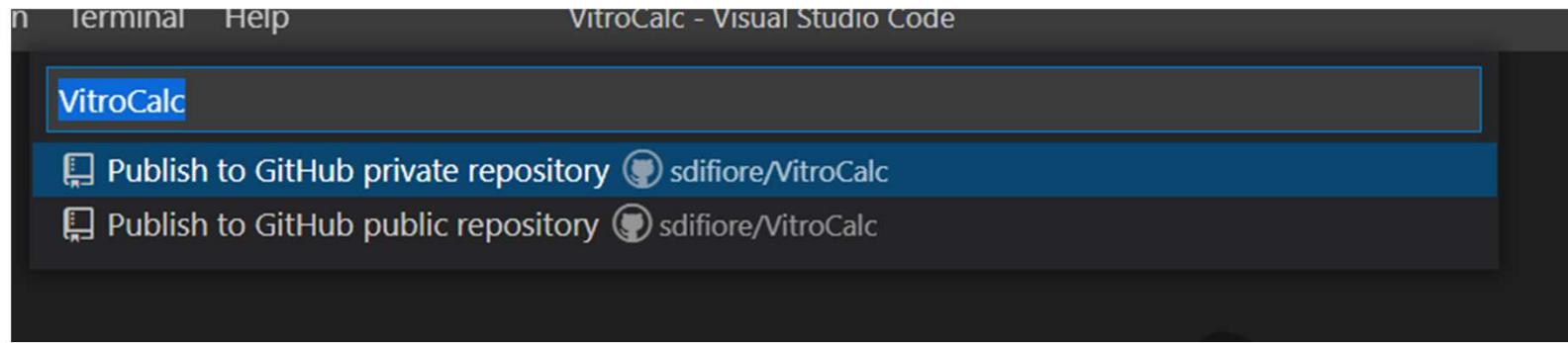
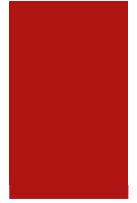
## Git e GitHub direto do Visual Studio Code





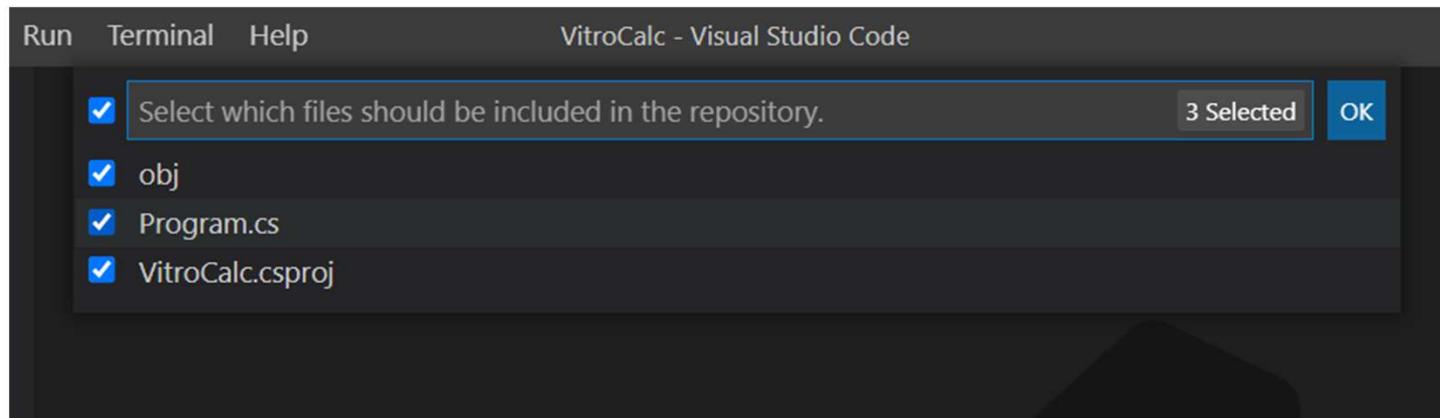
# Git básico (81)

