

Controle de Versão e Colaboração

Comandos de navegação e listagem de arquivos

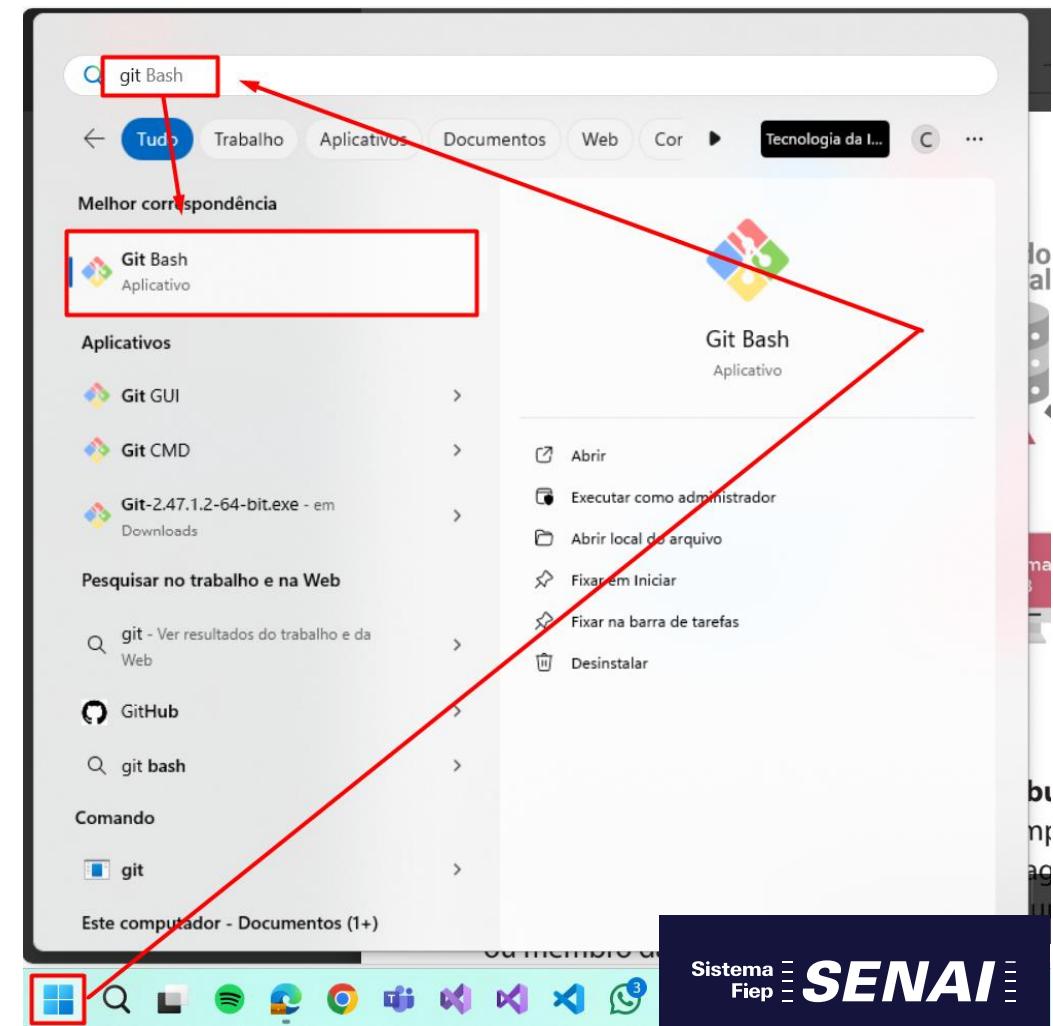
Controle de Versão e Colaboração

Comandos de navegação e listagem de arquivos



Terminal Git Bash

- Na barra de tarefas do Windows, clicar no botão iniciar;
- Na caixa de edição digitar “git”;
- Clicar no aplicativo “Git Bash” que deverá aparecer automaticamente conforme indicado na figura ao lado;



Git Bash
Aplicativo

Controle de Versão e Colaboração

Comandos de Navegação e Listagem de Arquivos



| Comando | Descrição | Exemplo |
|---------------------|---|--|
| <code>pwd</code> | Exibe o diretório atual | <code>pwd</code> → <code>/home/usuario</code> |
| <code>cd</code> | Navega entre diretórios | <code>cd /home/usuario/docs</code> |
| <code>ls</code> | Lista arquivos/diretórios | <code>ls</code> → <code>arquivo.txt pasta</code> |
| <code>ls -l</code> | Lista arquivos com detalhes | <code>ls -l</code> → Exibe permissões, tamanho, etc. |
| <code>ls -al</code> | Lista arquivos ocultos e detalhes | <code>ls -al</code> → Inclui arquivos iniciados com <code>.</code> |
| <code>cd ~</code> | Retorna ao diretório pessoal | <code>cd ~</code> |
| <code>cd /c</code> | Acessa unidade C: (Git Bash no Windows) | <code>cd /c/Users/usuario</code> |

Controle de Versão e Colaboração

Comandos de Navegação e Listagem de Arquivos



| Comando | Descrição | Exemplo |
|--------------------|--------------------------------------|--|
| <code>clear</code> | Limpa a tela do terminal | <code>clear</code> |
| <code>exit</code> | Fechá a sessão do terminal | <code>exit</code> |
| <code>which</code> | Mostra o caminho de um comando | <code>which python</code> → <code>/usr/bin/python</code> |
| <code>help</code> | Exibe ajuda sobre um comando interno | <code>help cd</code> |
| <code>mkdir</code> | Cria um diretório | <code>mkdir projetos</code> |
| <code>touch</code> | Cria um arquivo vazio | <code>touch novo.txt</code> |
| <code>cat</code> | Exibe o conteúdo de um arquivo | <code>cat texto.txt</code> |

Controle de Versão e Colaboração

Comandos de Navegação e Listagem de Arquivos



| Comando | Descrição | Exemplo |
|---------------------|--|---|
| <code>less</code> | Exibe arquivos grandes de forma paginada | <code>less log.txt</code> (<code>q</code> para sair) |
| <code>cp</code> | Copia arquivos ou diretórios | <code>cp arquivo.txt backup/</code> |
| <code>mv</code> | Move ou renomeia arquivos | <code>mv documento.txt docs/</code> |
| <code>rm</code> | Remove arquivos | <code>rm arquivo.txt</code> |
| <code>rm -rf</code> | Remove diretórios e arquivos sem confirmação | <code>rm -rf pasta_antiga</code> |
| <code>rmdir</code> | Remove diretórios vazios | <code>rmdir pasta_vazia</code> |

Controle de Versão e Colaboração

Comandos de Navegação e Listagem de Arquivos



Comandos Básicos

No terminal Git Bash:

- Descubra onde você está;
- Liste os arquivos e pastas do diretório atual;
- Navegue para a pasta do usuário;
- Crie um novo diretório chamado “projetos”;
- Entre no diretório recém criado;
- Crie um arquivo chamado “lembrete.txt” dentro da pasta “projeto”;
- Edite o arquivo e escreva algo nele (use echo);
- Exiba o conteúdo na tela;

Controle de Versão e Colaboração

Comandos de Navegação e Listagem de Arquivos



Comandos Básicos

... Continuação do slide anterior

- Copie o arquivo “lembrete.txt” para um novo arquivo “backup.txt”
- Renomeie o arquivo “backup.txt” para “importante.txt”
- Crie uma subdiretório chamado “antigos”
- Mova o arquivo antigo “importante.txt” para o subdiretório “antigos”
- Estenda o conteúdo do arquivo “backup.txt” com uma nova mensagem;
- Exiba o conteúdo na tela do arquivo “backup.txt”
- Copie o arquivo “backup.txt” para “sobre.txt”
- Remova o arquivo “lembrete.txt”
- Remova o diretório “projetos” e todo o seu conteúdo



A dark blue-toned photograph of two men in a workshop setting. One man is in the foreground, wearing a blue shirt and a red cap, looking down at a task. Another man is visible behind him, also focused on work. The background is slightly blurred.

Controle de Versão e Colaboração

Introdução

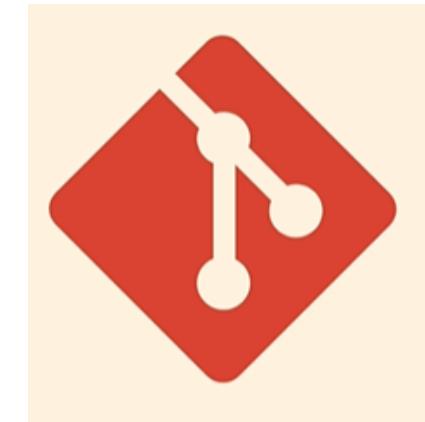
Controle de Versão e Colaboração

Introdução



O que é o Git?

- Sistema de controle de versão distribuído;
- É utilizado para rastrear alterações em arquivos e coordenar o trabalho de múltiplos desenvolvedores em um projeto;
- Permite acompanhar o histórico de modificações no código;
- Permite gerenciar diversas versões de um software;



Controle de Versão e Colaboração

Introdução



Principais características do Git

- **Distribuído:** Cada desenvolvedor tem uma cópia completa do repositório, incluindo o histórico de alterações;
- **Rápido e eficiente:** As operações são executadas localmente na máquina do usuário, tornando o processo rápido;
- **Branching e merging poderosos:** O Git facilita a criação e fusão de ramificações (branches), permitindo o desenvolvimento paralelo e testes isolados;
- **Controle de versões robusto:** Permite reverter mudanças, comparar versões e criar registros detalhados do histórico do projeto

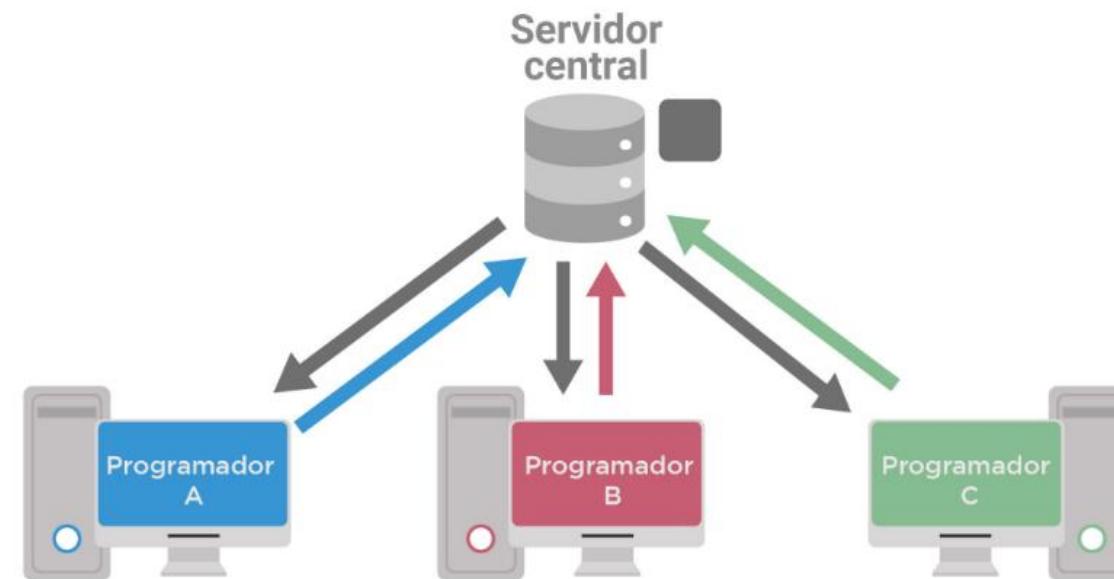
Controle de Versão e Colaboração

Introdução



Topologia de Funcionamento

$$\text{[blue square]} + \text{[red square]} + \text{[green square]} = \text{[grey square]}$$