## Git e GitHub direto do Visual Studio Code



# Git básico (82)

## Git e GitHub direto do Visual Studio Code



# Git básico (83)

## Git e GitHub direto do Visual Studio Code

# Programação C# básica (13)

## 2º Programa área envidraçada e comprimento de batentes

```
using System;
namespace VitroCalc
{
    class Program
    {
        static void Main()
        {
            float largura, comprimento, comprimentoMadeira, areaEnvidraca;
            const float larguraMax = 5.0f;
            const float larguraMin = 0.5f;
            const float comprimentoMax = 3.0f;
            const float comprimentoMin = 0.75f;

            string larguraString, comprimentoString;

            Console.WriteLine("Forneça a largura da janela: ");
            larguraString = Console.ReadLine();
            largura = float.Parse(larguraString);

            if (largura < larguraMin)
            {
                Console.WriteLine("Largura é muito pequena.\n\n");
                Console.WriteLine("Usando o mínimo");
                largura = larguraMin;
            }

            if (largura > larguraMax)
            {
                Console.WriteLine("Largura é muito grande.\n\n");
                Console.WriteLine("Usando o máximo");
                largura = larguraMax;
            }

            Console.WriteLine("Forneça o comprimento da janela: ");
            comprimentoString = Console.ReadLine();
            comprimento = float.Parse(comprimentoString);

            if (comprimento < comprimentoMin)
            {
                Console.WriteLine("O comprimento é muito pequeno.\n\n");
                Console.WriteLine("Usando o mínimo");
                comprimento = comprimentoMin;
            }

            if (comprimento > comprimentoMax)
            {
                Console.WriteLine("O comprimento é muito grande.\n\n");
                Console.WriteLine("Usando o máximo");
                comprimento = comprimentoMax;
            }

            comprimentoMadeira = (2.0f * (largura + comprimento)) + (0.25f);
            areaEnvidraca = 2 * (largura * comprimento);

            Console.WriteLine($"O comprimento da madeira é { comprimentoMadeira } pés");
            Console.WriteLine($"A área envidraca é { areaEnvidraca } metros quadrados");
        }
    }
}
```

```
C# Program.cs
1  using System;
2
3  namespace VitroCalc
4  {
5      class Program
6      {
7          static void Main()
8          {
9              float largura, comprimento, comprimentoMadeira, areaEnvidraca;
10
11             const float larguraMax = 5.0f;
12             const float larguraMin = 0.5f;
13             const float comprimentoMax = 3.0f;
14             const float comprimentoMin = 0.75f;
15
16             string larguraString, comprimentoString;
17
18             Console.Write("Forneça a largura da janela: ");
19             larguraString = Console.ReadLine();
20             largura = float.Parse(larguraString);
```

# Programação C# básica (14)

## **2º Programa área envidraçada e comprimento de batentes**

```
using System;
namespace VitroCalc
{
    class Program
    {
        static void Main()
        {
            float largura, comprimento, comprimentoMadeira, areaEnvadrada;
            const float larguraMin = 5.0f;
            const float larguraMax = 10.0f;
            const float comprimentoMax = 8.0f;
            const float comprimentoMadeira = 6.75f;

            string larguraString, comprimentoString;

            Console.WriteLine("Forneça a largura da janela: ");
            larguraString = Console.ReadLine();
            largura = float.Parse(larguraString);

            if (largura < larguraMin)
            {
                Console.WriteLine("Largura é muito pequena.\n\n");
                Console.WriteLine("Usando o mínimo");
                largura = larguraMin;
            }

            if (largura > larguraMax)
            {
                Console.WriteLine("Largura é muito grande.\n\n");
                Console.WriteLine("Usando o máximo");
                largura = larguraMax;
            }

            Console.WriteLine("Forneça o comprimento da janela: ");
            comprimentoString = Console.ReadLine();
            comprimento = float.Parse(comprimentoString);

            if (comprimento < comprimentoMin)
            {
                Console.WriteLine("O comprimento é muito pequeno.\n\n");
                Console.WriteLine("Usando o mínimo");
                comprimento = comprimentoMin;
            }

            if (comprimento > comprimentoMax)
            {
                Console.WriteLine("O comprimento é muito grande.\n\n");
                Console.WriteLine("Usando o máximo");
                comprimento = comprimentoMax;
            }

            comprimentoMadeira = (2.0f * (largura + comprimento) * 3.25f);

            areaEnvadrada = 2 * (largura * comprimento);

            Console.WriteLine($"O comprimento da madeira é {comprimentoMadeira} pés");
            Console.WriteLine($"A área envadrada é {areaEnvadrada} metros quadrados");
        }
    }
}
```

```
if (largura < larguraMin)
{
    Console.WriteLine("Largura é muito pequena.\n\n");
    Console.WriteLine("Usando o mínimo");
    largura = larguraMin;
}

if (largura > larguraMax)
{
    Console.WriteLine("Largura é muito grande.\n\n");
    Console.WriteLine("Usando o máximo");
    largura = larguraMax;
}

Console.Write("Forneça o comprimento da janela: ");
comprimentoString = Console.ReadLine();
comprimento = float.Parse(comprimentoString);
```

# Programação C# básica (15)

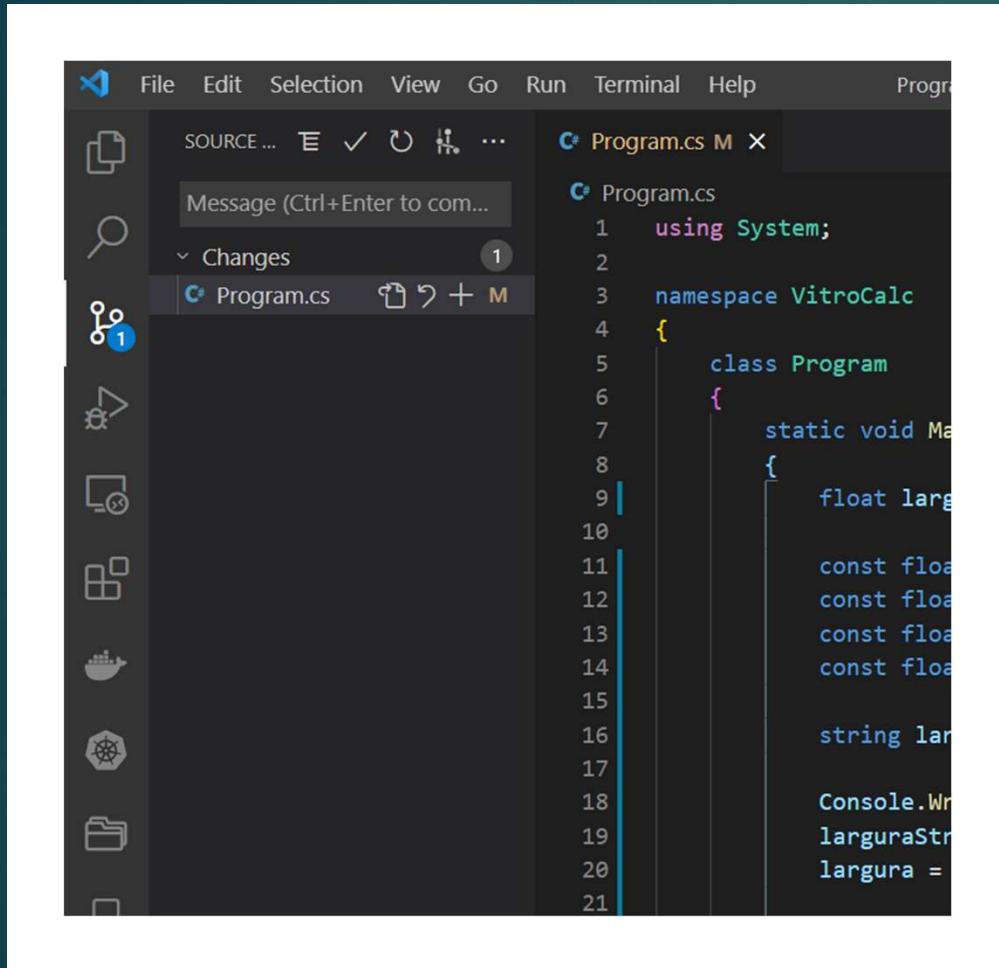
## 2º Programa área envidraçada e comprimento de batentes

```
1  using System;
2
3  namespace VetroCalc
4  {
5      class Program
6      {
7          static void Main()
8          {
9              float largura, comprimento, comprimentoMadeira, areaEnvidracada;
10
11             const float larguraMax = 5.0f;
12             const float larguraMin = 0.5f;
13             const float comprimentoOkMax = 3.0f;
14             const float comprimentoOkMin = 0.75f;
15
16             string larguraString, comprimentoString;
17
18             Console.WriteLine("Digite a largura da janela: ");
19             larguraString = Console.ReadLine();
20             largura = float.Parse(larguraString);
21
22             if (largura < larguraMin)
23             {
24                 Console.WriteLine("Largura é muito pequena.\n\n");
25                 Console.WriteLine("Usando o mínimo");
26                 largura = larguraMin;
27             }
28
29             if (largura > larguraMax)
30             {
31                 Console.WriteLine("Largura é muito grande.\n\n");
32                 Console.WriteLine("Usando o máximo");
33                 largura = larguraMax;
34             }
35
36             Console.Write("Força o comprimento da madeira: ");
37             comprimentoString = Console.ReadLine();
38             comprimento = float.Parse(comprimentoString);
39
40             if (comprimento < comprimentoOkMin)
41             {
42                 Console.WriteLine("O comprimento é muito pequeno.\n\n");
43                 Console.WriteLine("Usando o mínimo");
44                 comprimento = comprimentoMin;
45             }
46
47             if (comprimento > comprimentoOkMax)
48             {
49                 Console.WriteLine("O comprimento é muito grande.\n\n");
50                 Console.WriteLine("Usando o máximo");
51                 comprimento = comprimentoMax;
52             }
53
54             comprimentoMadeira = (2.0f * (largura + comprimento) * 3.25f);
55
56             areaEnvidracada = 2 * (largura * comprimento);
57
58             Console.WriteLine($"O comprimento da madeira é { comprimentoMadeira } pés");
59             Console.WriteLine($"A área envidraçada é { areaEnvidracada } metros quadrados");
60
61         }
62     }
63 }
```