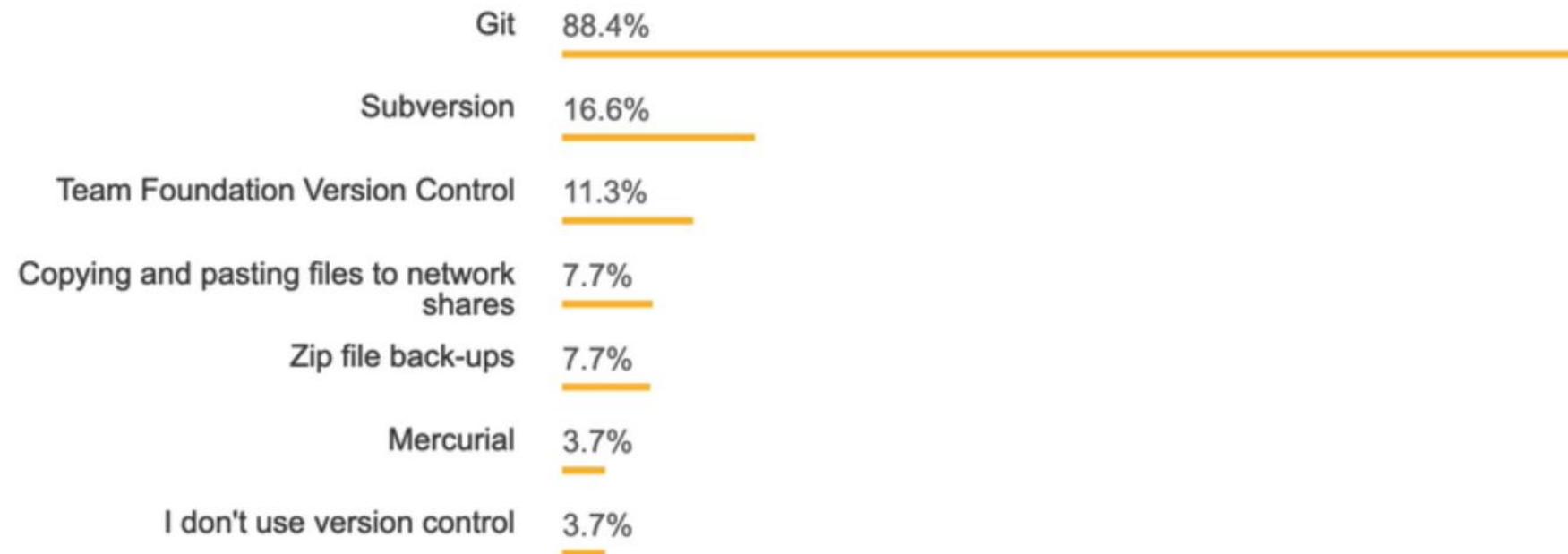


Controle de Versão e Colaboração

Introdução



Controlador de versionamento mais utilizados



Controle de Versão e Colaboração

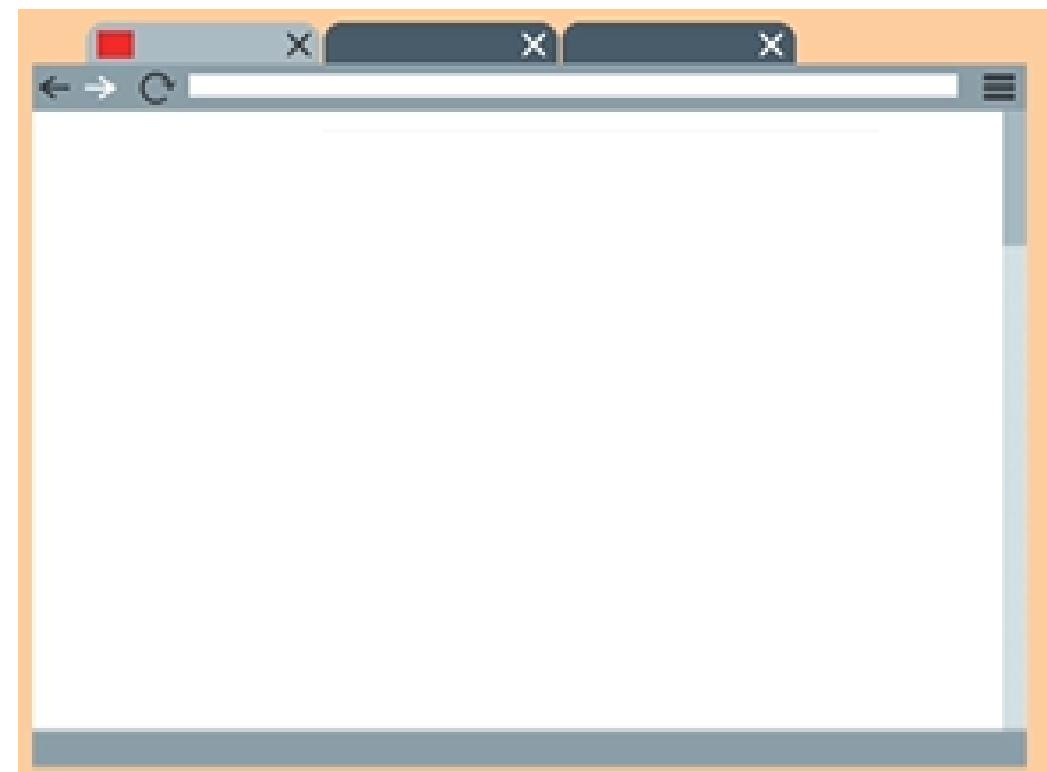
Introdução



Como funciona

- Começamos um novo projeto
- O projeto é um website
- O projeto está em branco

Progresso do projeto

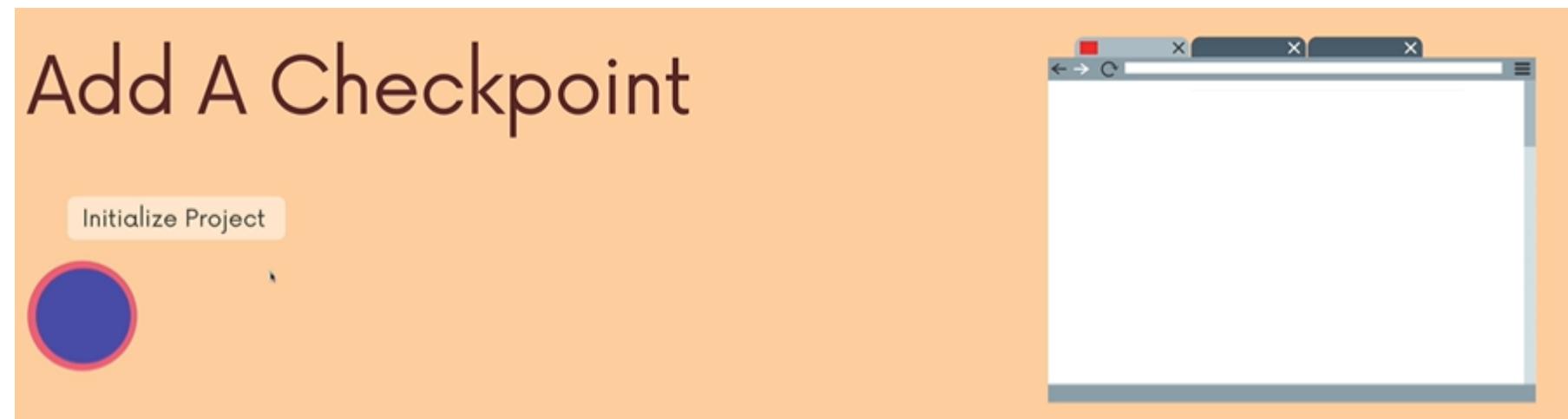


Controle de Versão e Colaboração

Introdução



Adicionamos um ponto checagem do projeto



Controle de Versão e Colaboração

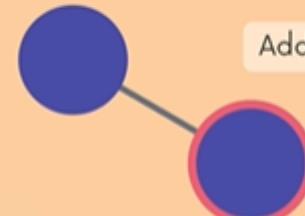
Introdução



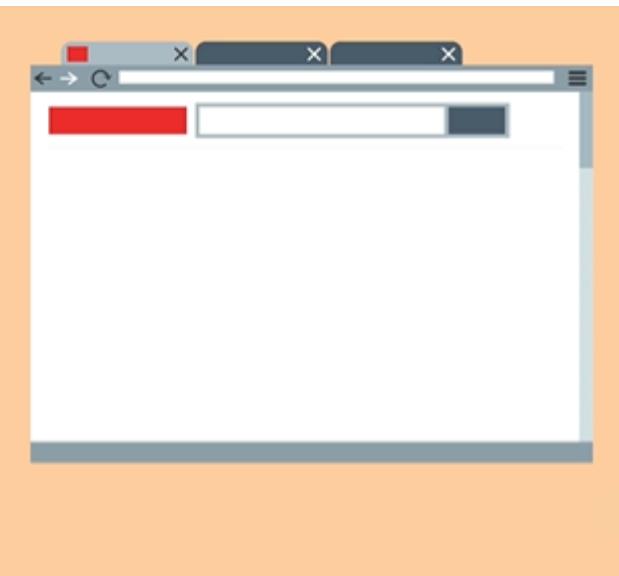
Alteração do projeto: Navbar

Add A Checkpoint

Initialize Project



Add Top Navbar

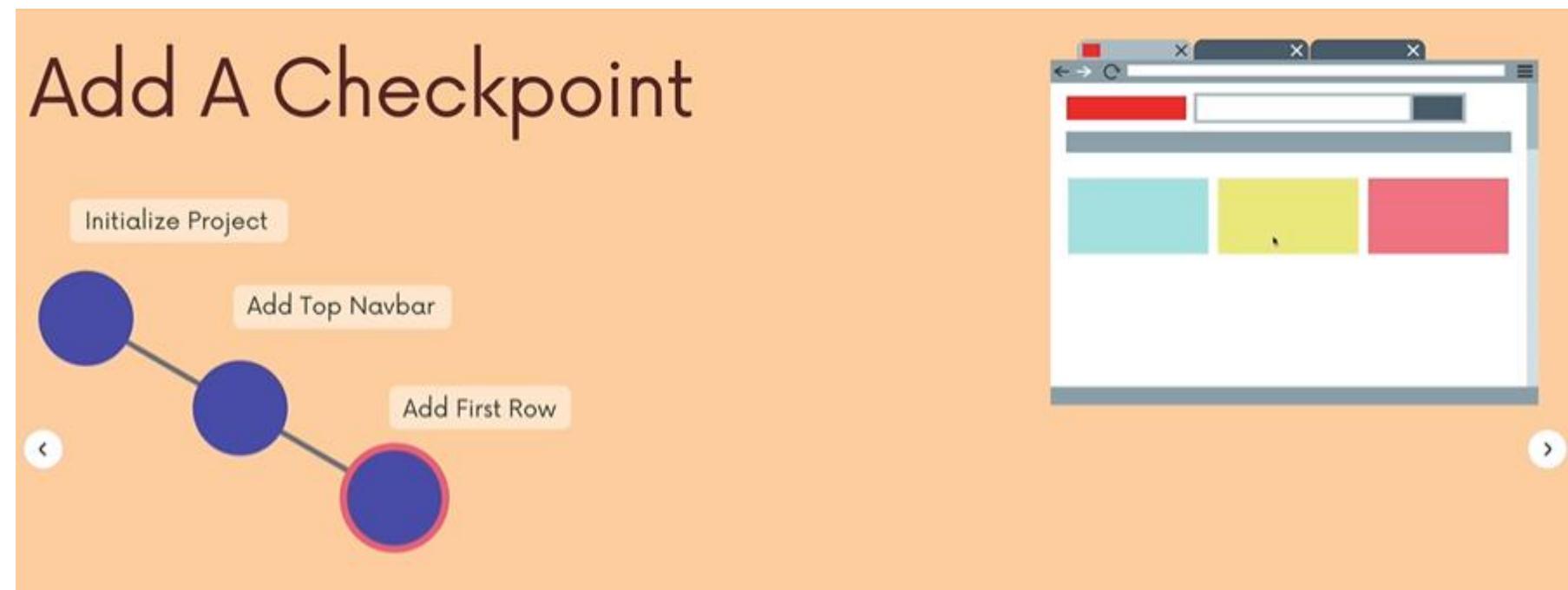


Controle de Versão e Colaboração

Introdução

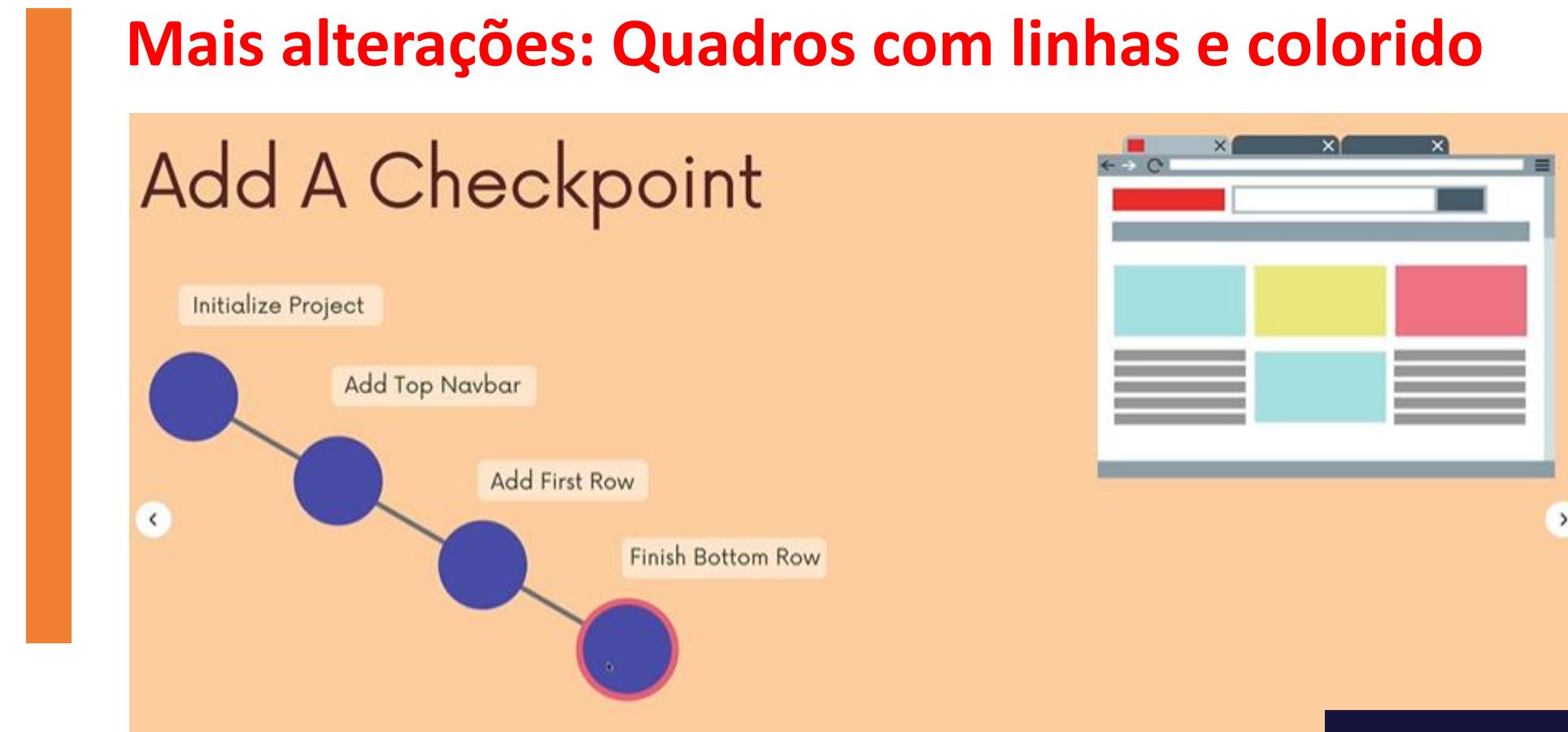


Mais alterações: Quadros coloridos



Controle de Versão e Colaboração

Introdução



Controle de Versão e Colaboração

Introdução



Alterações: Dark Mode e cores

Add A Checkpoint

Initialize Project



Add Top Navbar



Add First Row



Finish Bottom Row



Add Dark Theme



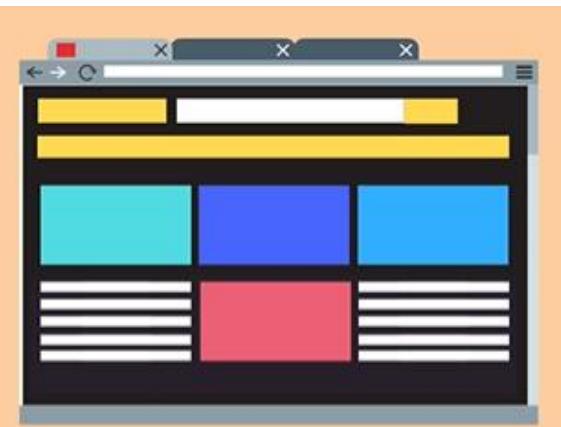
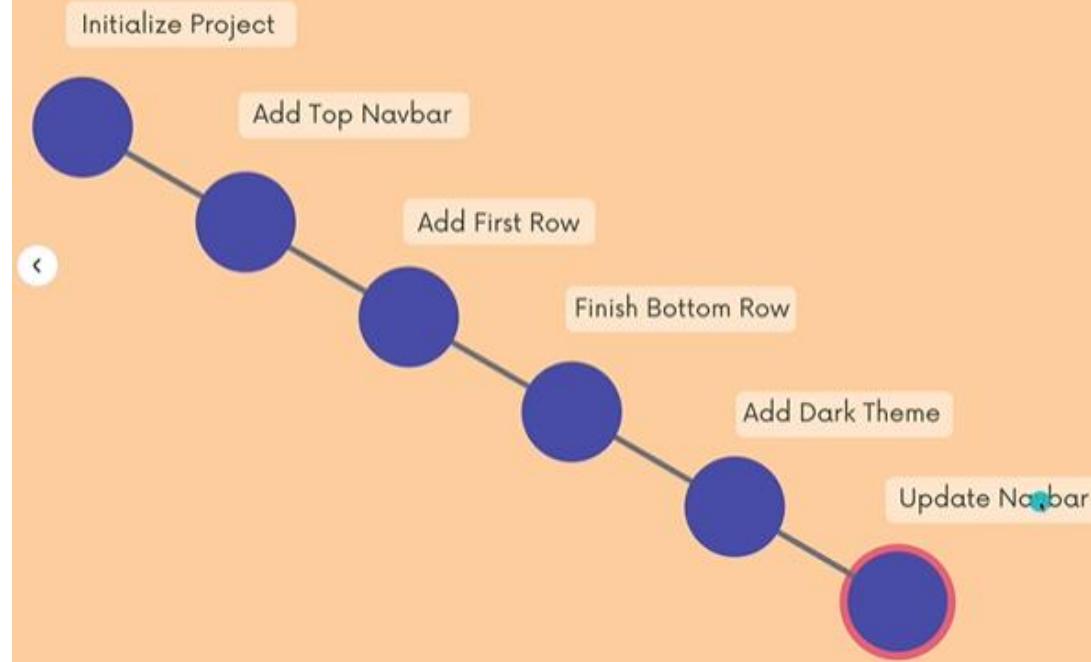
Controle de Versão e Colaboração

Introdução



Alterações: Cores do navbar

Add A Checkpoint



Controle de Versão e Colaboração

Introdução



Podemos voltar a um checkpoint antigo

I can go back to prior checkpoints I made!

Initialize Project

Add Top Navbar

Add First Row

Finish Bottom Row

Add Dark Theme

Update Navbar

Controle de Versão e Colaboração

Introdução



Adicionamos novas funcionalidades

I add a new checkpoint!

Initialize Project

Add Top Navbar

Add First Row

Finish Bottom Row

Add Dark Theme

Add Third Row

Update Navbar

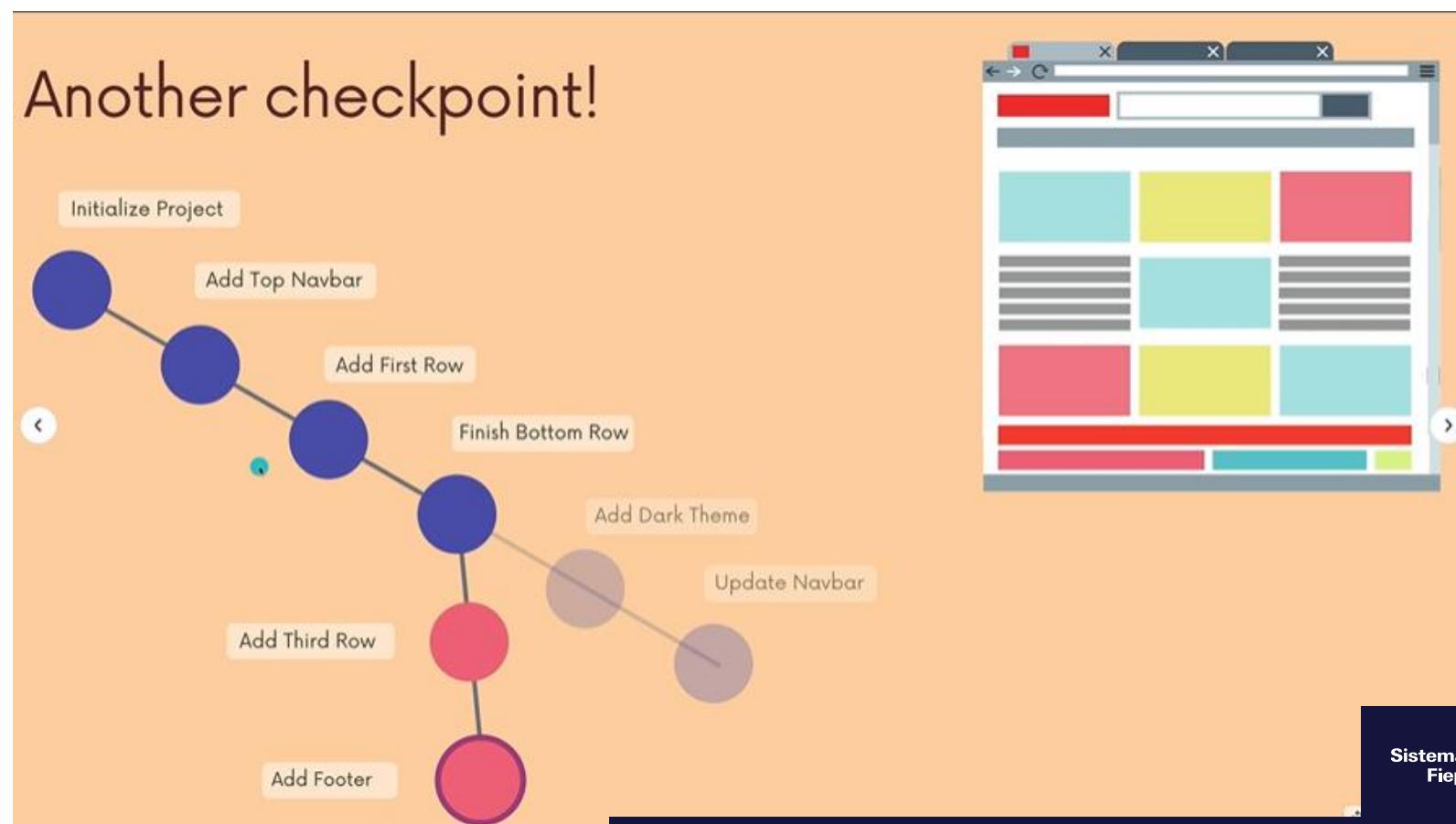


Controle de Versão e Colaboração

Introdução



Alteração: Footer



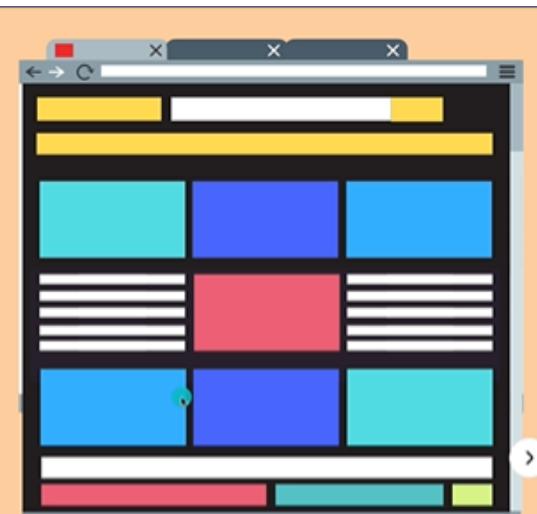
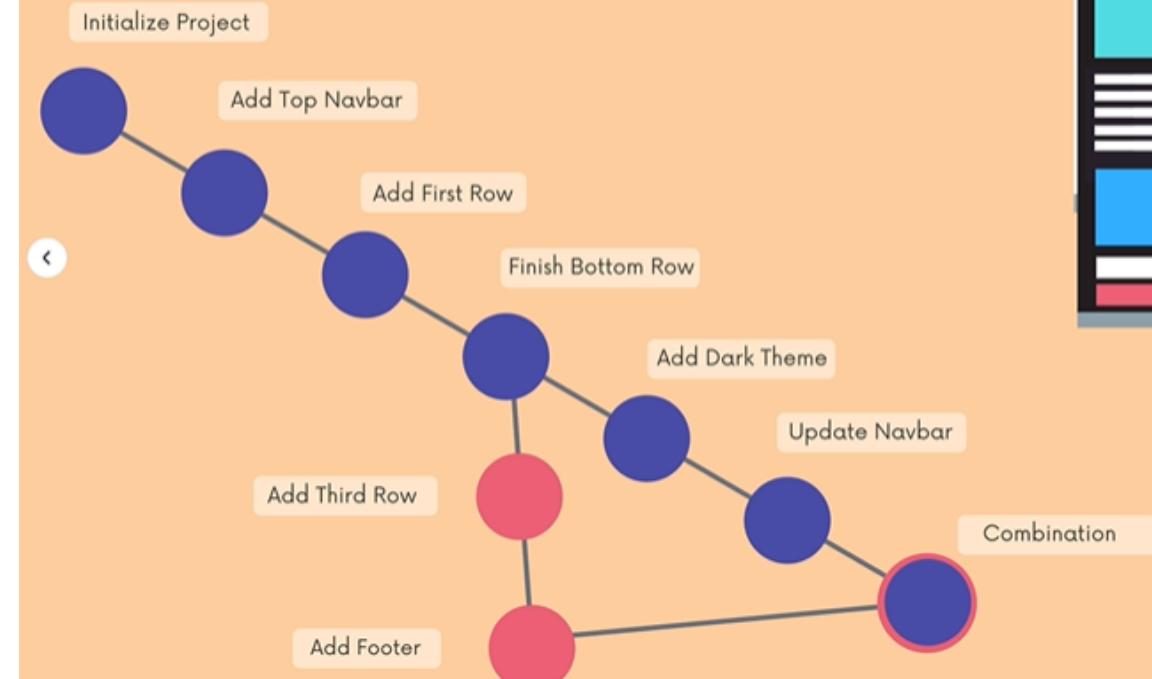
Controle de Versão e Colaboração

Introdução



Combinar as alterações em um novo checkpoint

And I can even combine checkpoints!



Controle de Versão e Colaboração

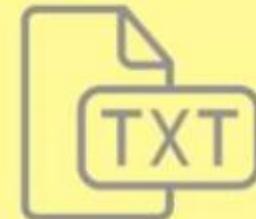
Introdução



Diretório de trabalho

Working Directory

~/projects/demo

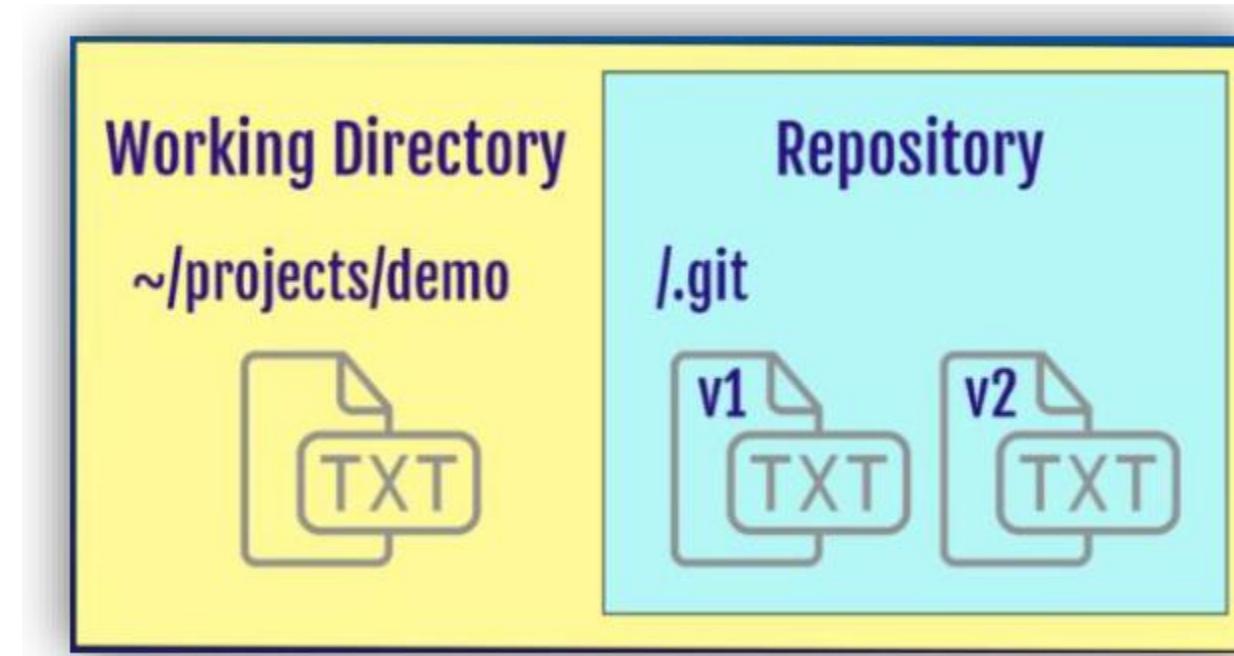


Controle de Versão e Colaboração

Introdução



Repositório

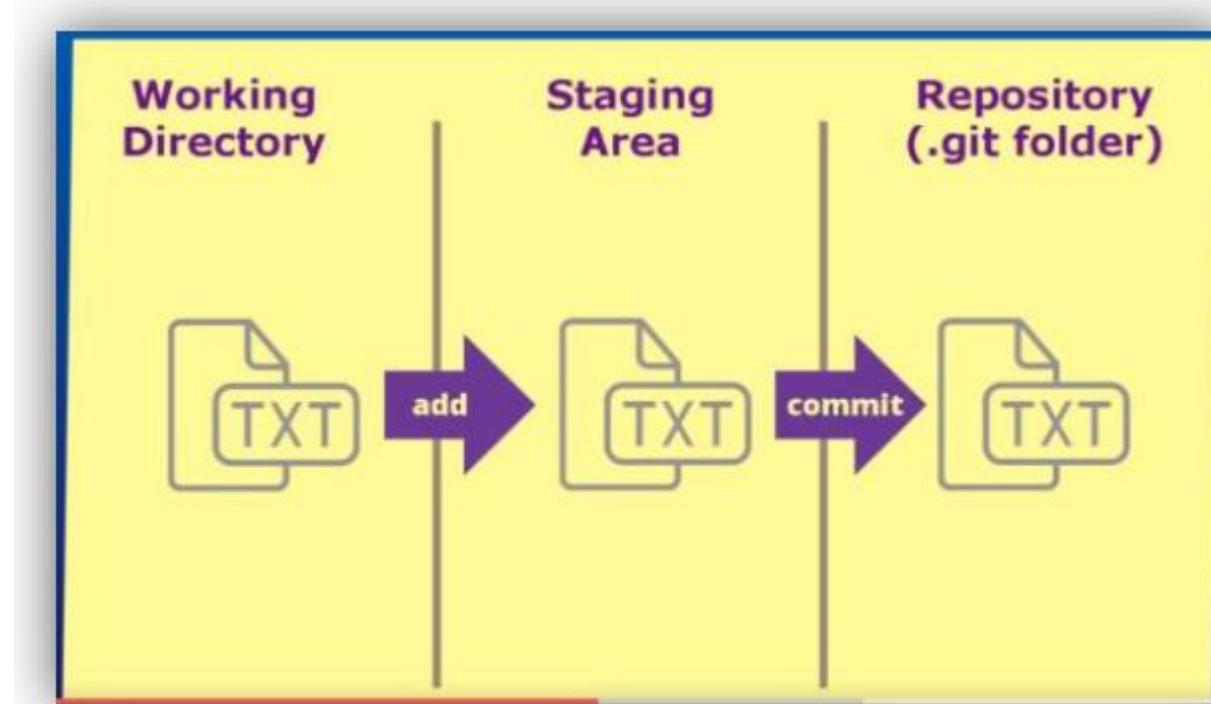


Controle de Versão e Colaboração

Introdução



Fluxo de trabalho local

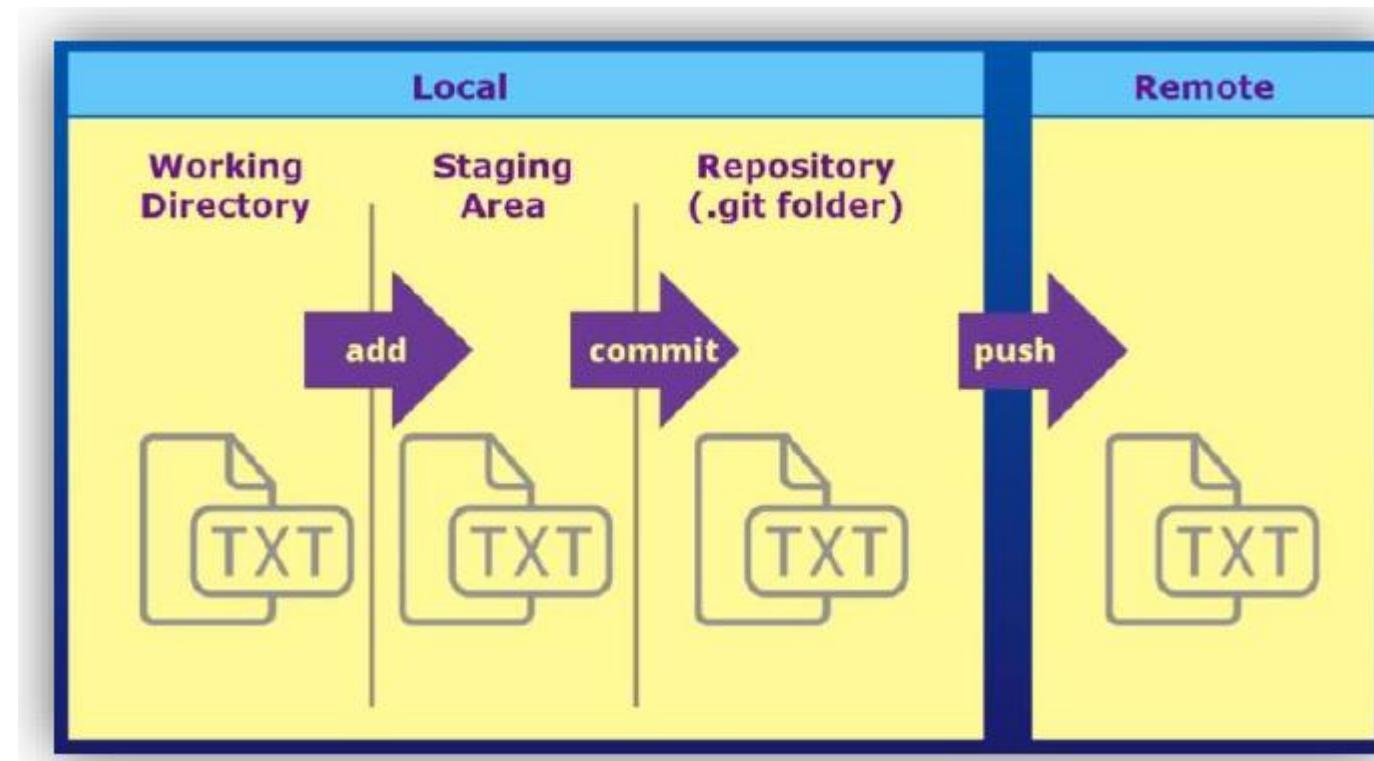


Controle de Versão e Colaboração

Introdução



Fluxo de trabalho remoto



Controle de Versão e Colaboração

Termos e Comandos

Controle de Versão e Colaboração

Termos e comandos

.....