# Git básico (84)

## Git e GitHub direto do Visual Studio Code

stage changes



# Git básico (85)

## Git e GitHub direto do Visual Studio Code

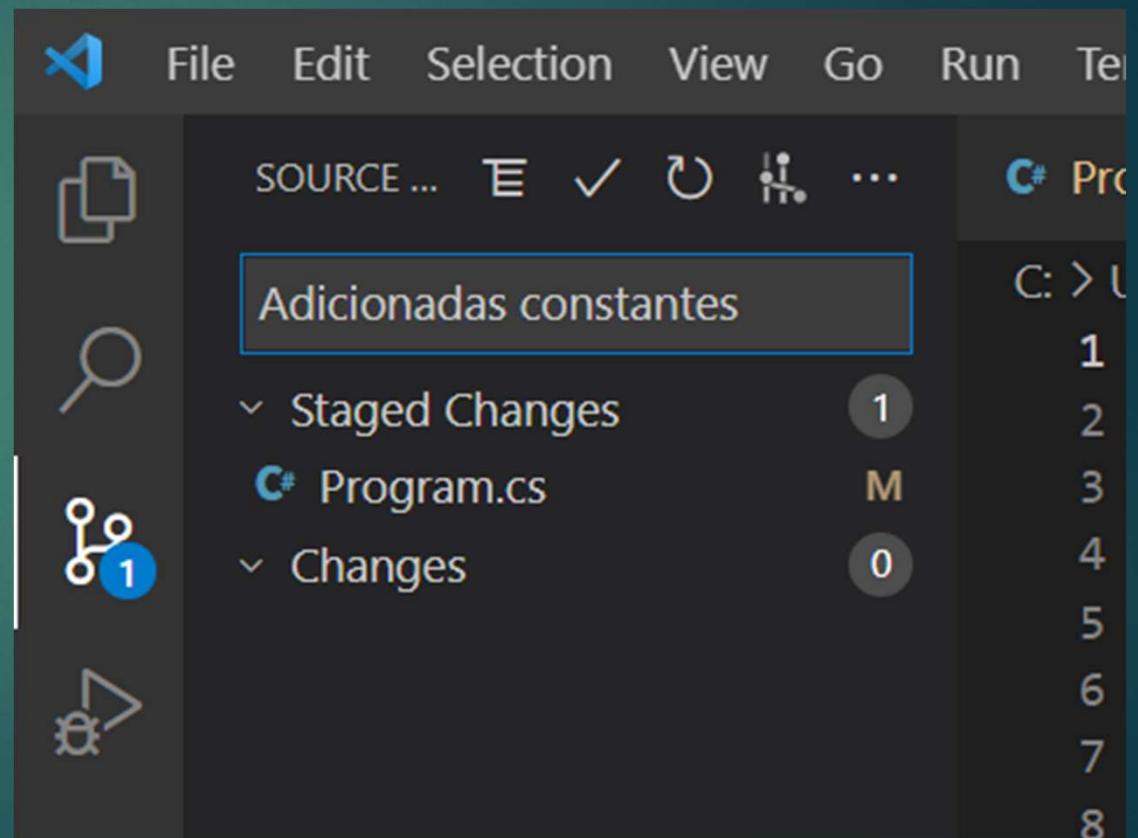
# diff

# Git básico (86)

Git e GitHub direto do Visual Studio Code

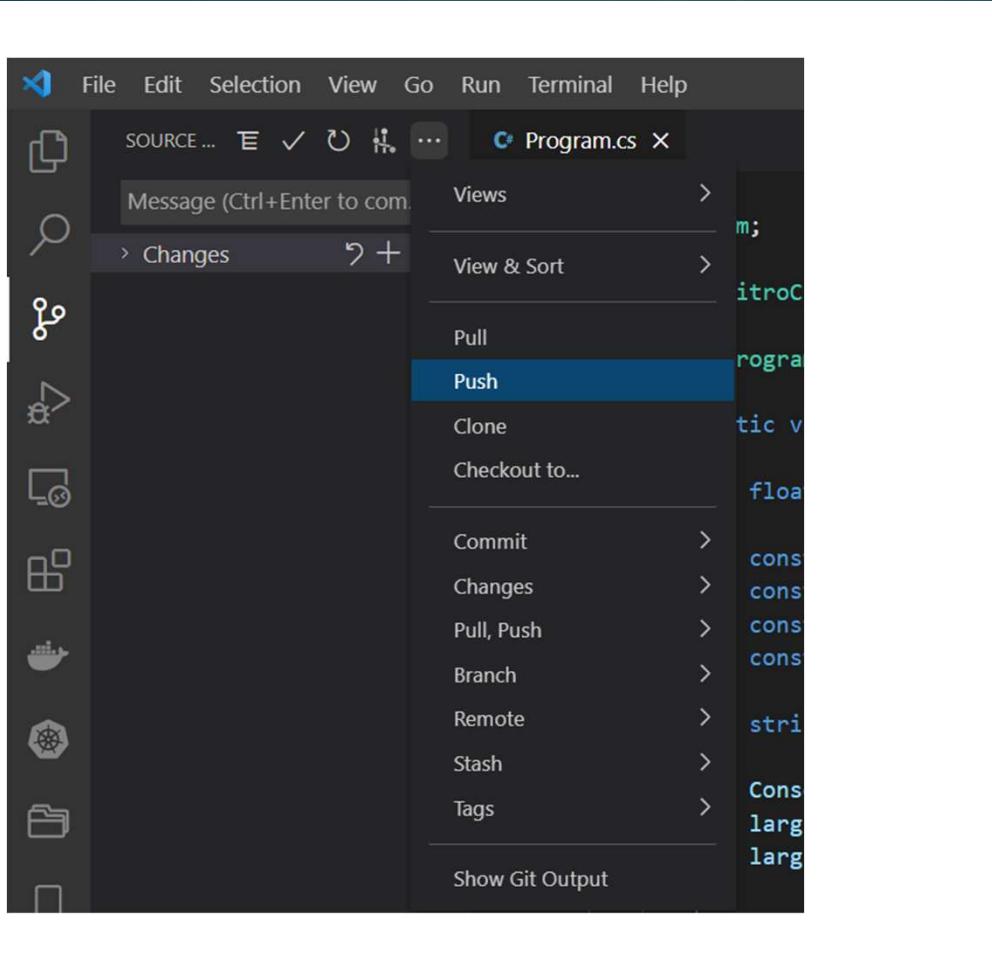
Mensagem e

# commit



# Git básico (87)

push

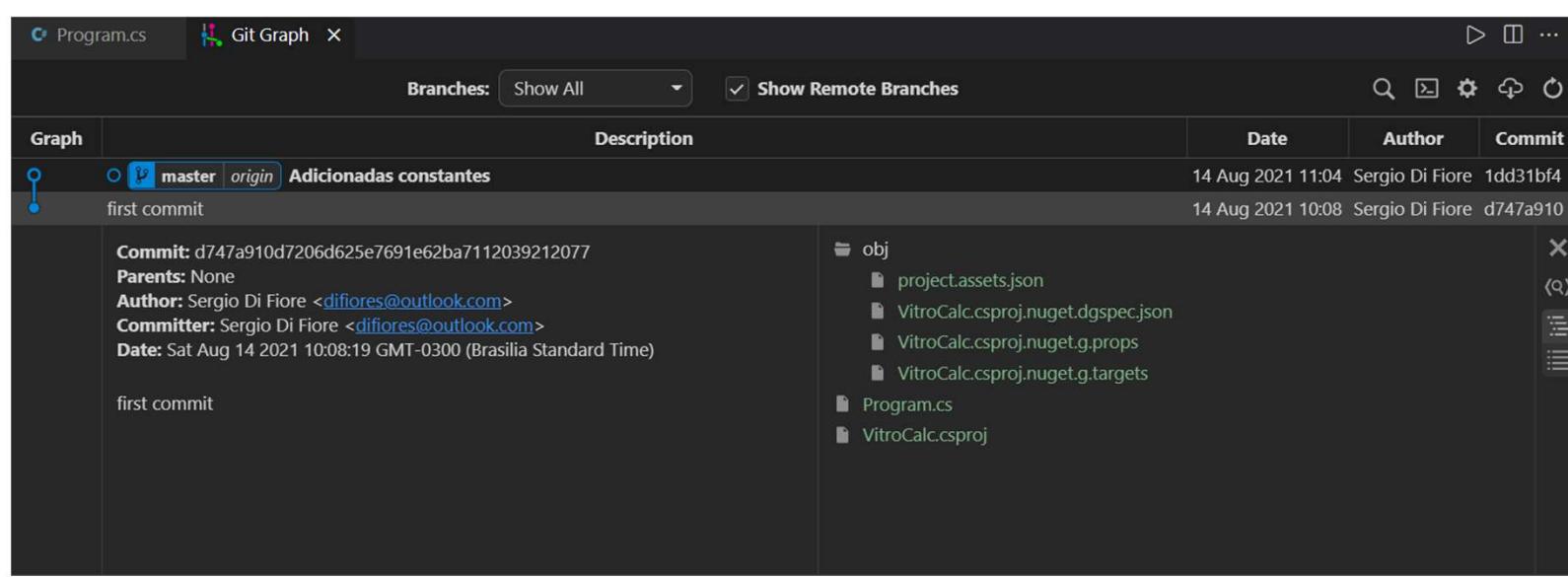


# Git básico (88)

Explore o github

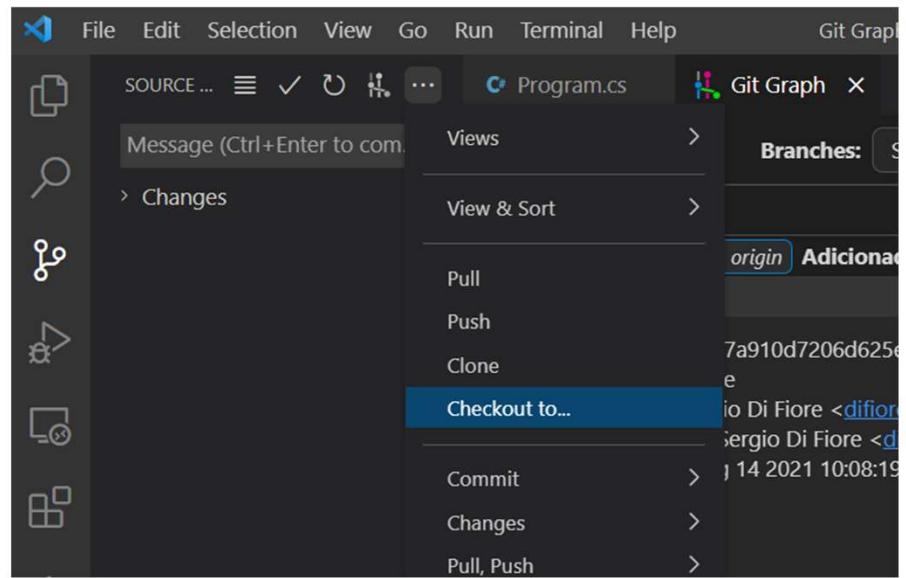
The screenshot shows a GitHub repository page with the following details:

- Repository Header:** Shows tabs for Projects, Wiki, Security, Insights, Settings, and a prominent green **Code** button.
- Commit History:** A list of recent commits:
  - 1dd31bf 8 minutes ago (2 commits)
  - first commit 1 hour ago
  - Adicionadas constantes 8 minutes ago
  - first commit 1 hour ago
- README Section:** A dark blue box with the text "Understand your project by adding a README." and a green "Add a README" button.
- Right Sidebar:**
  - About:** No description, website, or topics provided.
  - Releases:** No releases published. Create a new release.
  - Packages:** No packages published. Publish your first package.
  - Languages:** C# 100.0% (represented by a long green bar).
- Footer:** Links to Security, Status, Docs, Contact GitHub, Pricing, API, Training, and Blog.



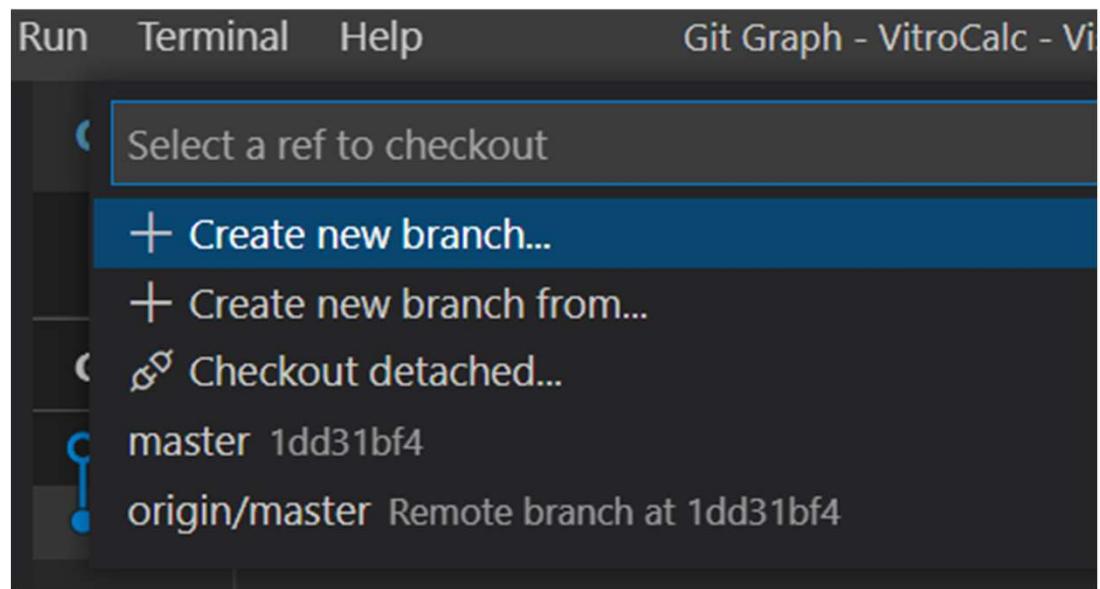
# Git básico (89)