Agora vamos abordar outros termos típicos da área para que você comece a se familiarizar.

Prompt

É a janelinha na qual digitamos os comandos, também chamada de terminal.

Repositório local

Pasta ou diretório na máquina do programador, na qual são gravadas todas as versões do código.

Repositório remoto

Pasta ou diretório na nuvem, na qual é gravado o projeto completo, com todas as versões do código.

Controle de Versão e Colaboração

Comandos de Navegação e Listagem de Arquivos



Staging

Local intermediário entre a máquina do desenvolvedor e o repositório remoto, para revisão de alterações antes de salvar no histórico.

Commit

Ato de registrar ou salvar as alterações.

Ramo principal

Sequência principal do código-fonte.

Controle de Versão e Colaboração

Comandos de Navegação e Listagem de Arquivos



Branch

É uma ramificação do ramo principal, para testes em geral.

README

Arquivo que é uma mescla de cartão de visitas e documentação do projeto.

.gitignore

Arquivo de texto que guarda os arquivos ou pastas que devem ser ignorados, ou seja, os arquivos nomeados no *.gitignore* não serão monitorados pelo Git.



Controle de Versão e
Colaboração
Configurações Iniciais
`git config`



Controle de Versão e Colaboração

Configurações Iniciais



A primeira configuração é identificar o nome de usuário e um endereço de e-mail. Assim, todas as atualizações (ou *commits*) feitas poderão ser identificadas.

Essa configuração pode ser global (válida para todo o sistema) ou local (para um projeto específico).

Fazendo a configuração global, não é necessário configurar cada projeto. Se a configuração local precisa ser alterada para um projeto específico, você pode rodar o comando sem a configuração global neste projeto.



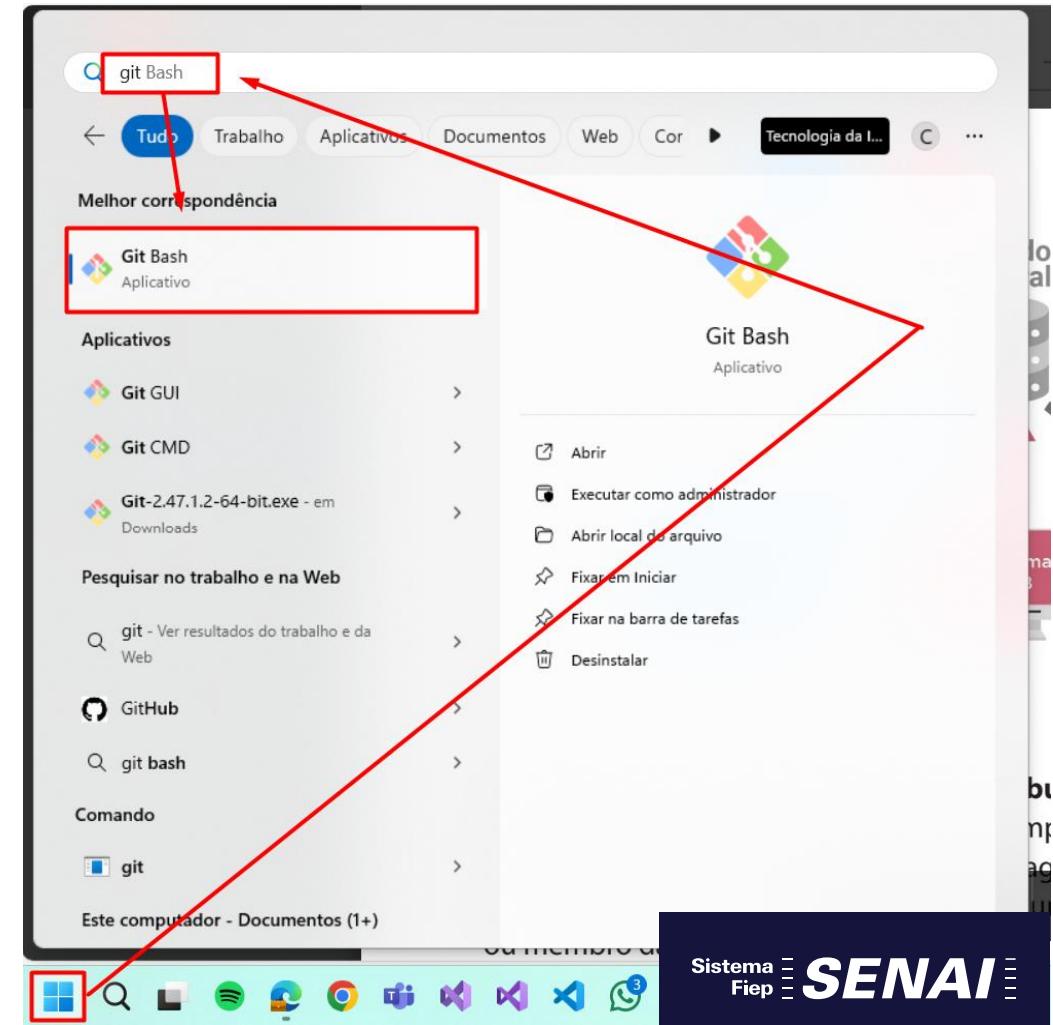
Controle de Versão e Colaboração

Configurações Iniciais



Terminal Git Bash

- Na barra de tarefas do Windows, clicar no botão iniciar;
- Na caixa de edição digitar “git”;
- Clicar no aplicativo “Git Bash” que deverá aparecer automaticamente conforme indicado na figura ao lado;



Git Bash
Aplicativo

Controle de Versão e Colaboração

Configurações Iniciais

.....



The screenshot shows a terminal window titled 'MINGW64:/c/Users/Administrador/Documents/_git/senai-versoes-colaboracione...'. It displays three commands entered by the user:

```
Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracione
s (master)
$ git config --global user.name "fulano de tal"

Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracione
s (master)
$ git config --global user.email "fulano@email.com"

Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracione
s (master)
$ |
```

Importante!

Esses dois passos atualizarão suas informações de usuário e cada alteração feita manterá sua identidade como autor.



Controle de Versão e Colaboração

Configurações Iniciais



Configurando usuário e email

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ git config --global user.name "senai-tds"

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ git config --global user.email "carlos.stunitz@outlook.com"
```

Controle de Versão e Colaboração

Configurações Iniciais



Consultando configurações