EXTENSÃO RECOMENDADA: GIT GRAPH



# Git básico (90)

## checkout (1/2)



# Git básico (90)

## checkout (2/2)

```
const float comprimentoMin = 0.75f; // Este é um comentário de uma linha

/*
Este é um
comentário que
usa várias linhas
*/

string larguraString, comprimentoString;

Console.WriteLine("Forneça a largura da janela: ");
```

# Programação C# básica (16)

## Comentários

# Programação C# básica (17)

Designando valores a variáveis:

```
1  using System;
2
3  class Assignment
4  {
5      static void Main()
6      {
7          int primeiro, segundo, terceiro;
8
9          primeiro = 1;
10         segundo = 2;
11
12         terceiro = segundo + primeiro;
13
14         Console.WriteLine(primeiro);
15         Console.WriteLine(segundo);
16         Console.WriteLine(terceiro);
17     }
18 }
```

# Programação C# básica (18) Casting

```
1  using System;
2
3  class CastDemo
4  {
5      static void Main()
6      {
7          int i = 3, j = 2;
8          float fracao;
9          double duplo;
10
11         fracao = (float)i / (float)j;
12         duplo = fracao;
13
14         Console.WriteLine($"fração : {fracao}");
15     }
16 }
17 }
```

# Programação C# básica (19)

## Loops infinitos

```
1  using System;
2
3  namespace LoopInfinito
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              do
10                 Console.WriteLine("Não é a mamãe!");
11             while (true);
12         }
13     }
14 }
```

# Programação C# básica (21) VitroCalc Completo

```
1  using System;
2
3  namespace VitroCalc
4  {
5      class Program
6      {
7          static void Main()
8          {
9              double largura, altura, comprimentoMadeira, areaVidro;
10             const double comprimentoMaximo = 5.0;
11             const double comprimentoMinimo = 0.5;
12             const double alturaMaxima = 3.0;
13             const double alturaMinima = 0.75;
14             string larguraString, alturaString;
15
16             do
17             {
18                 Console.Write("Forneça a largura da janela entre " + comprimentoMinimo + " e " + comprimentoMaximo + " :");
19                 larguraString = Console.ReadLine();
20                 largura = double.Parse(larguraString);
21             } while (largura < comprimentoMinimo || largura > comprimentoMaximo);
22
23             do
24             {
25                 Console.Write("Forneça a altura da janela entre " + alturaMinima + " e " + alturaMaxima + " :");
26                 alturaString = Console.ReadLine();
27                 altura = double.Parse(alturaString);
28             } while (altura < alturaMinima || altura > alturaMaxima);
29
30             comprimentoMadeira = 2 * (largura + altura) * 3.25;
31             areaVidro = 2 * largura * altura;
32
33             Console.WriteLine($"O comprimento da madeira é { comprimentoMadeira } pés.\n"
34             | | | | + $"A área envidraçada é { areaVidro } metros quadrados");
35         }
36     }
37 }
```



# VII: Orientação a objetos

# Orientação a objetos básica (1)

O que é a orientação a objeto?

OOP é a **Representação**  
em uma **linguagem** de  
**programação** de algo **real**

# Orientação a objetos básica (2)

Conceitos

**Classe**

**PROJETO** de um objeto

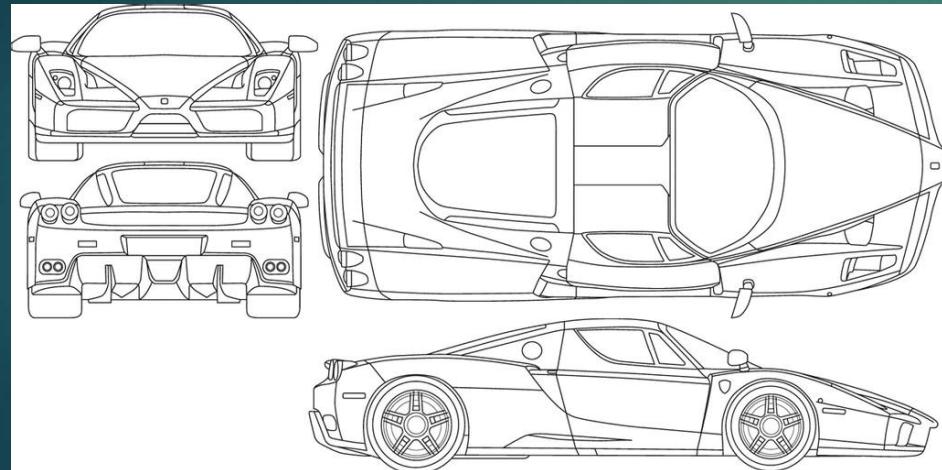
**Instância**

**OBJETO** à partir da classe

# Orientação a objetos básica (3)

## Classe x Instância

Classe



Instância



# Orientação a objetos básica (4)

## Classe

```
public class Carro
{
    public string Fabricante { get; set; }
    public string Modelo { get; set; }
    public string Cor { get; set; }
    public int NumeroPneus { get; set; }
    public int NumeroRetrovisores { get; set; }
    public bool DirecaoEsquerda { get; set; }

    public string Pais()
    {
        if (DirecaoEsquerda)
            return "Inglaterra";

        return "Europa continental";
    }
}
```