```
git config --global --list
```

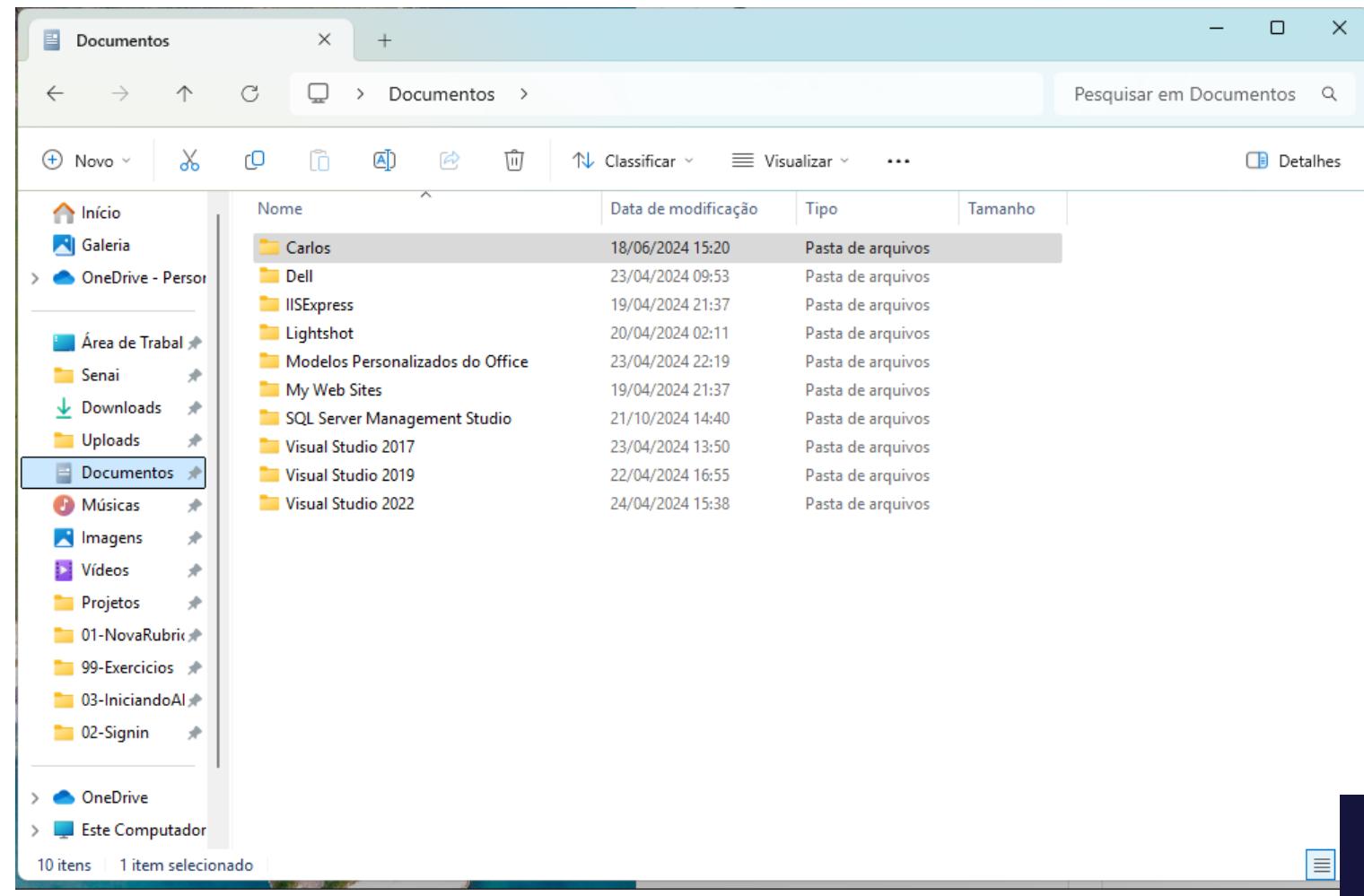
```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ git config --global --list
user.name=senai-tds
user.email=carlos.stunitz@outlook.com
gui.recentrepo=C:/Projetos/VS2017/TInfo/Seg2C
```



Controle de Versão e
Colaboração
Criando a estrutura

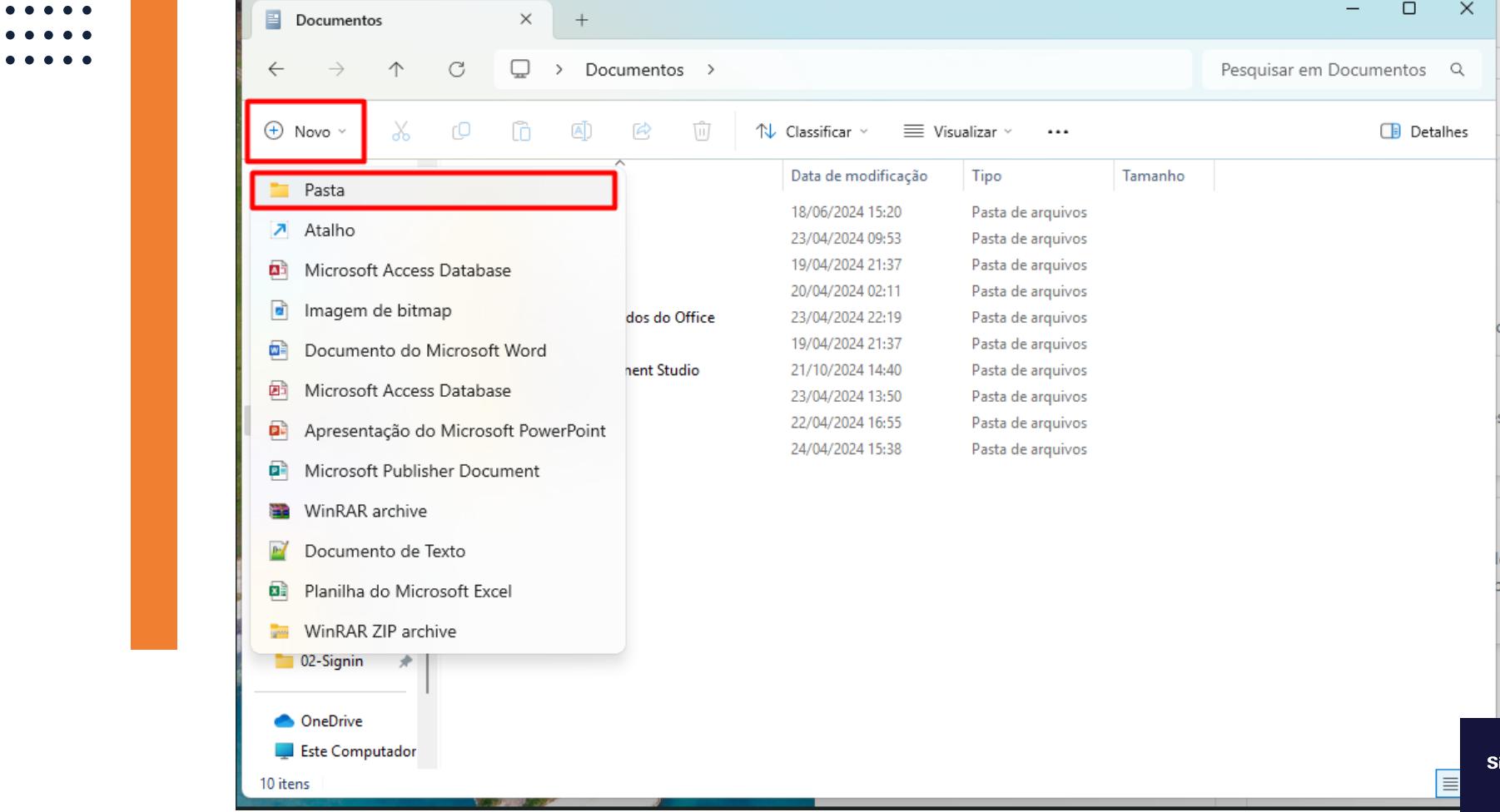
Controle de Versão e Colaboração

Criando a estrutura



Controle de Versão e Colaboração

Criando a estrutura



Controle de Versão e Colaboração

Criando a estrutura



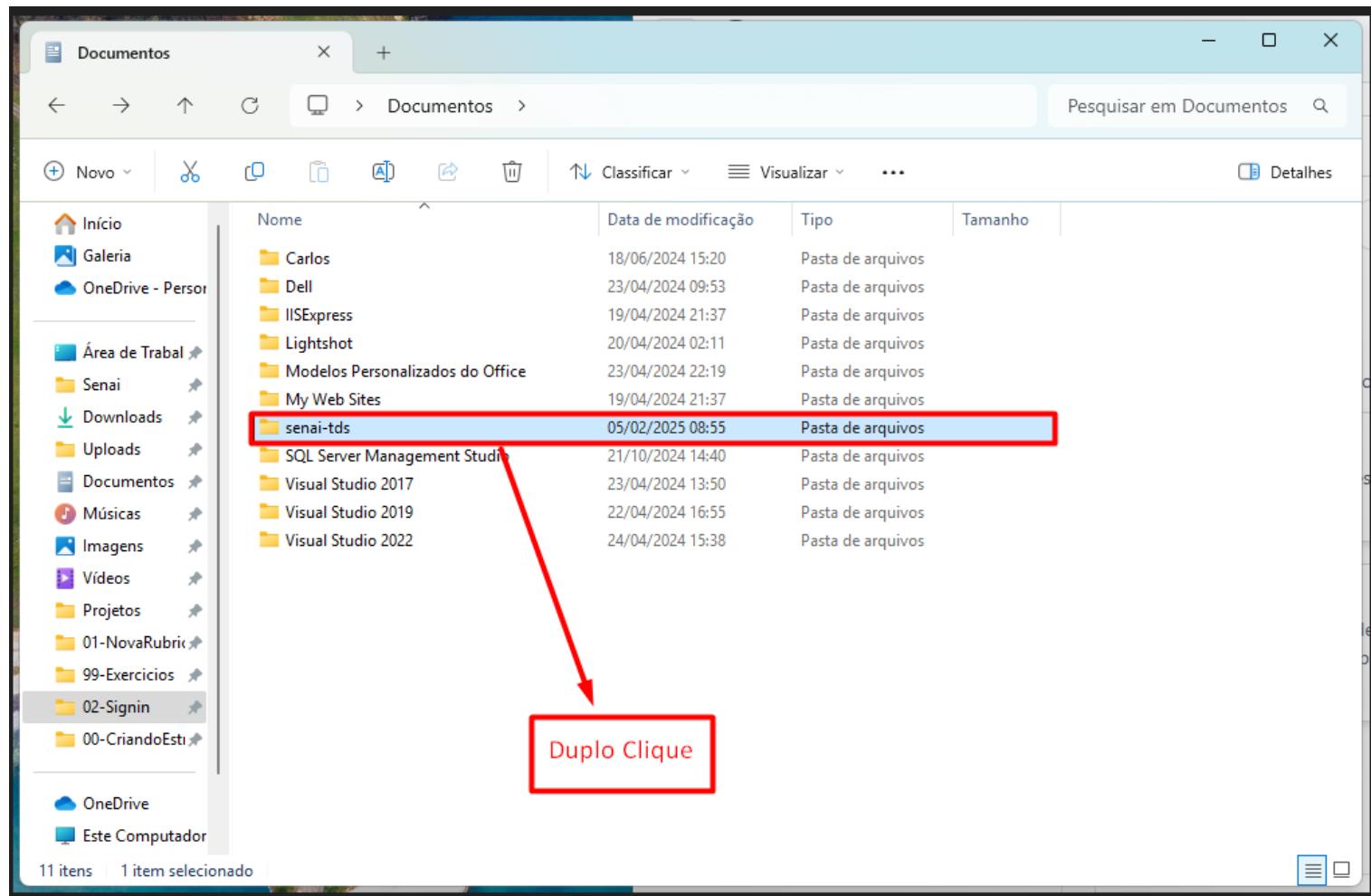
| Nome | Data de modificação | Tipo |
|----------------------------------|---------------------|-------------------|
| Carlos | 18/06/2024 15:20 | Pasta de arquivos |
| Dell | 23/04/2024 09:53 | Pasta de arquivos |
| IISExpress | 19/04/2024 21:37 | Pasta de arquivos |
| Lightshot | 20/04/2024 02:11 | Pasta de arquivos |
| Modelos Personalizados do Office | 23/04/2024 22:19 | Pasta de arquivos |
| My Web Sites | 19/04/2024 21:37 | Pasta de arquivos |
| SQL Server Management Studio | 21/10/2024 14:40 | Pasta de arquivos |
| Visual Studio 2017 | 23/04/2024 13:50 | Pasta de arquivos |
| Visual Studio 2019 | 22/04/2024 16:55 | Pasta de arquivos |
| Visual Studio 2022 | 24/04/2024 15:38 | Pasta de arquivos |
| senai-tds | 05/02/2025 08:55 | Pasta de arquivos |

é importante que o nome desta pasta seja exatamente igual ao nome da conta criada no GitHub pois o padrão de nomenclatura de projetos do GitHub é, por exemplo:

==> senai-tds/Algoritmos

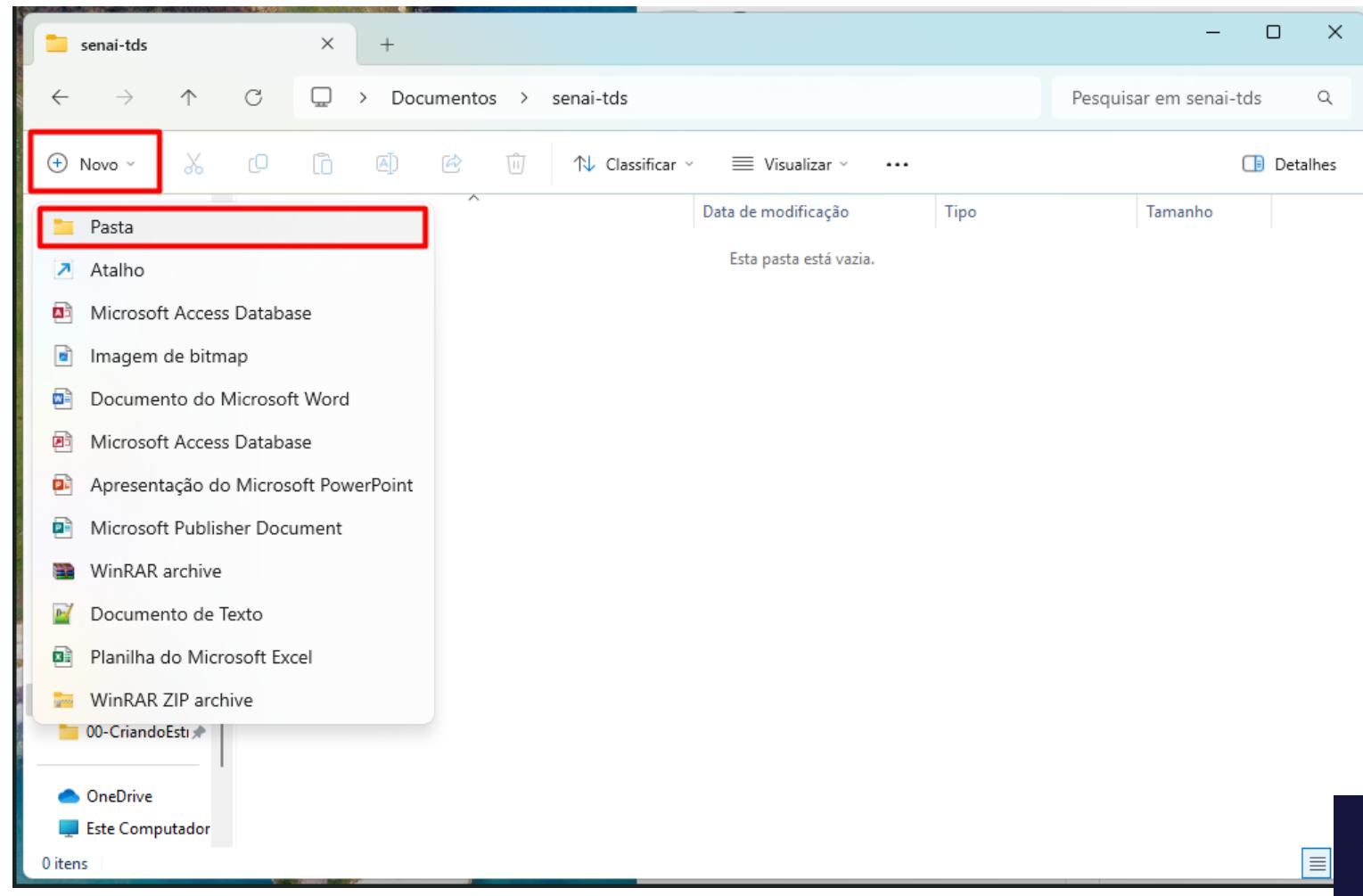
Controle de Versão e Colaboração

Criando a estrutura



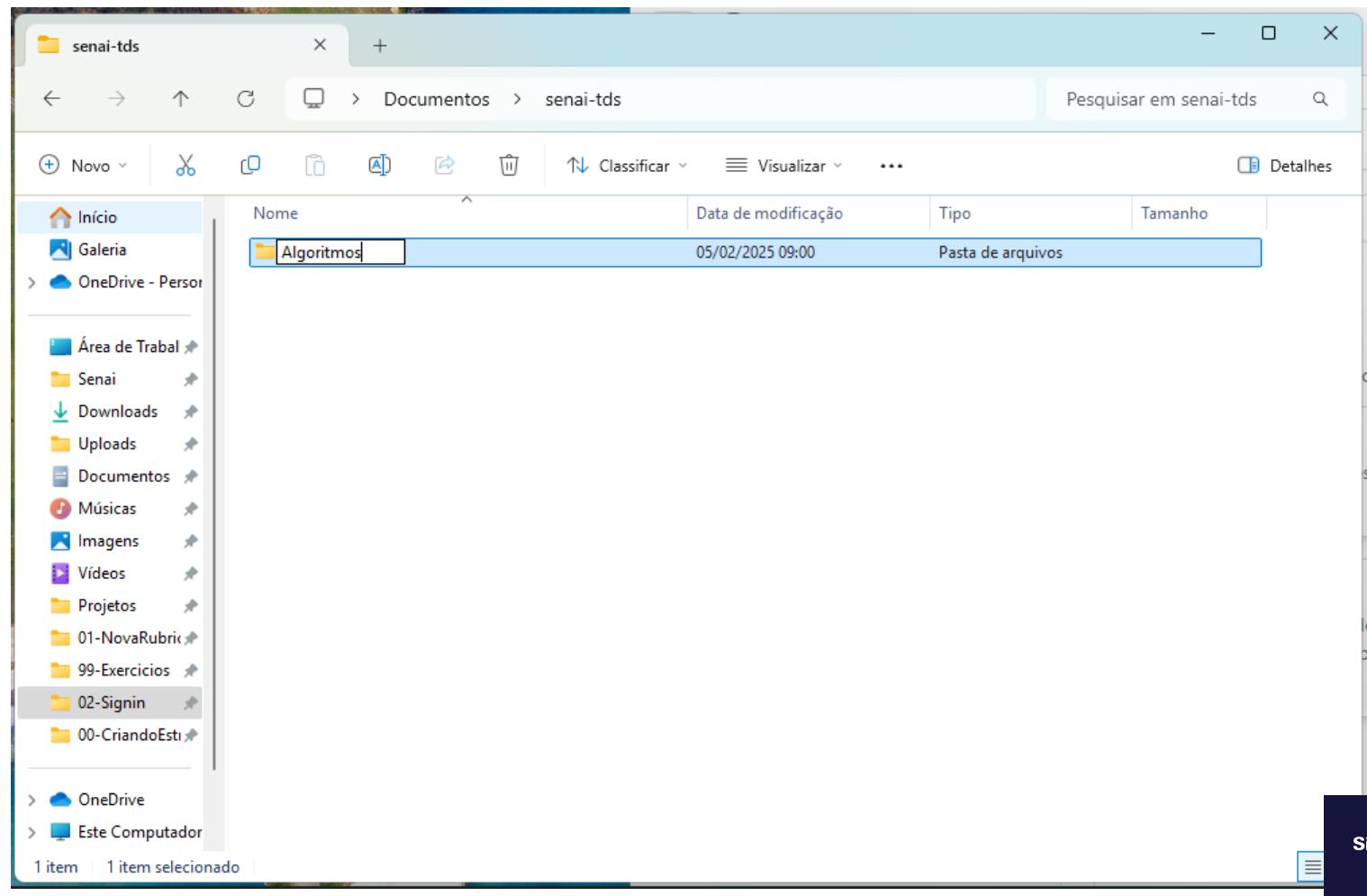
Controle de Versão e Colaboração

Criando a estrutura



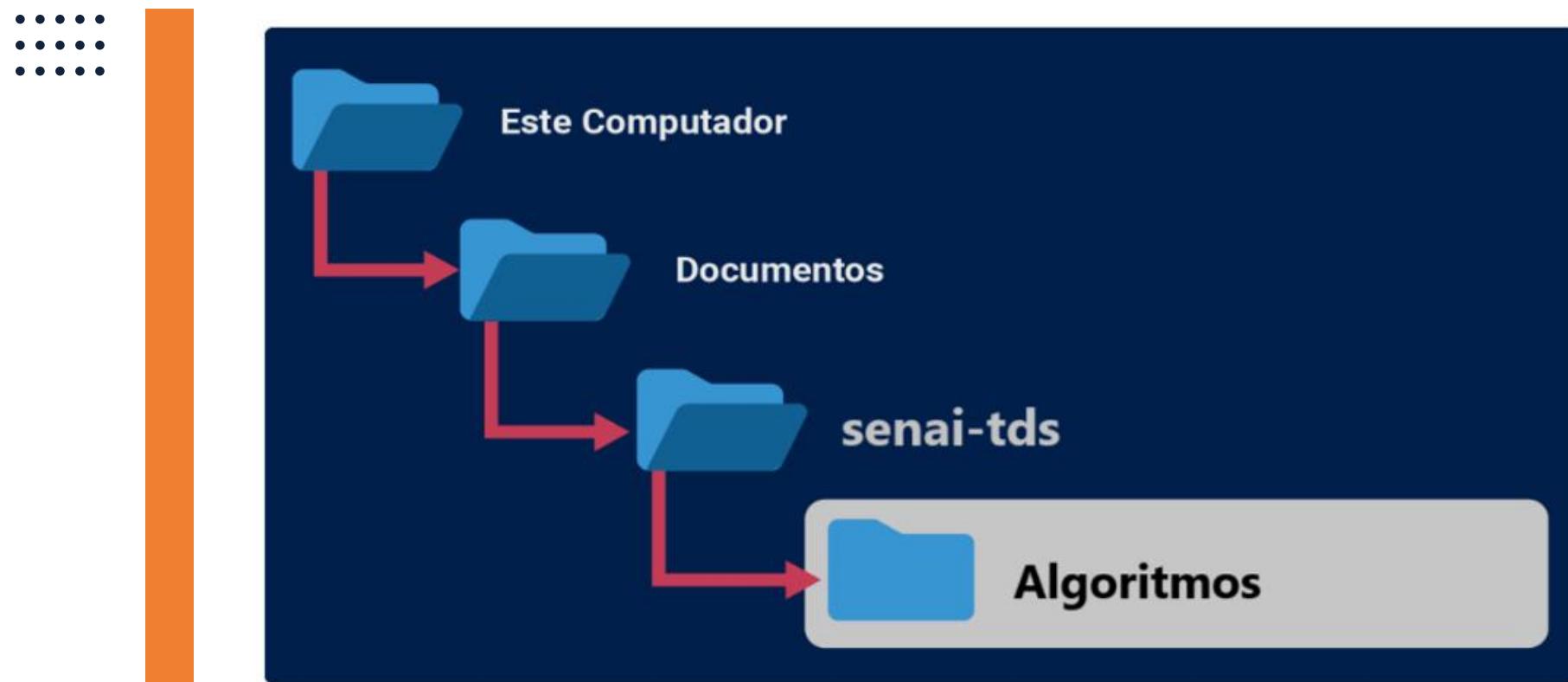
Controle de Versão e Colaboração

Criando a estrutura



Controle de Versão e Colaboração

Criando a estrutura



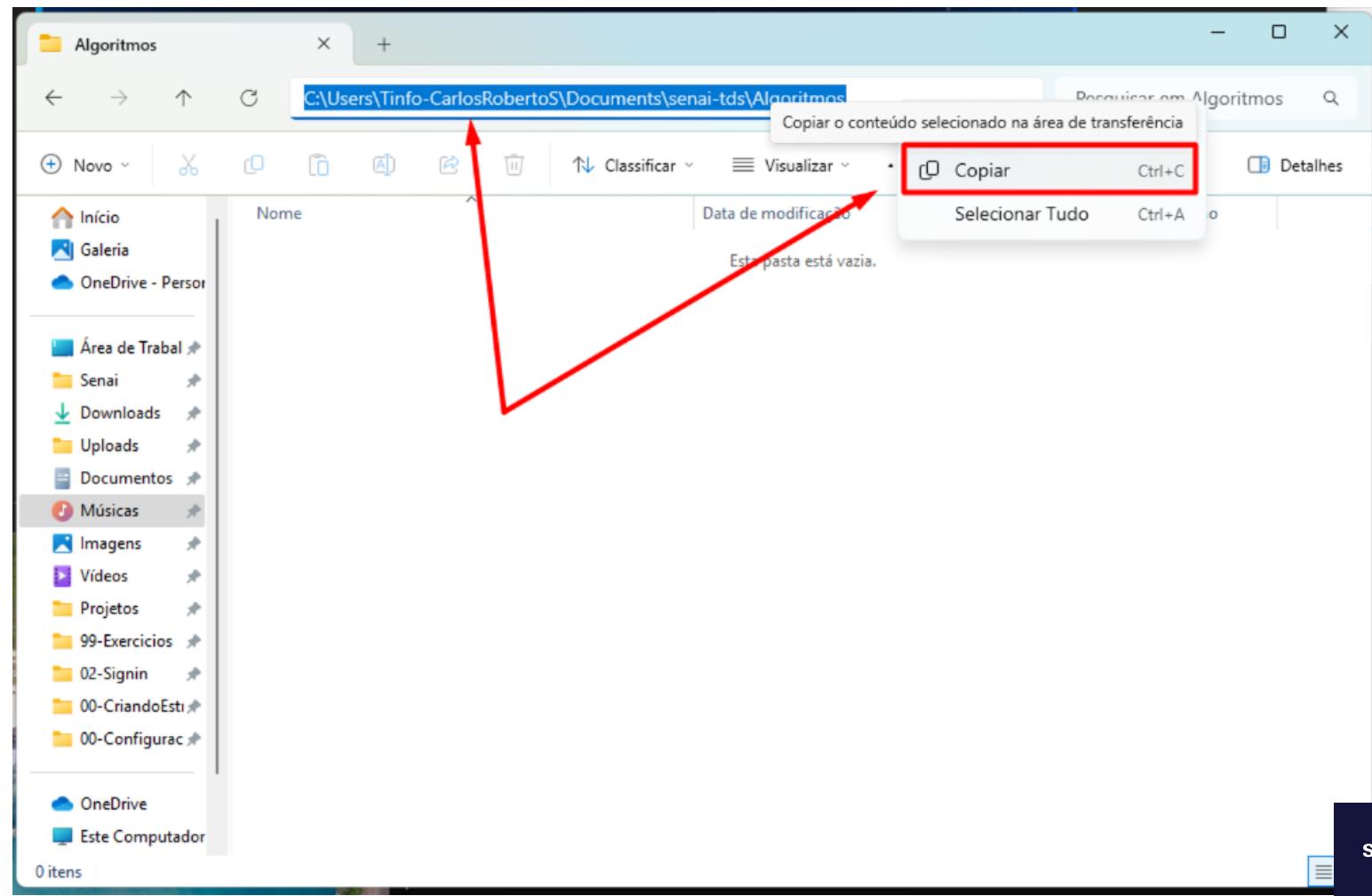
Controle de Versão e Colaboração

Criando repositório local **git init**



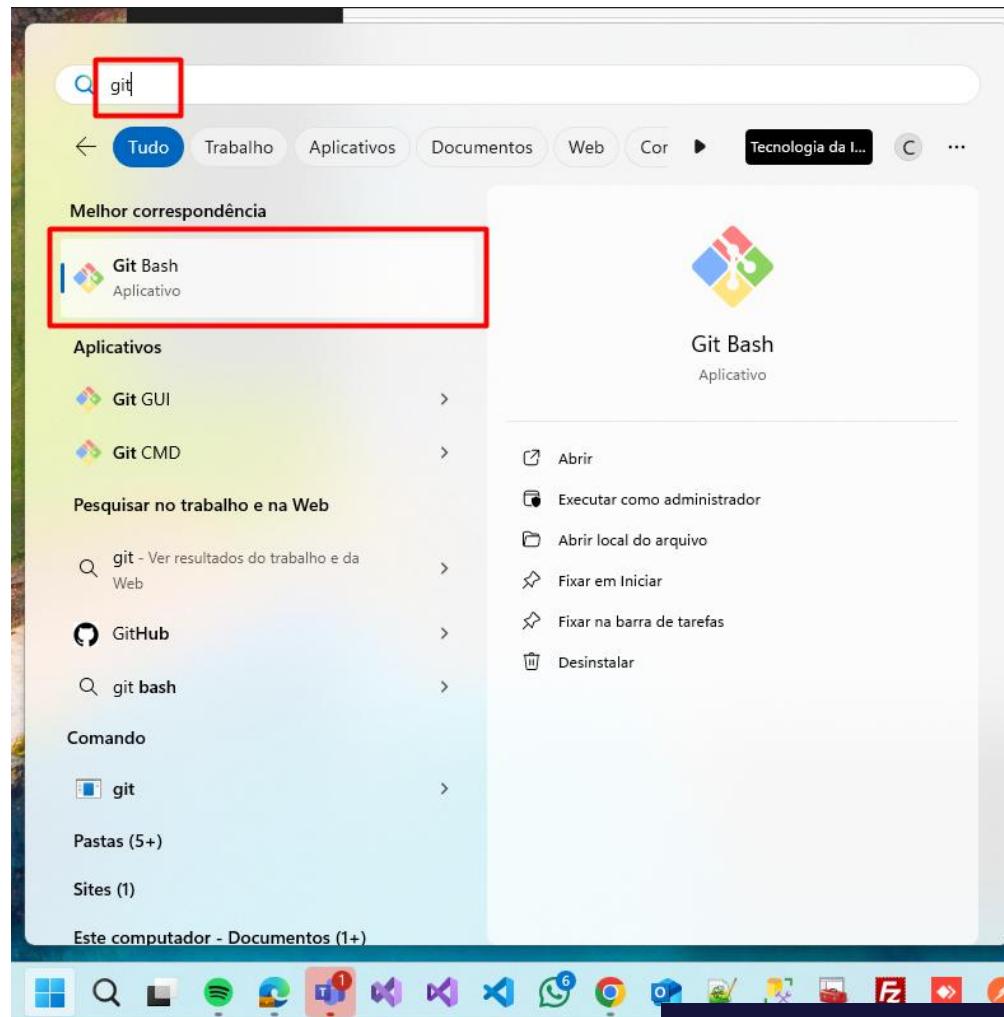
Controle de Versão e Colaboração

Criando o repositório local



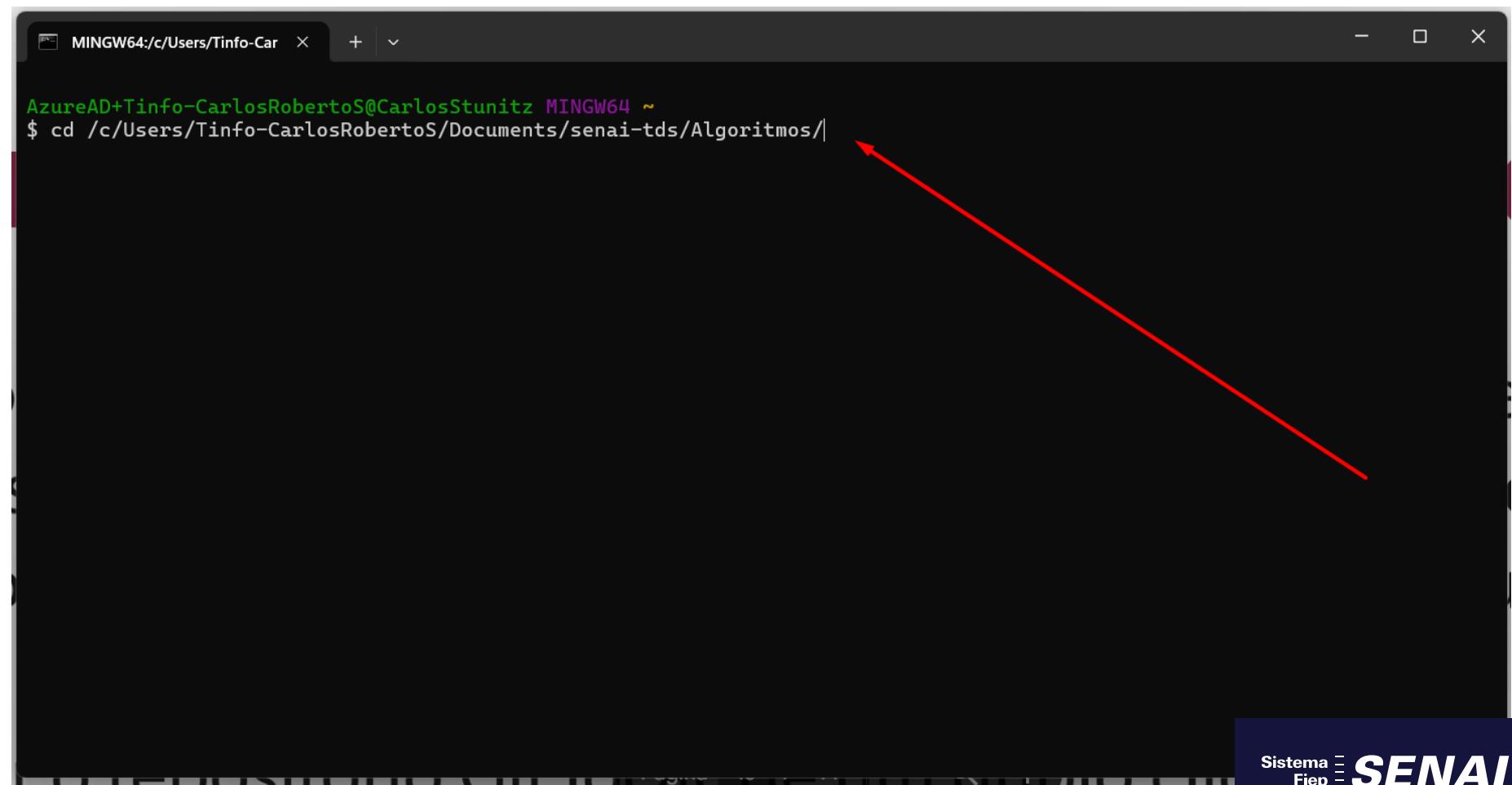
Controle de Versão e Colaboração

Criando o repositório local



Controle de Versão e Colaboração

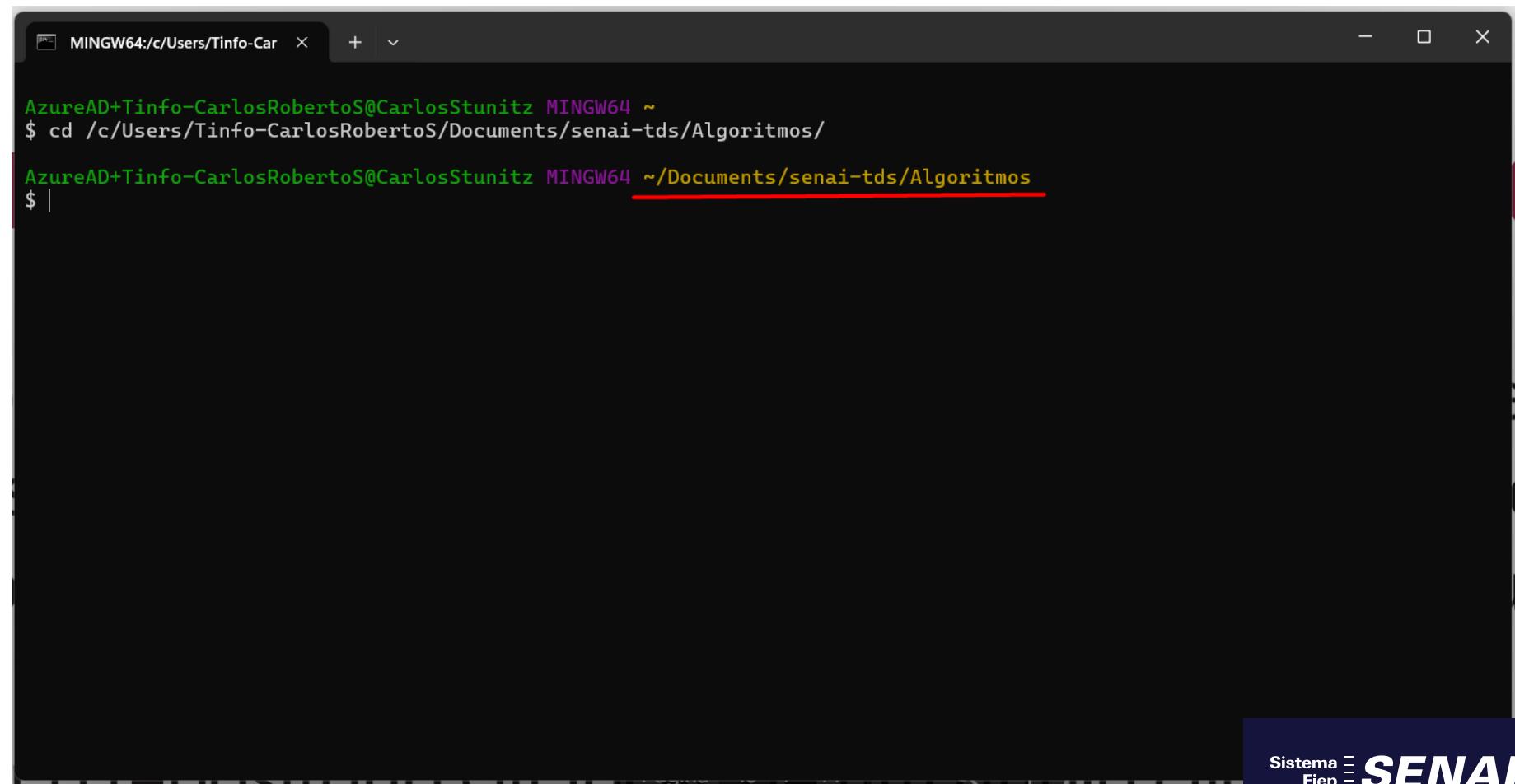
Criando o repositório local



```
MINGW64:/c/Users/Tinfo-Car ~
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/|
```

Controle de Versão e Colaboração

Criando o repositório local



```
MINGW64:/c/Users/Tinfo-Car ~
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ |
```

Controle de Versão e Colaboração

Criando o repositório local



A screenshot of a terminal window titled "MINGW64:c/Users/Tinfo-Car". The window shows the following command sequence:

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~  
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/  
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos  
$ pwd  
/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos  
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos  
$
```

Two red arrows point from the right side of the slide towards the "pwd" command and its output path.

Controle de Versão e Colaboração

Criando o repositório local



INICIAR O REPOSITÓRIO: COMANDO *git init*

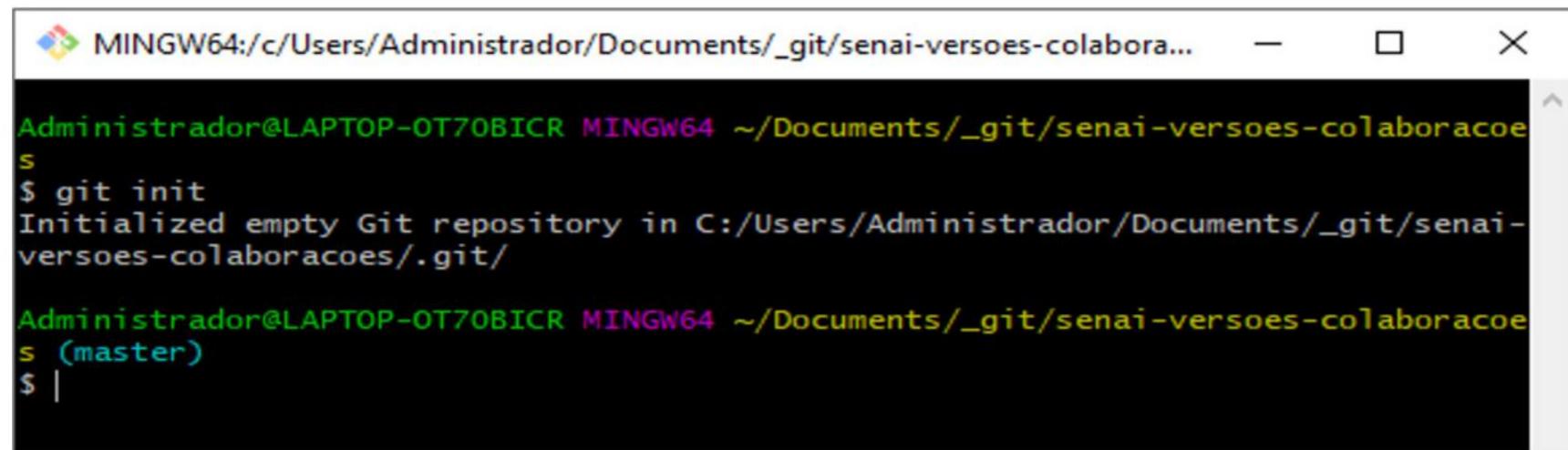
O primeiro comando que executaremos é o **git init**, que vai inicializar o repositório local em nossa pasta senai-versoes-colaboracoes.

Controle de Versão e Colaboração

Criando o repositório local

:::::

Ao ser executado, esse comando cria **uma pasta chamada .git, com subdiretórios (objects, refs/heads, refs/tags)**, e transforma o diretório atual em um repositório do Git. Na imagem a seguir, o terminal informa que o repositório está vazio, pois nenhum arquivo foi incluído.



```
MINGW64:/c/Users/Administrador/Documents/_git/senai-versoes-colaboracoes
Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracoes
$ git init
Initialized empty Git repository in C:/Users/Administrador/Documents/_git/senai-versoes-colaboracoes/.git/
Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracoes (master)
$ |
```

Controle de Versão e Colaboração

Criando o repositório local



The screenshot shows a terminal window titled "MINGW64:/c/Users/Tinfo-Car...". The user has navigated to the directory "/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/" and run the command "\$ git init". The output indicates that a new empty Git repository was initialized in the specified directory. Red arrows highlight the command "\$ git init", the resulting repository path "C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/", and the current branch "(master)".

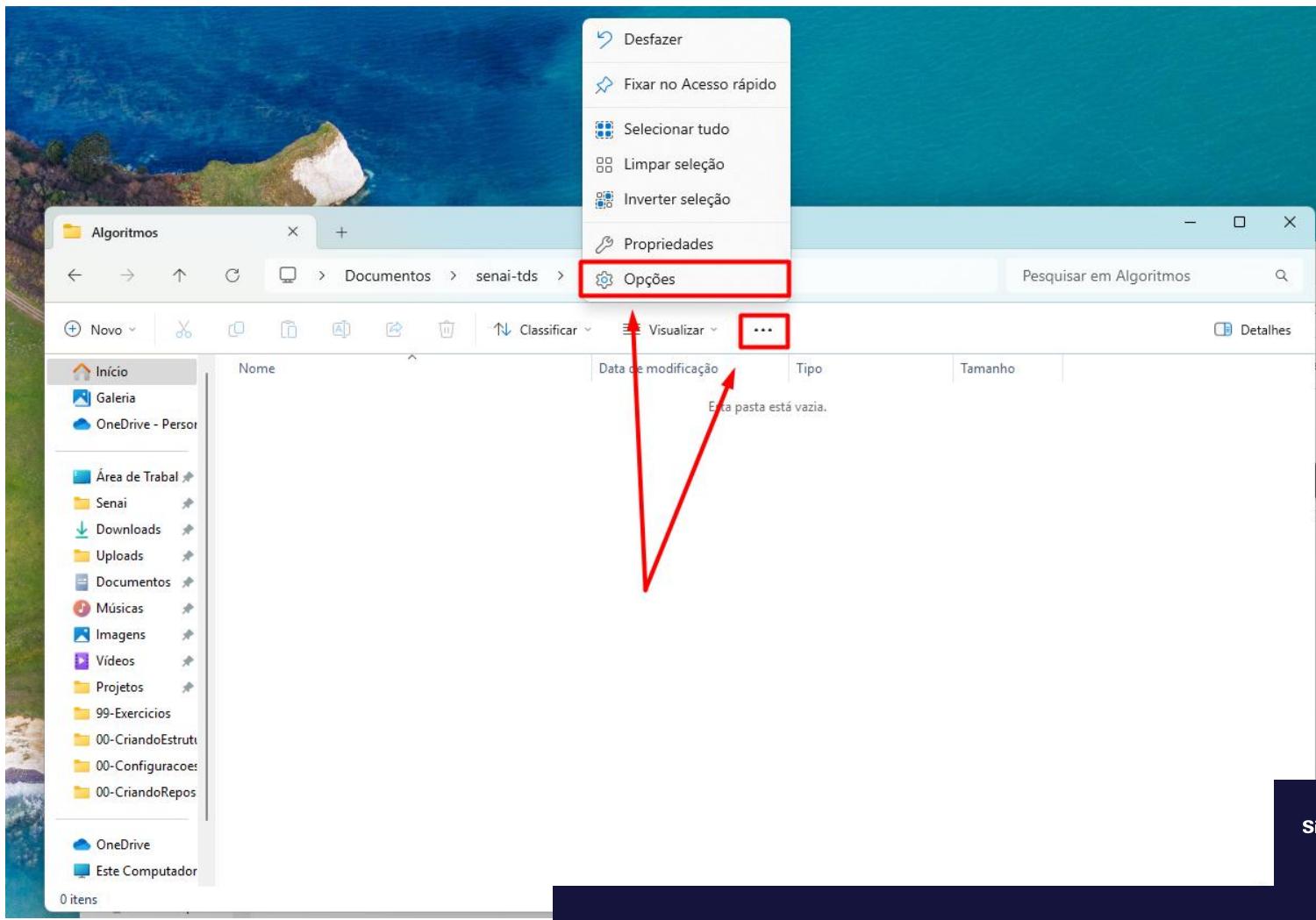
```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ pwd
/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ git init
Initialized empty Git repository in C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```

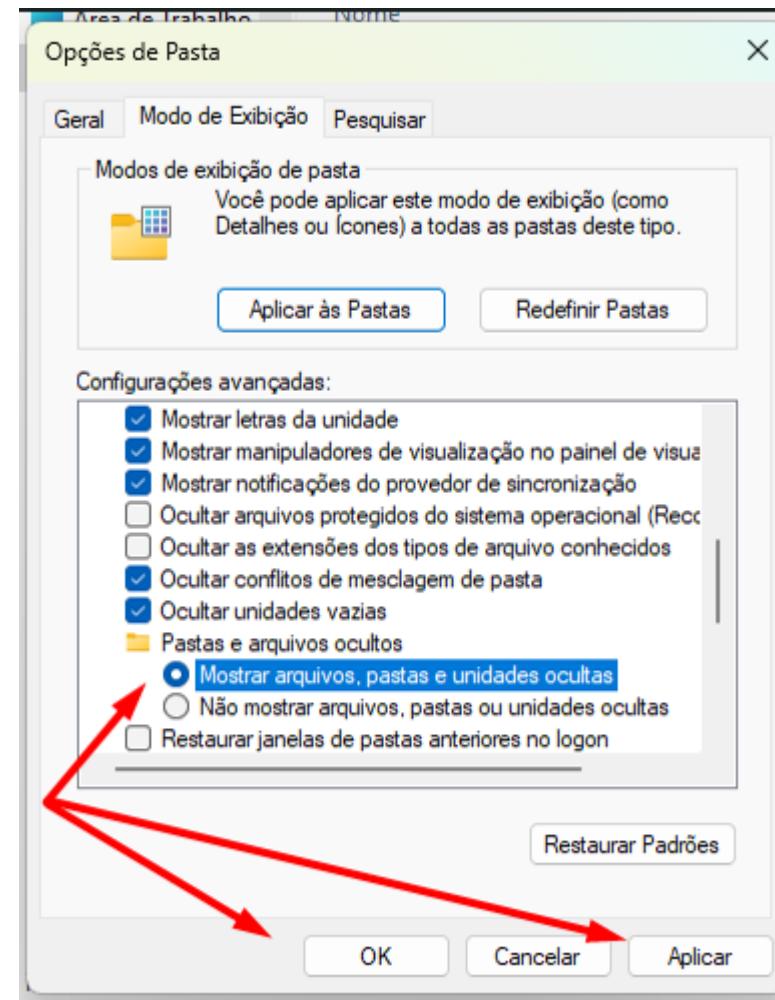
Controle de Versão e Colaboração

Criando o repositório local



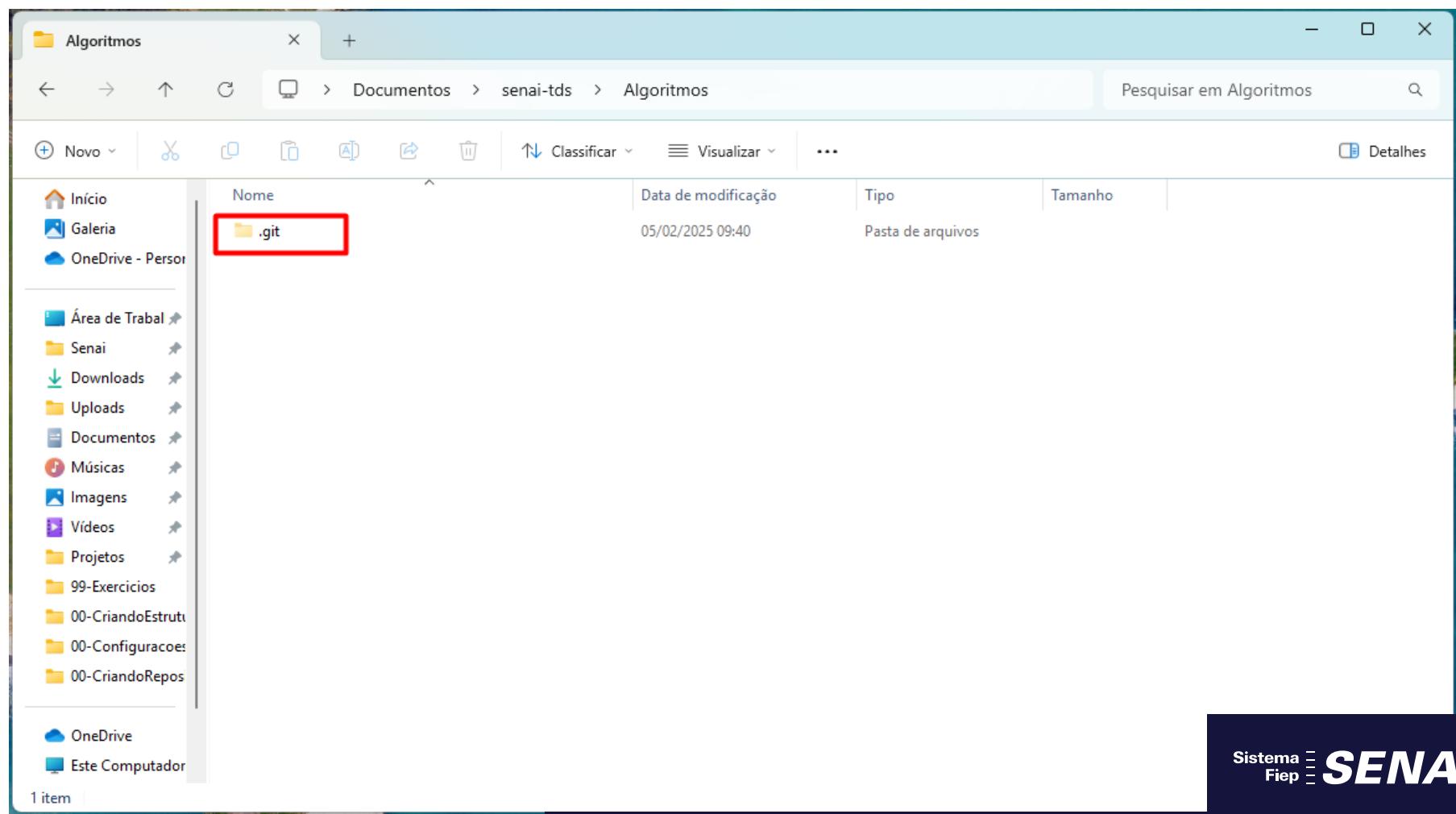
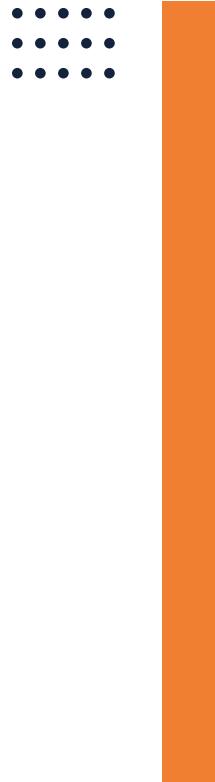
Controle de Versão e Colaboração

Criando o repositório local



Controle de Versão e Colaboração

Criando o repositório local



Controle de Versão e Colaboração

Criando o repositório local

:::::

Agora, a pasta Algoritmos é o repositório local e, dentro dela, há outra pasta chamada .git, criada com o comando **git init**.

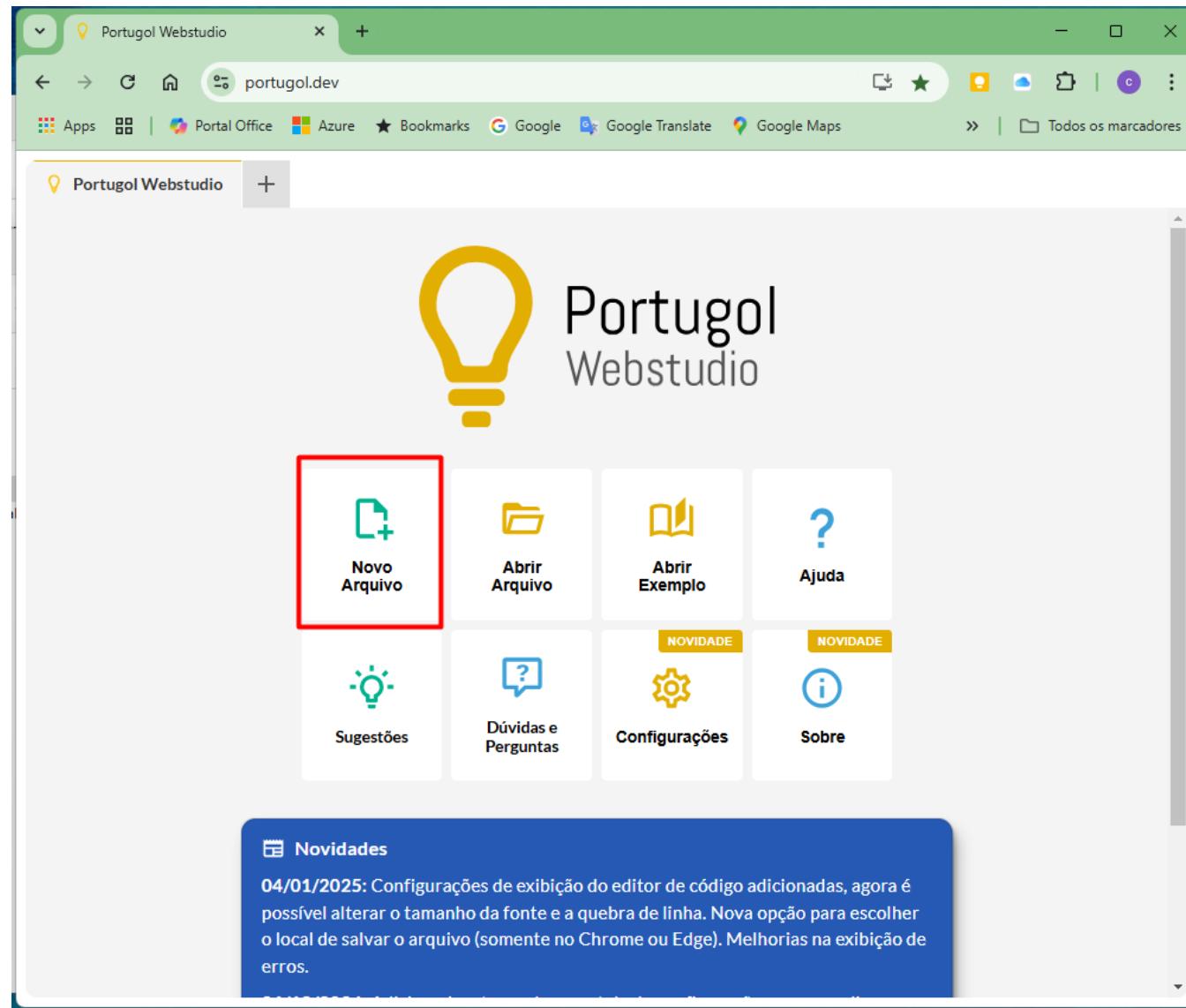




Controle de Versão e
Colaboração
Criando arquivo no repositório local

Controle de Versão e Colaboração

Criando arquivo no repositório local



Controle de Versão e Colaboração

Criando arquivo no repositório local



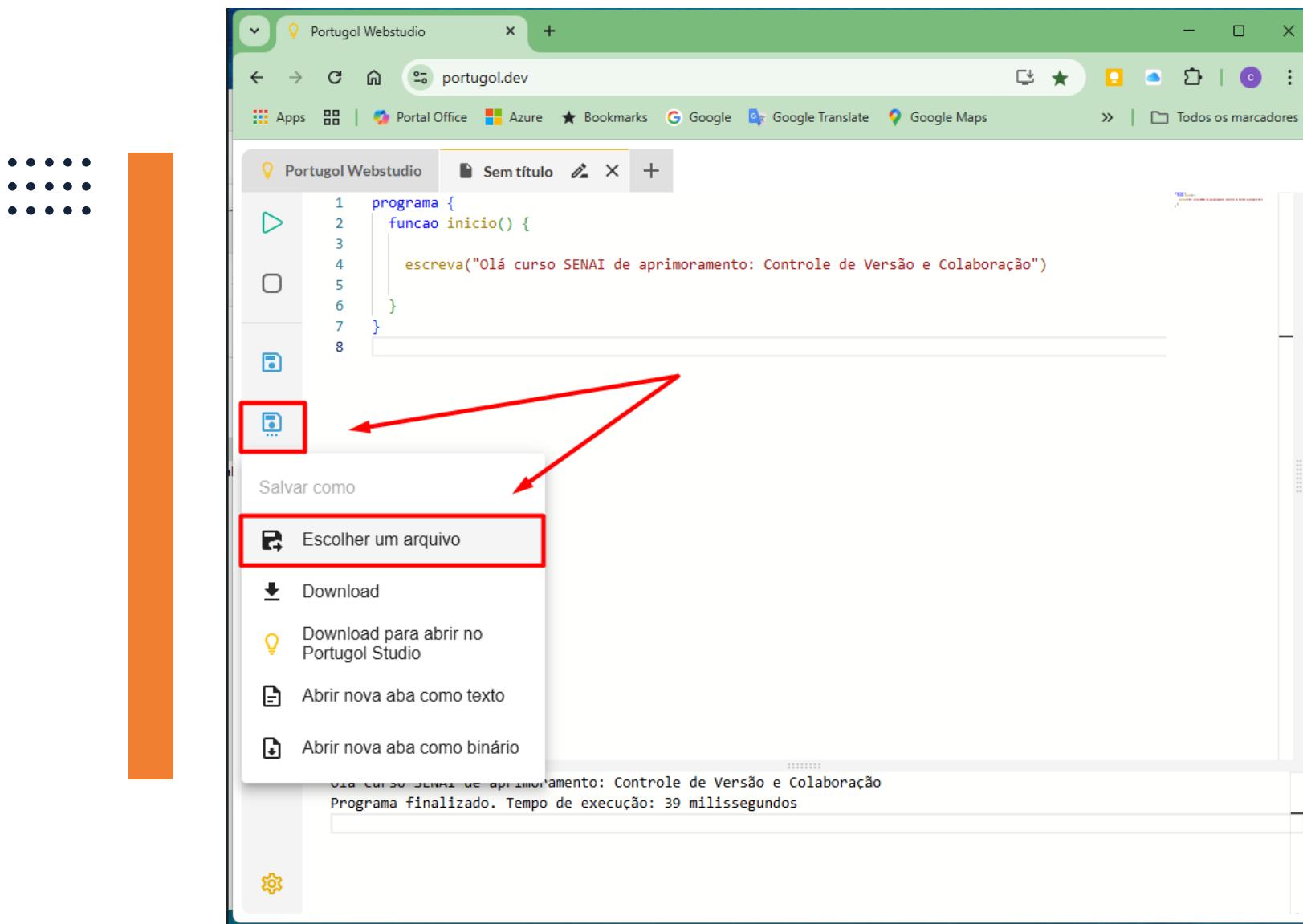
A screenshot of a web-based code editor window titled "Sem título". The code editor displays the following pseudocode:

```
1 programa {
2     funcao inicio() {
3
4         escreva("Olá curso SENAI de aprimoramento: Controle de Versão e Colaboração")
5
6     }
7 }
8 }
```

The browser's address bar shows "portugol.dev". The top navigation bar includes links for "Apps", "Portal Office", "Azure", "Bookmarks", "Google", "Google Translate", "Google Maps", and "Todos os marcadores".

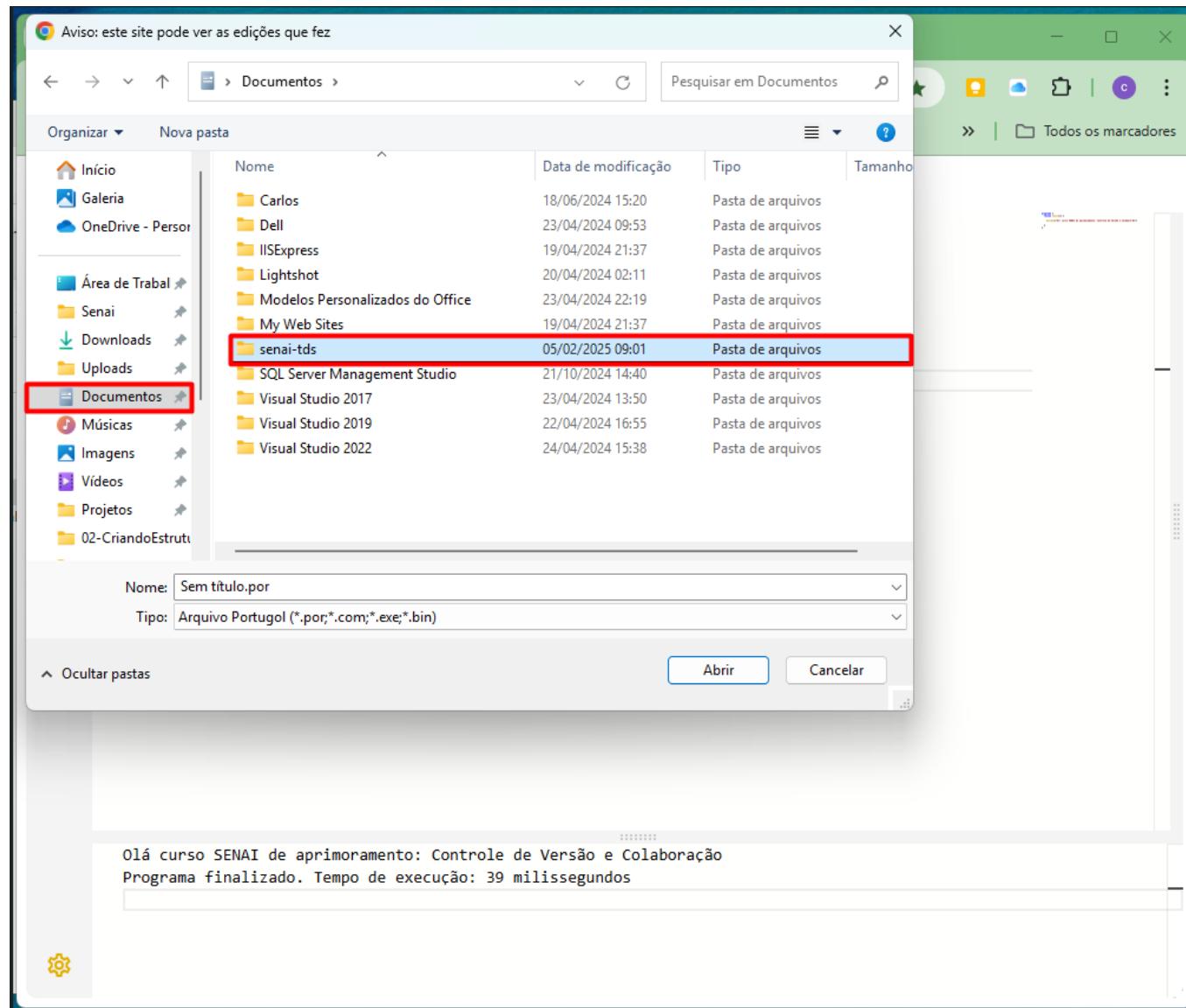
Controle de Versão e Colaboração

Criando arquivo no repositório local



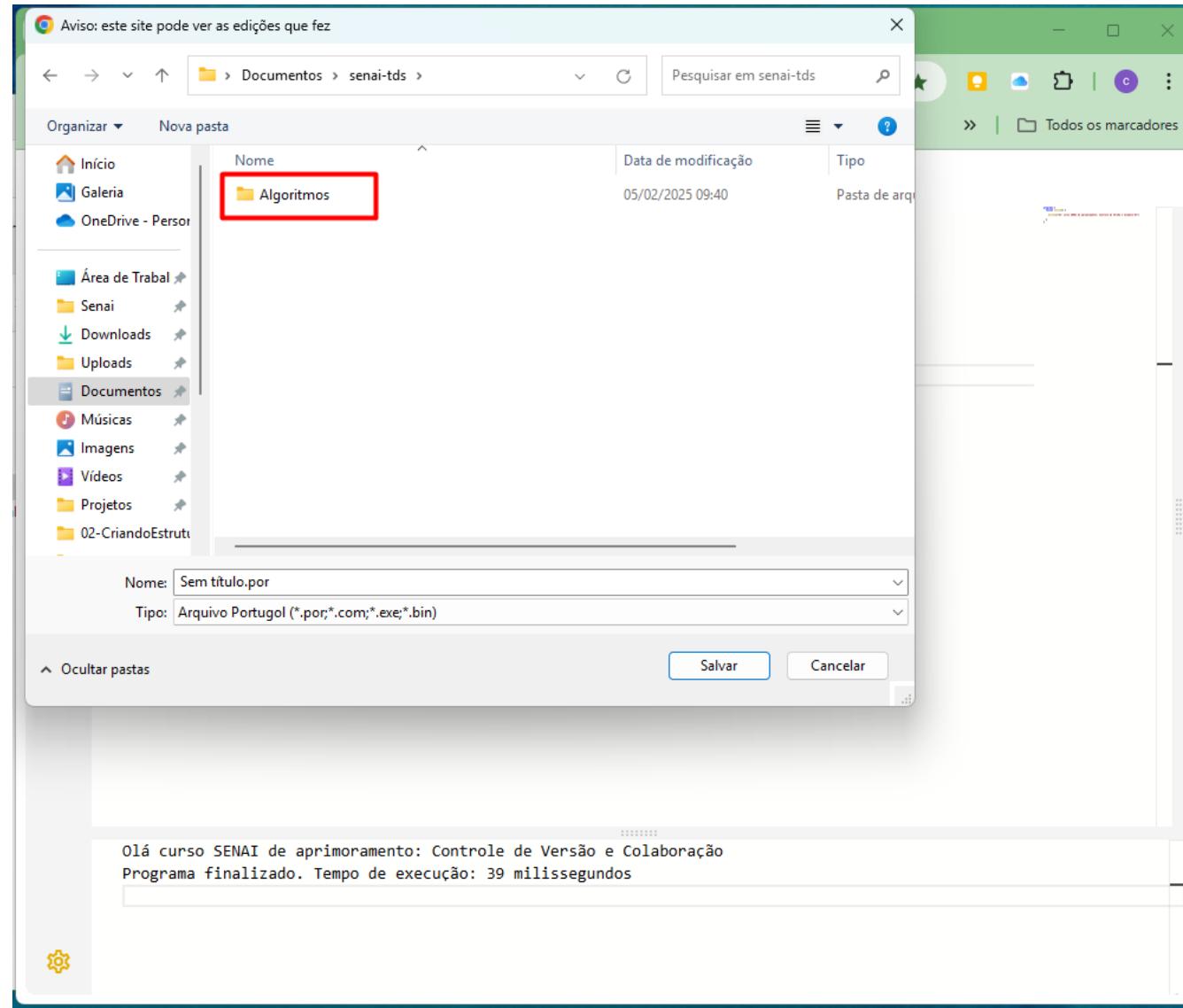
Controle de Versão e Colaboração

Criando arquivo no repositório local



Controle de Versão e Colaboração

Criando arquivo no repositório local



Controle de Versão e Colaboração

Criando arquivo no repositório local



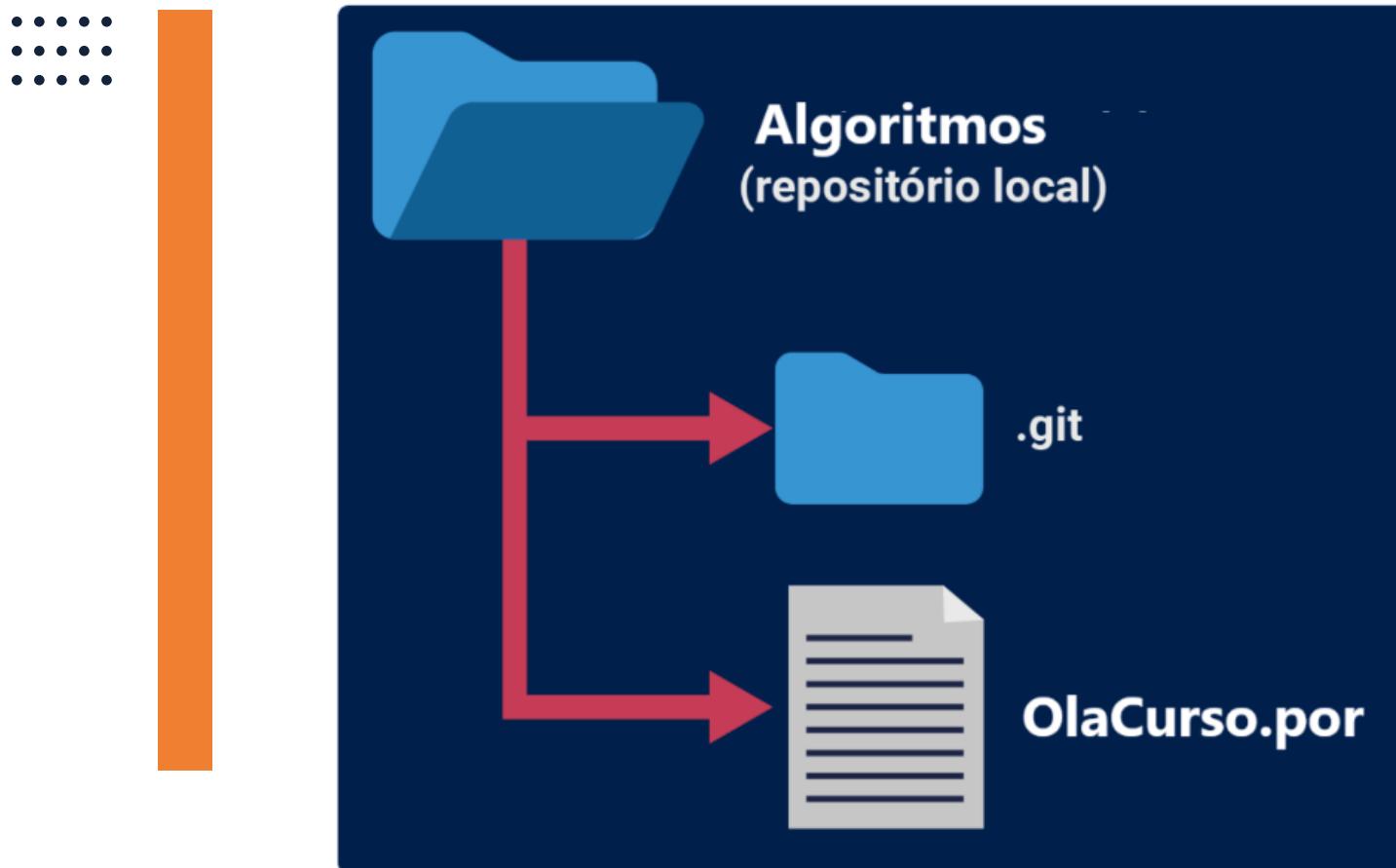
```
1 programa {
2     funcao inicio() {
3
4         escreva("Olá curso SENAI de aprimoramento: Controle de Versão e Colaboração")
5
6     }
7 }
8
```

Olá curso SENAI de aprimoramento: Controle de Versão e Colaboração
Programa finalizado. Tempo de execução: 39 milissegundos

Arquivo salvo com sucesso! OK

Controle de Versão e Colaboração

Criando o repositório local





Controle de Versão e
Colaboração
Rastreando arquivos modificados
git status



Controle de Versão e Colaboração

Rastreando arquivos modificados

.....

Para verificarmos o status das alterações que foram realizadas dentro de um repositório, utiliza-se o comando **git status**. Esse comando nos ajuda a verificar ao longo do tempo de desenvolvimento de seu projeto, quais arquivos estão sendo alterados por você em sua máquina dentro do repositório.

Controle de Versão e Colaboração

Rastreando arquivos modificados



```
MINGW64:/c/Users/Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ pwd
/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ git init
Initialized empty Git repository in C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
```

A red arrow points from the word "status" in the last command to the bottom right corner of the terminal window.

Controle de Versão e Colaboração

Rastreando arquivos modificados



The screenshot shows a terminal window titled "MINGW64:/c/Users/Tinfo-Car...". The user has navigated to the directory "/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/". They run "git init" which initializes an empty Git repository in the current directory. Then, they run "git status" which shows:

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ git init
Initialized empty Git repository in C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master

No commits yet

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

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

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```

A red rectangular box highlights the output of the "git status" command, specifically the section showing untracked files.

Controle de Versão e Colaboração

Rastreando arquivos modificados

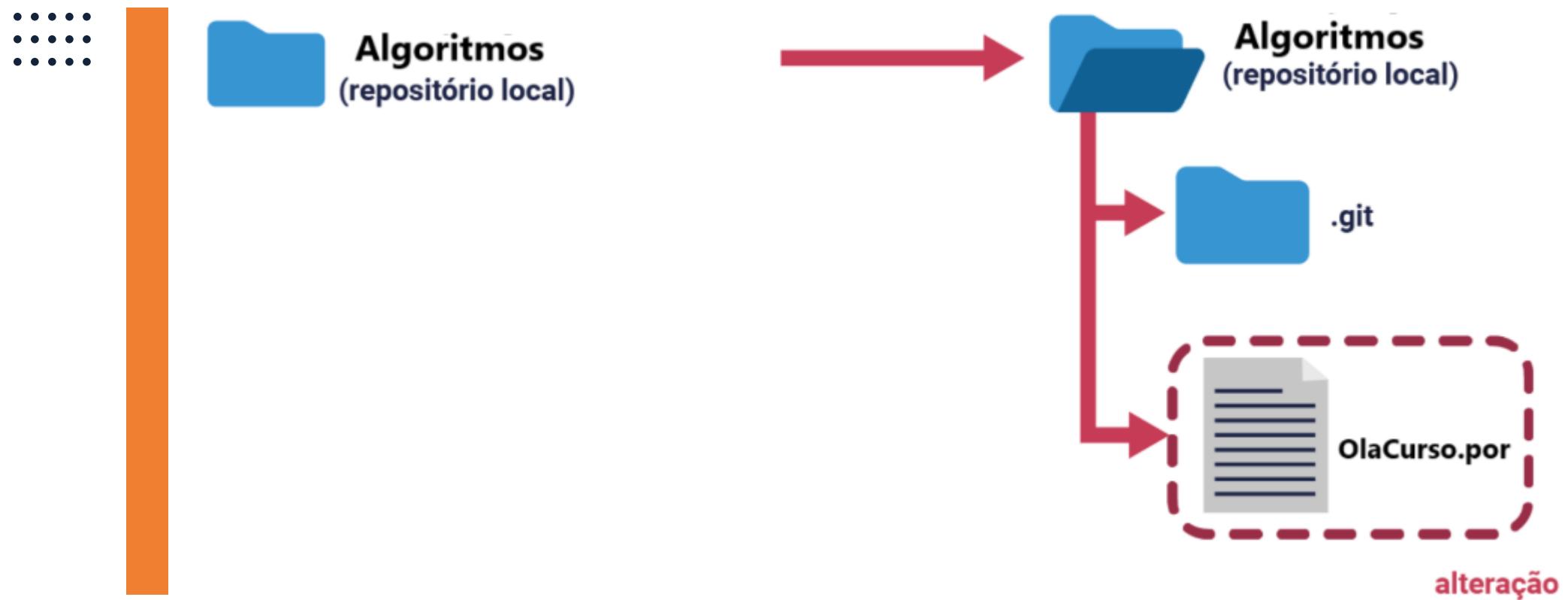
.....

Então, de um repositório Git vazio do início, agora temos o diretório com uma pasta .git, **que não conta como uma alteração** e que contém informações relacionadas a nosso versionamento e informações do nosso repositório, e um arquivo de texto chamado **OlaCurso.por**

Ao adicionar um arquivo, fizemos uma alteração.

Controle de Versão e Colaboração

Rastreando arquivos modificados



Controle de Versão e Colaboração

Adicionando arquivos em staging

git add

Controle de Versão e Colaboração

Adicionando arquivos em staging



O arquivo **OlaCurso.por** ainda não foi adicionado ao histórico do projeto. Para isso, primeiro devemos registrar a alteração com o comando **git add**, depois salvá-la no nosso repositório com o comando **git commit** e, aí sim, podemos considerar a alteração adicionada ao histórico. Depois do comando **git add**, as alterações vão para um ponto seguro, onde é possível verificar as alterações antes de salvá-las, chamado *Staging*.

Controle de Versão e Colaboração

Adicionando arquivos em staging

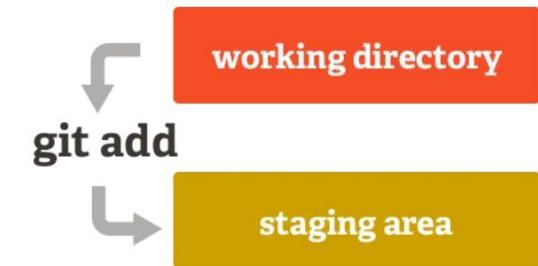


O comando **git add [nome-do-arquivo]** adiciona os arquivos modificados na área de *staging*, que é um ponto intermediário entre a máquina do desenvolvedor e o repositório de trabalho. A *staging* é uma área que você pode fazer uma revisão do que foi alterado antes de ser salvo no histórico de alterações.



Controle de Versão e Colaboração

Adicionando arquivos em staging



O comando **git add [nome-do-arquivo]** adiciona os arquivos modificados na área de *staging*, que é um ponto intermediário entre a máquina do desenvolvedor e o repositório de trabalho. A *staging* é uma área que você pode fazer uma revisão do que foi alterado antes de ser salvo no histórico de alterações.

Caso você tenha mais de um arquivo, você pode executar o comando **git add .** para adicionar todos os arquivos que foram modificados.

Controle de Versão e Colaboração

Adicionando arquivos em staging



```
MINGW64:/c/Users/Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ pwd
/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ git init
Initialized empty Git repository in C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master

No commits yet

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

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

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git add OlaCurso.por
```

A red arrow points from the bottom left towards the command `git add OlaCurso.por`.

Controle de Versão e Colaboração

Adicionando arquivos em staging



```
MINGW64:/c/Users/Tinfo-Car ~
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ pwd
/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ git init
Initialized empty Git repository in C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master

No commits yet

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

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

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git add OlaCurso.por
warning: in the working copy of 'OlaCurso.por', LF will be replaced by CRLF the next time Git touches it
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```

Controle de Versão e Colaboração

Adicionando arquivos em staging



VERIFICAR ALTERAÇÕES ANTES DE SALVAR: COMANDO `git status`

Para verificar essas alterações antes de confirmar, você pode utilizar o comando `git status`

As alterações no Git, diferentemente de outros tipos de versionamento de código, podem ser feitas localmente e depois publicadas em um repositório remoto (disponível online e para outras pessoas).