## Instância

```
Carro carro = new Carro();

carro.Fabricante = "Ferrari";
carro.Modelo = "Enzo";
carro.Cor = "Vermelha";
carro.NumeroPneus = 4;
carro.NumeroRetrovisores = 3;
carro.DirecaoEsquerda = false;

Console.WriteLine(carro.Pais());
```

# Orientação a objetos básica (5)

Instância

```
Carro sonho = new Carro();  
  
sonho.Fabricante = "Ferrari";  
sonho.Modelo = "Enzo";  
sonho.Cor = "Vermelha";  
sonho.NumeroPneus = 4;  
sonho.NumeroRetrovisores = 3;  
sonho.DirecaoEsquerda = false;
```

Instância

```
Carro real = new Carro();  
  
real.Fabricante = "Honda";  
real.Modelo = "Civic";  
real.Cor = "Prata";  
real.NumeroPneus = 4;  
real.NumeroRetrovisores = 3;  
real.DirecaoEsquerda = false;
```

# Orientação a objetos básica (6)

## Termos usados

**Propriedades:** Armazenam valores

```
public string Nome { get; set; }
public string Cor { get; set; }
public int NumeroPneus { get; set; }
public int NumeroRetrovisores { get; set; }
public bool DirecaoEsquerda { get; set; }
```

```
carro.Nome = "Ferrari Enzo";
carro.Cor = "Vermelha";
carro.NumeroPneus = 4;
carro.NumeroRetrovisores = 3;
carro.DirecaoEsquerda = true;
```

**Métodos:** Código executável

```
public string pais()
{
    if (DirecaoEsquerda)
        return "Inglaterra";
    return "Europa continental";
}
```

```
Console.WriteLine(carro.Pais());
```

# Orientação a objetos básica (7)

```
{ get; set; }
```

- ▶ **get**: Como o valor será recuperado
- ▶ **set**: Como um valor será armazenado

# Orientação a objetos básica (8)

```
{ get; set; }
```

Essa sintaxe diz que  
armazenará e recuperará  
valores sem nada especial  
acontecer

## Orientação a objetos básica (9)

```
private int myVar;  
  
public int MyProperty  
{  
    get { return myVar; }  
    set { myVar = value; }  
}
```

Permite alterar o comportamento default

# Orientação a objetos básica (9)

## Palavras reservadas

```
public int MyProperty
{
    get { return myVar; }
    set { myVar = value; }
}
```

**return:** indica a expressão de retorno

**value:** o valor originalmente atribuído sem nenhuma manipulação

# Orientação a objetos básica (10)

## Palavras reservadas

```
private int marcha;  
  
public int Marcha  
{  
    get { return marcha; }  
    set { marcha = value + 1; }  
}
```

```
carro.DirecaoEsquerda = true;  
Console.WriteLine(carro.pais());  
carro.Marcha = 5;  
Console.WriteLine($"A marcha atual é a {carro.Marcha}");
```

Inglaterra  
A marcha atual é a 6

# Orientação a objetos básica (11)

## Construtor

Método que tem o mesmo nome da classe.

É executado automaticamente quando a classe é instanciadas

```
public class NovoBoletim
{
    public string Aluno { get; set; }
    public List<float> Notas { get; set; }

    ...

    public NovoBoletim()
    {
        Executa alguma coisa...
    }
}
```

# Orientação a objetos básica (12)

## Modificadores de Acesso

**private**: Só é “enxergado” no arquivo que estiver presente

**public**: É “enxergado” em todo o aplicativo

**static**: Não é necessário instanciar  
O que *implica que só existe uma cópia!*

Obs: Existem muitos outros que neste momento não são mencionados com intenção de manter simples para o curso presente



# VIII: Estrutura de dados

# Estrutura de dados (1)

- ▶ Crie um diretório para conter o código
- ▶ Faça um fork para a sua conta de:
  - ▶ <https://github.com/sdifiore/Revisao.git>

## Estrutura de dados (2)

- ▶ Abra um terminal PowerShell no diretório criado;
- ▶ Clone o repositório
- ▶ Abra o VS Code nele