Controle de Versão e Colaboração

Adicionando arquivos em staging



The screenshot shows a terminal window titled "MINGW64:c/Users/Tinfo-Car" with the following command history:

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ cd /c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ pwd
/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ git init
Initialized empty Git repository in C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master

No commits yet

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

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

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git add OlaCurso.por
warning: in the working copy of 'OlaCurso.por', LF will be replaced by CRLF the next time Git touches it

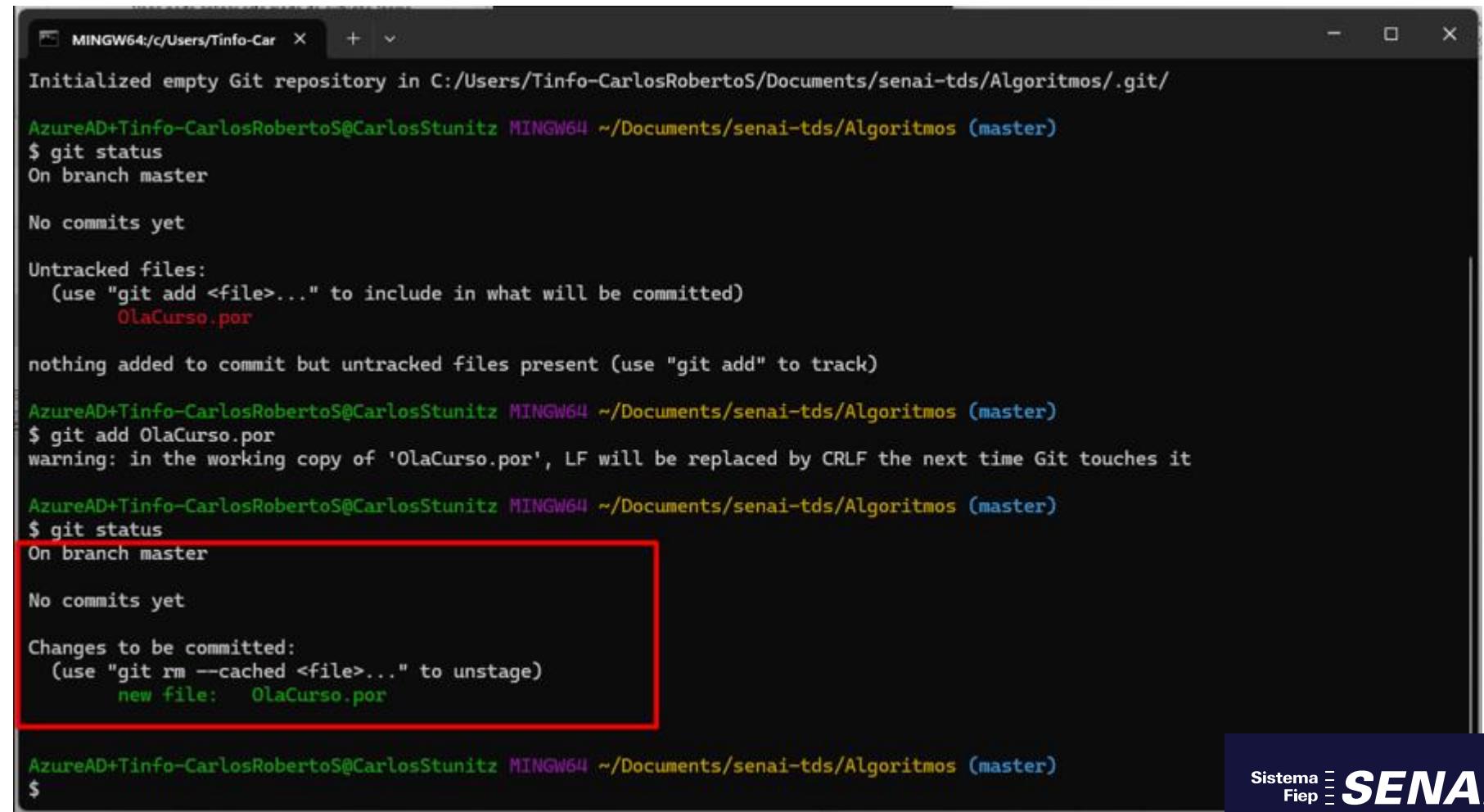
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
```

A red arrow points from the bottom of the terminal window towards the bottom-left corner of the slide.

Controle de Versão e Colaboração

Adicionando arquivos em staging

• • • •
• • • •
• • • •



```
MINGW64:/c/Users/Tinfo-Car X + v
Initialized empty Git repository in C:/Users/Tinfo-CarlosRobertoS/Documents/senai-tds/Algoritmos/.git/
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master

No commits yet

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

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

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git add OlaCurso.por
warning: in the working copy of 'OlaCurso.por', LF will be replaced by CRLF the next time Git touches it

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master

No commits yet

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

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```

Controle de Versão e Colaboração

Adicionando arquivos em staging



Controle de Versão e Colaboração

Adicionando alterações no repositório `git commit`



Controle de Versão e Colaboração

Adicionando alterações no repositório



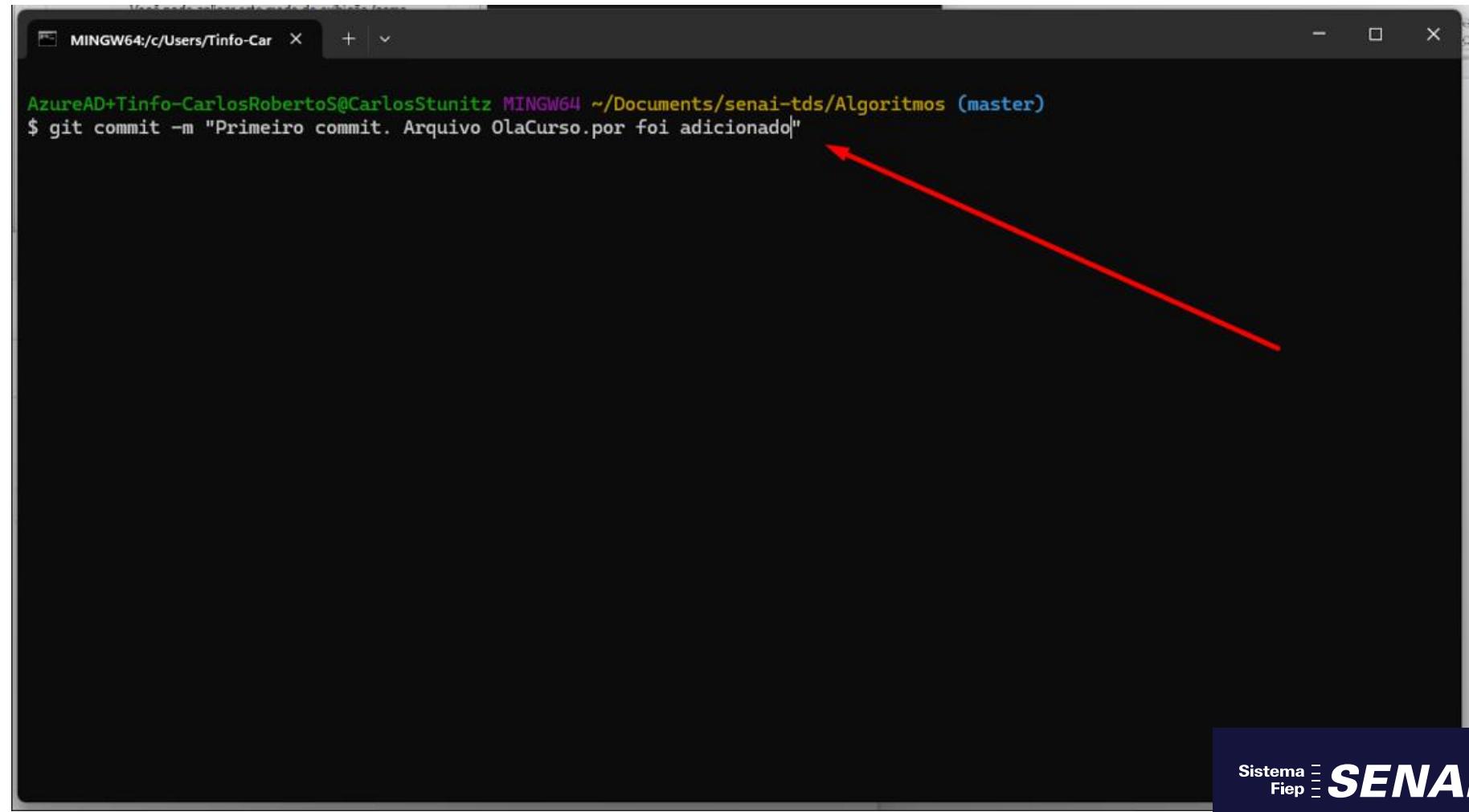
SALVAR ALTERAÇÕES: COMANDO *git commit*

Um dos passos mais importantes é confirmar as alterações do fluxo de trabalho em versões no histórico do repositório. O Git irá confirmar a captura do diretório de *staging* (isto é, a captura de todas as alterações adicionadas no **git add**) e salvar no histórico de confirmação de repositórios, completando o *commit*.



Controle de Versão e Colaboração

Adicionando alterações no repositório



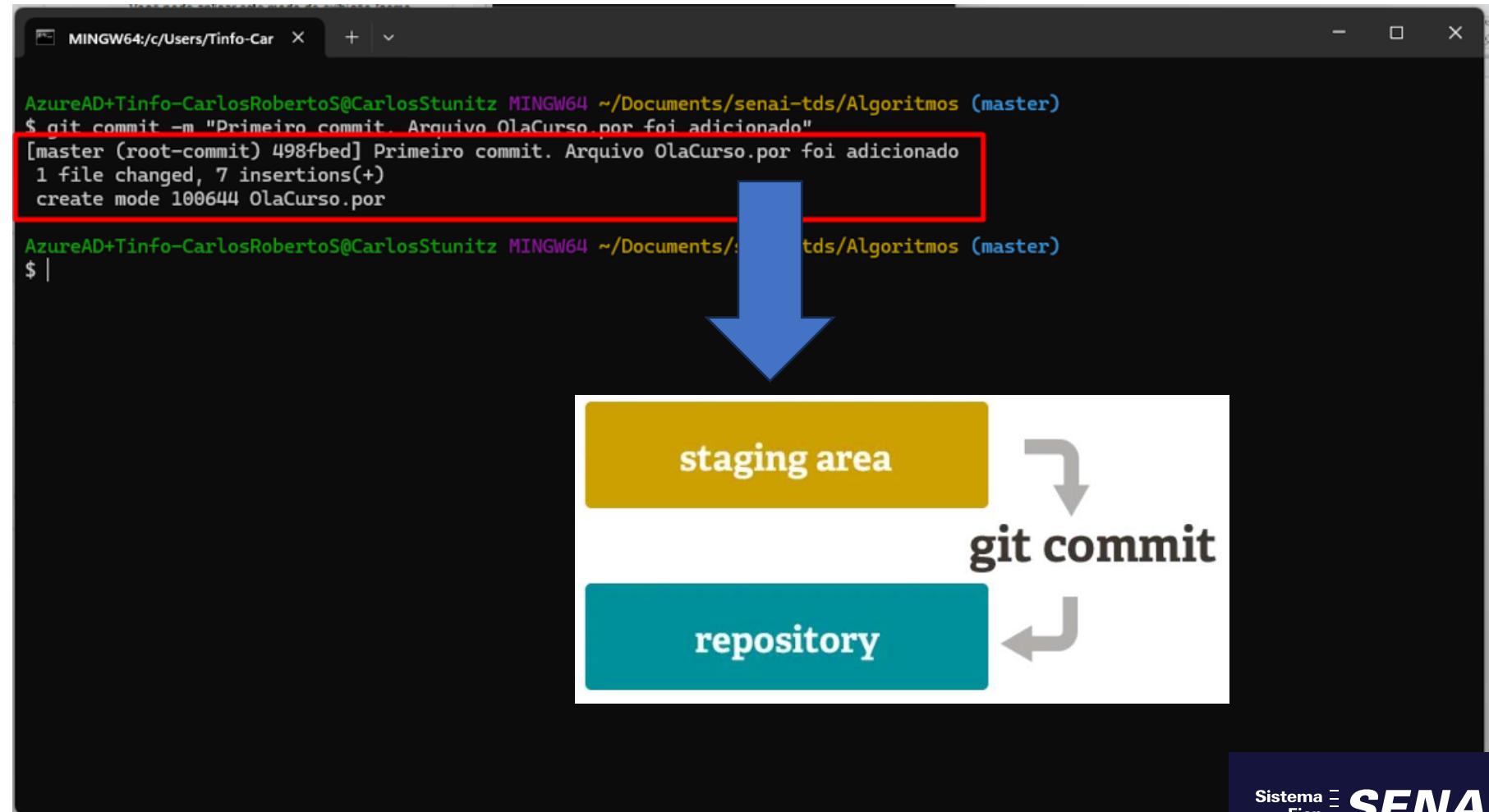
A screenshot of a terminal window titled "MINGW64:c/Users/Tinfo-Car...". The window shows the command:

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git commit -m "Primeiro commit. Arquivo OlaCurso.por foi adicionado!"
```

A red arrow points from the text "Arquivo OlaCurso.por foi adicionado!" towards the bottom right corner of the slide.

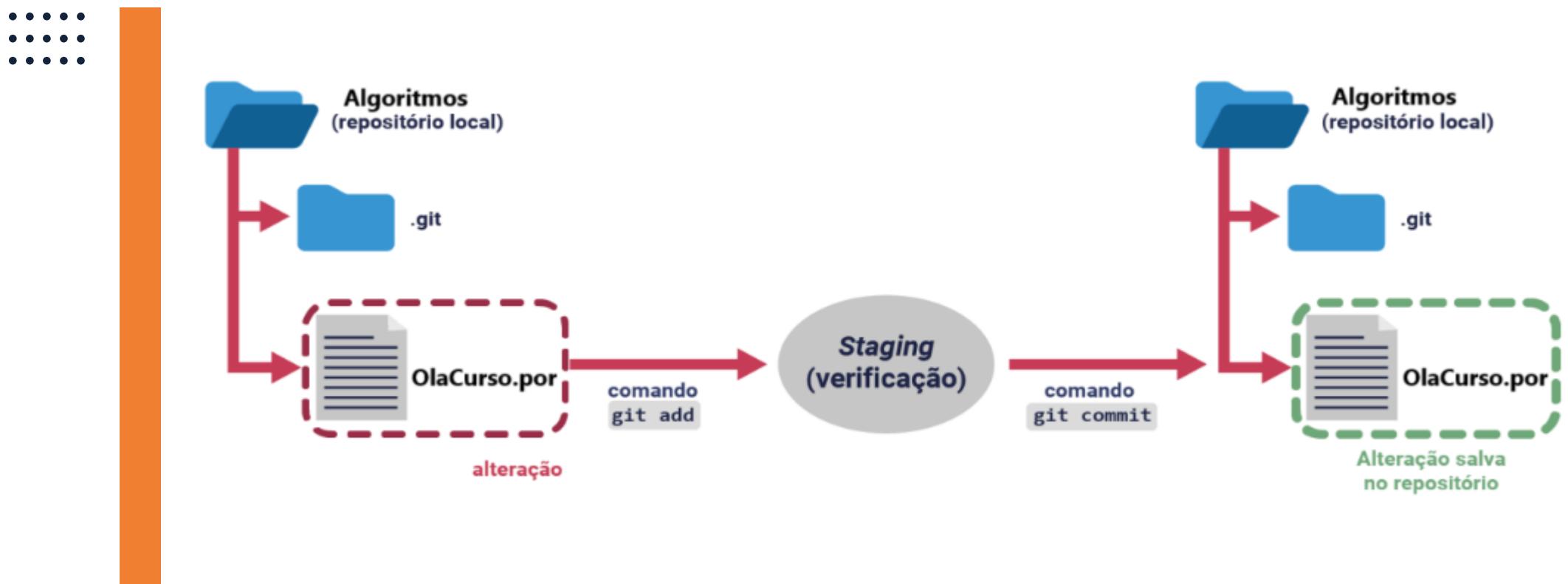
Controle de Versão e Colaboração

Adicionando alterações no repositório



Controle de Versão e Colaboração

Adicionando alterações no repositório

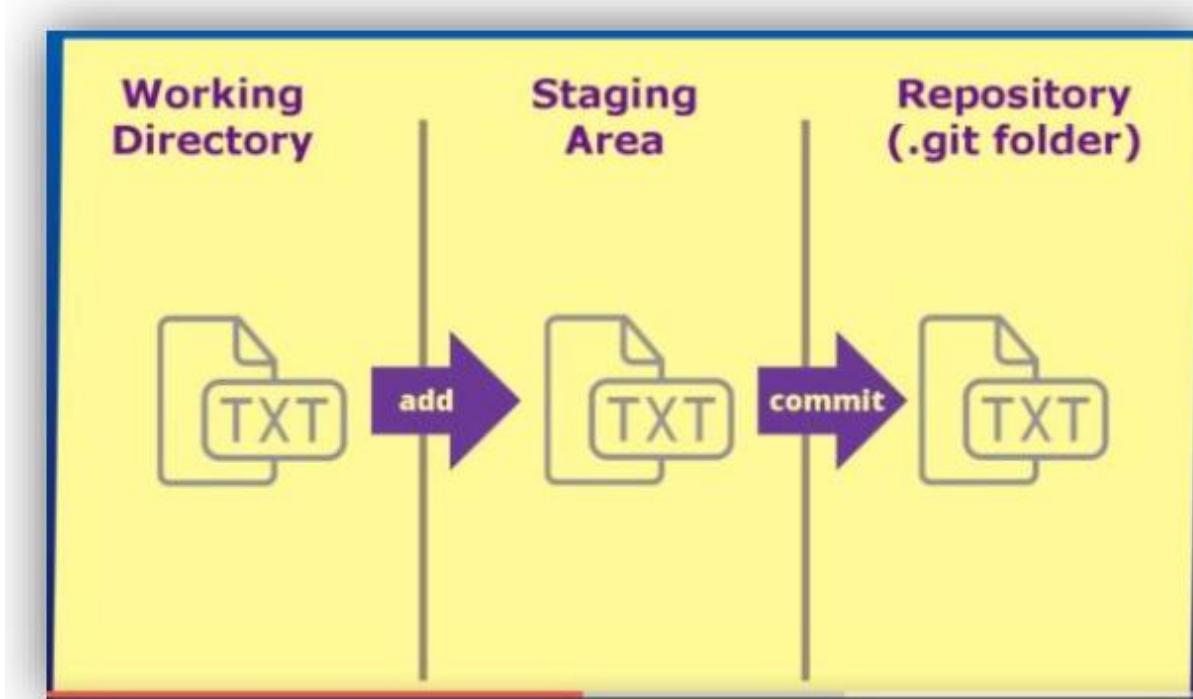


Controle de Versão e Colaboração

Adicionando alterações no repositório



Fluxo de trabalho local



Controle de Versão e Colaboração

Visualizando log da alteração

git log / git show

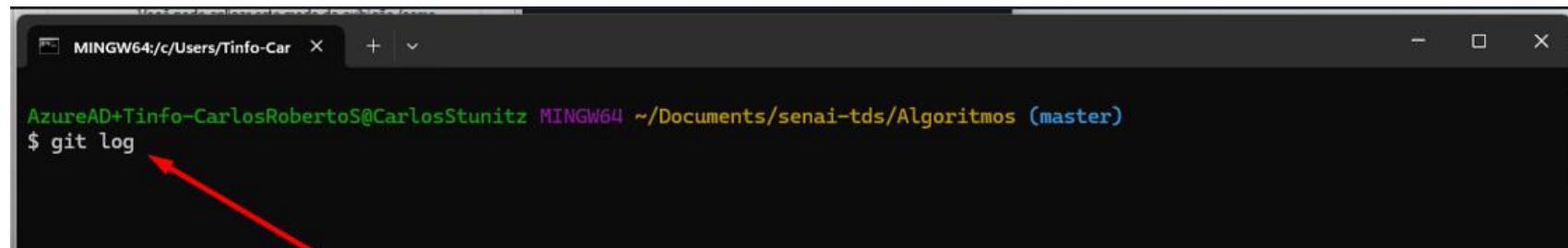
Controle de Versão e Colaboração

Visualizando log da alteração

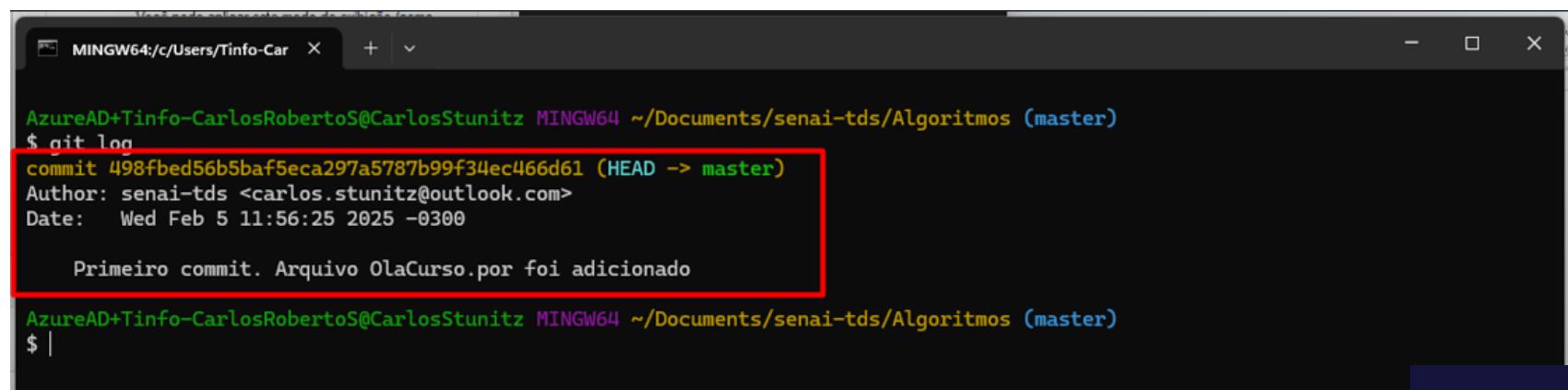
VISUALIZAR LOG DA ALTERAÇÃO: COMANDO *git log*

• • • •
• • • •
• • • •

Para visualizar as alterações feitas no seu repositório de trabalho, você pode executar o **git log** no terminal.



```
MINGW64:/c/Users/Tinfo-Car X + ▾ AzureAD+Tinfo-Car@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git log
```



```
MINGW64:/c/Users/Tinfo-Car X + ▾ AzureAD+Tinfo-Car@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git log
commit 498fbed56b5baf5eca297a5787b99f34ec466d61 (HEAD -> master)
Author: senai-tds <carlos.stunitz@outlook.com>
Date:   Wed Feb 5 11:56:25 2025 -0300

    Primeiro commit. Arquivo OlaCurso.por foi adicionado

AzureAD+Tinfo-Car@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ |
```

Controle de Versão e Colaboração

Visualizando log da alteração

Para visualizar as alterações ao longo do projeto, usamos o comando **git show** **número-do-commit**.

```
MINGW64:/c/Users/Tinfo-Car X + ▾ AzureAD+Tinfo-CarosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git show
```

```
MINGW64:/c/Users/Tinfo-Car X + ▾ AzureAD+Tinfo-CarosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git show
commit 498fbed56b5baf5eca297a5787b99f34ec466d61 (HEAD -> master)
Author: senai-tds <carlos.stunitz@outlook.com>
Date:   Wed Feb 5 11:56:25 2025 -0300

    Primeiro commit. Arquivo OlaCurso.por foi adicionado

diff --git a/OlaCurso.por b/OlaCurso.por
new file mode 100644
index 0000000..0927472
--- /dev/null
+++ b/OlaCurso.por
@@ -0,0 +1,7 @@
+programa {
+    funcao inicio() {
+        escreva("Olá curso SENAI de aprimoramento: Controle de Versão e Colaboração")
+    }
+}
```

Controle de Versão e Colaboração

Criando repositório remoto



Controle de Versão e Colaboração

Criando repositório remoto



CRIANDO UM REPOSITÓRIO ONLINE



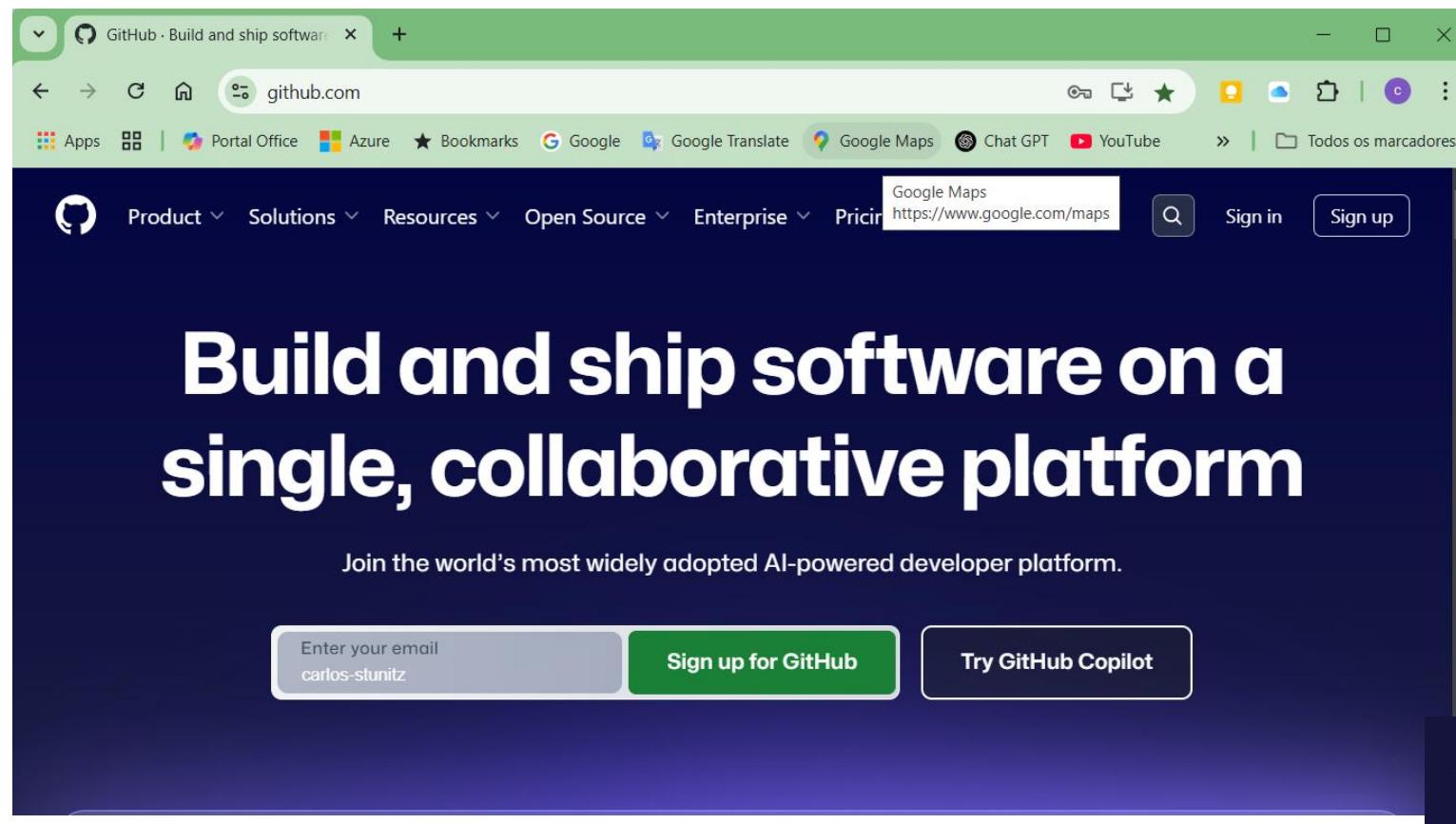
Para disponibilizar o repositório local, temos que utilizar um repositório online e, para isso, vamos utilizar a plataforma GitHub.

Controle de Versão e Colaboração

Criando repositório remoto



Acessar <https://github.com>



Controle de Versão e Colaboração

Criando repositório remoto



Fazer login (sign in) #1/2

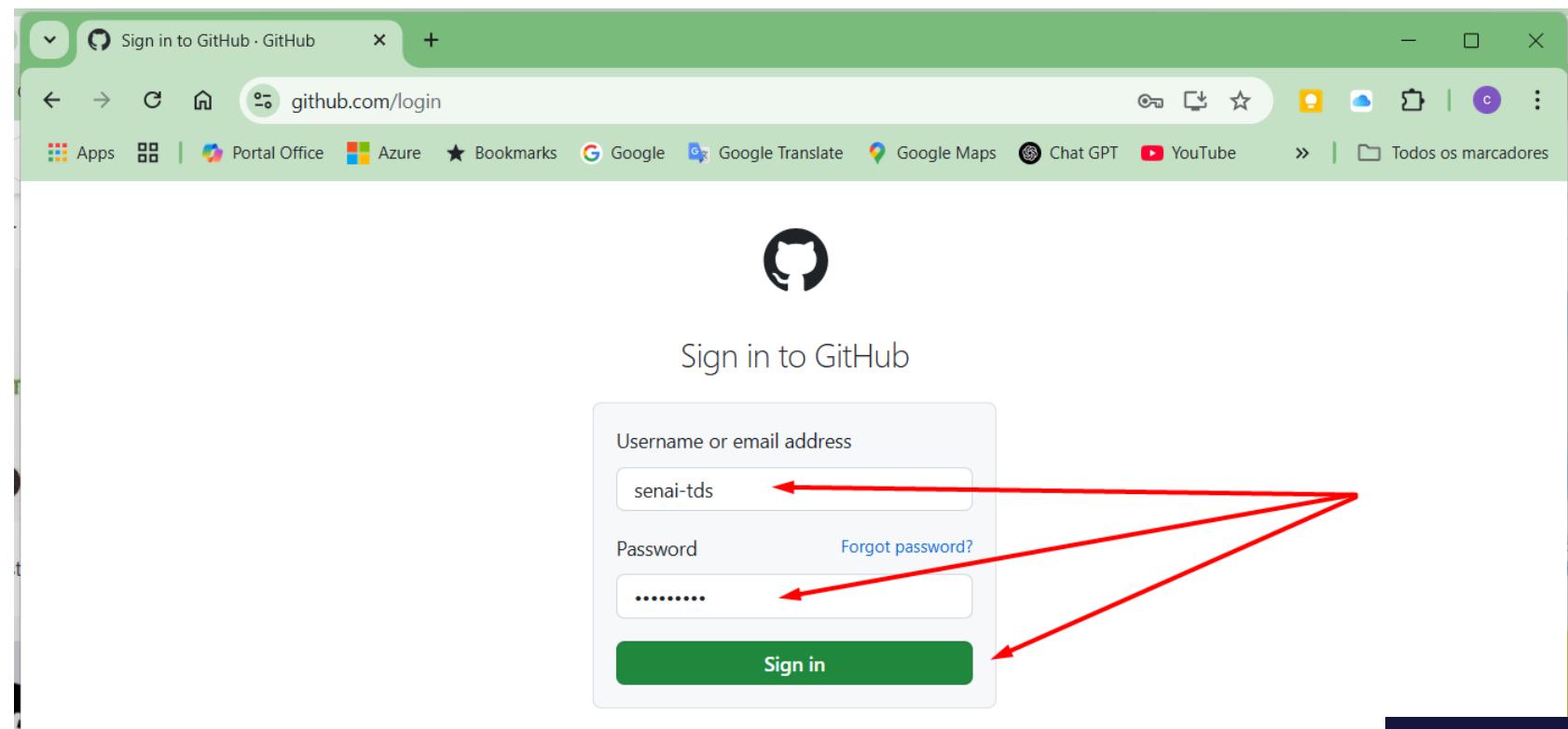
The screenshot shows a web browser window with the GitHub homepage loaded. The URL 'github.com' is visible in the address bar. The page features a dark blue header with navigation links like 'Product', 'Solutions', 'Resources', 'Open Source', 'Enterprise', and 'Pricing'. On the right side of the header, there is a 'Sign in' button highlighted with a red box and a red arrow pointing to it from below. The main content area has a large white text overlay that reads 'Build and ship software on a single, collaborative platform'. Below this, a smaller text says 'Join the world's most widely adopted AI-powered developer platform.' At the bottom, there is a form with a placeholder 'Enter your email' containing 'carlos-stunitz', a green 'Sign up for GitHub' button, and a white 'Try GitHub Copilot' button.

Controle de Versão e Colaboração

Criando repositório remoto

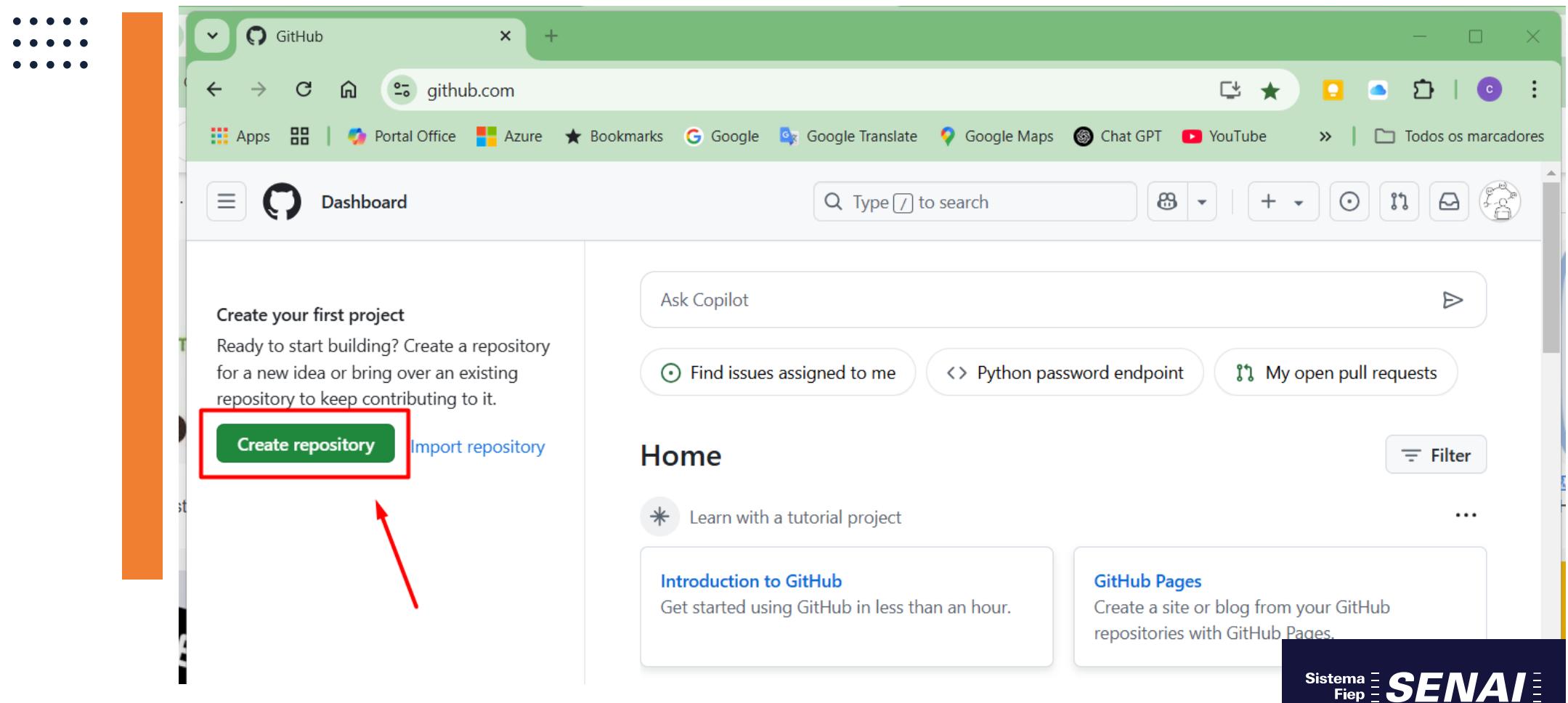


Fazer login (sign in) #2/2



Controle de Versão e Colaboração

Criando repositório remoto



Controle de Versão e Colaboração

Criando repositório remoto



The screenshot shows the GitHub interface for creating a new repository. The URL in the address bar is `github.com/new`. The main title is "Create a new repository". Below it, a sub-instruction says "A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)". A note below that says "Required fields are marked with an asterisk (*)." The "Owner" field is set to "senai-tds" and the "Repository name" field is "Algoritmos". A red arrow points from the "Repository name" field to the "Algoritmos" text. Below the name, a green checkmark says "Algoritmos is available.". Another red arrow points from the "Description (optional)" field to the text "Repositório de Algoritmos do curso de aperfeiçoamento Controle de Versão e Colaboração do SENAI". At the bottom, there are two radio button options: "Public" (selected) and "Private". A red arrow points from the "Public" radio button to the explanatory text "Anyone on the internet can see this repository. You choose who can commit." The "Private" option has its own explanatory text "You choose who can see and commit to this repository.".

Controle de Versão e Colaboração

Criando repositório remoto



New repository

github.com/new

Repositorio de Algoritmos do curso de aperteiçoamento Controle de Versao e Colaboração do SENAI

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None

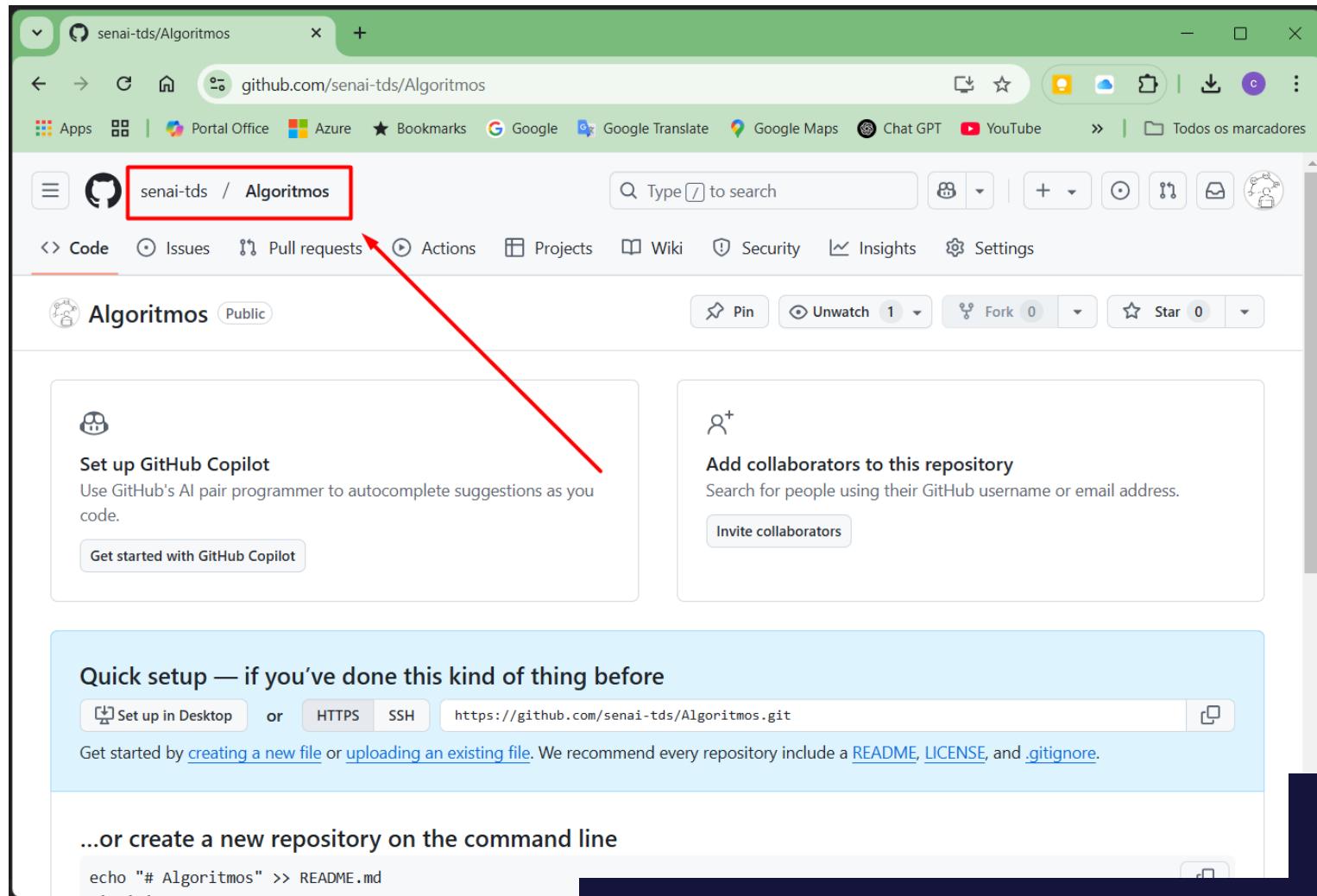
A license tells others what they can and can't do with your code. [Learn more about licenses](#).

ⓘ You are creating a public repository in your personal account.

Create repository

Controle de Versão e Colaboração

Criando repositório remoto



Controle de Versão e Colaboração

Publicando no repositório remoto

Controle de Versão e Colaboração

Publicando repositório remoto



PUBLICANDO NO REPOSITÓRIO ONLINE



Você criou um repositório online, mas não especificou que é neste repositório que gostaria de publicar seu trabalho.

Controle de Versão e Colaboração

Publicando repositório remoto

.....

Depois de criar o repositório remoto, precisamos fazer a ligação com o repositório local. Para isso, no terminal local, vamos:

- informar a pasta remota: **git remote add origin git@github.com:hstrada/senai-versoes-colaboracoes.git** (lembre-se de trocar o usuário no comando);
- visualizar o repositório remoto: **git remote -v**; e
- publicar as alterações no repositório remoto: **git push -u origin master**.

No repositório remoto, vamos:

- autorizar o usuário.

Depois, vamos verificar se todas as etapas foram completadas corretamente, então, devemos:

- no repositório local: visualizar a autenticação de usuário feita no repositório remoto;
- no repositório remoto: visualizar a publicação feita.

Controle de Versão e Colaboração

Publicando repositório remoto



INFORMAR NO REPOSITÓRIO LOCAL A PASTA REMOTA: *git remote add origin git@github.com:hstrada/senai-versoes-colaboracoes.git*

Para realizar a ligação dos repositórios, primeiro é necessário informar no repositório local que o repositório remoto que você deseja trabalhar é aquele criado no GitHub, através do comando **git remote add origin git@github.com:hstrada/senai-versoes-colaboracoes.git**.

Controle de Versão e Colaboração

Publicando repositório remoto



Obter informações da pasta remota

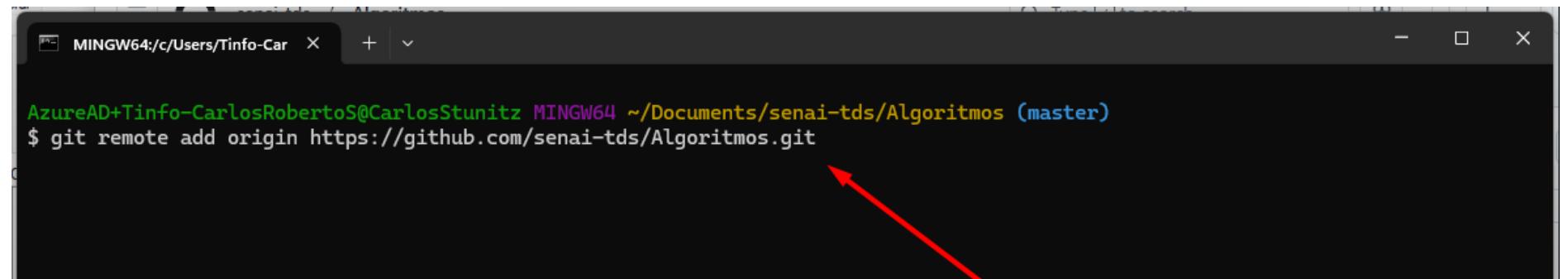
The screenshot shows a web browser window with the URL <https://github.com/senai-tds/Algoritmos>. The page displays the repository 'Algoritmos' under the user 'senai-tds'. The 'Code' tab is selected. At the top, there are sections for 'Set up GitHub Copilot' and 'Add collaborators to this repository'. Below these, a 'Quick setup' section provides instructions for cloning the repository using 'Set up in Desktop', 'HTTPS', or 'SSH', with the 'HTTPS' option highlighted and its URL (<https://github.com/senai-tds/Algoritmos.git>) boxed in red. A note below suggests creating a new file or uploading an existing one, mentioning README, LICENSE, and .gitignore. Further down, there's a command-line guide for initializing a new repository with commands like 'git init', 'git add README.md', 'git commit -m "first commit"', 'git branch -M main', 'git remote add origin https://github.com/senai-tds/Algoritmos.git', and 'git push -u origin main'. At the bottom, there's another command-line section for pushing an existing repository.

Controle de Versão e Colaboração

Publicando repositório remoto



Informar no repositório local a pasta remota



```
MINGW64:/c/Users/Tinfo-Car ~Documents/senai-tds/Algoritmos (master)
$ git remote add origin https://github.com/senai-tds/Algoritmos.git
```

Controle de Versão e Colaboração

Publicando repositório remoto



VISUALIZAR O REPOSITÓRIO REMOTO: `git remote -v`

Para visualizar o repositório remoto informado e visualizar as informações, digite no terminal `git remote -v`.

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git remote -v
origin https://github.com/senai-tds/Algoritmos.git (fetch)
origin https://github.com/senai-tds/Algoritmos.git (push)
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ |
```

Controle de Versão e Colaboração

Publicando repositório remoto

:::::

PUBLICAR AS ALTERAÇÕES NO REPOSITÓRIO REMOTO: *git push -u origin master*

Para publicar as alterações no repositório remoto, execute o comando **git push -u origin master**.

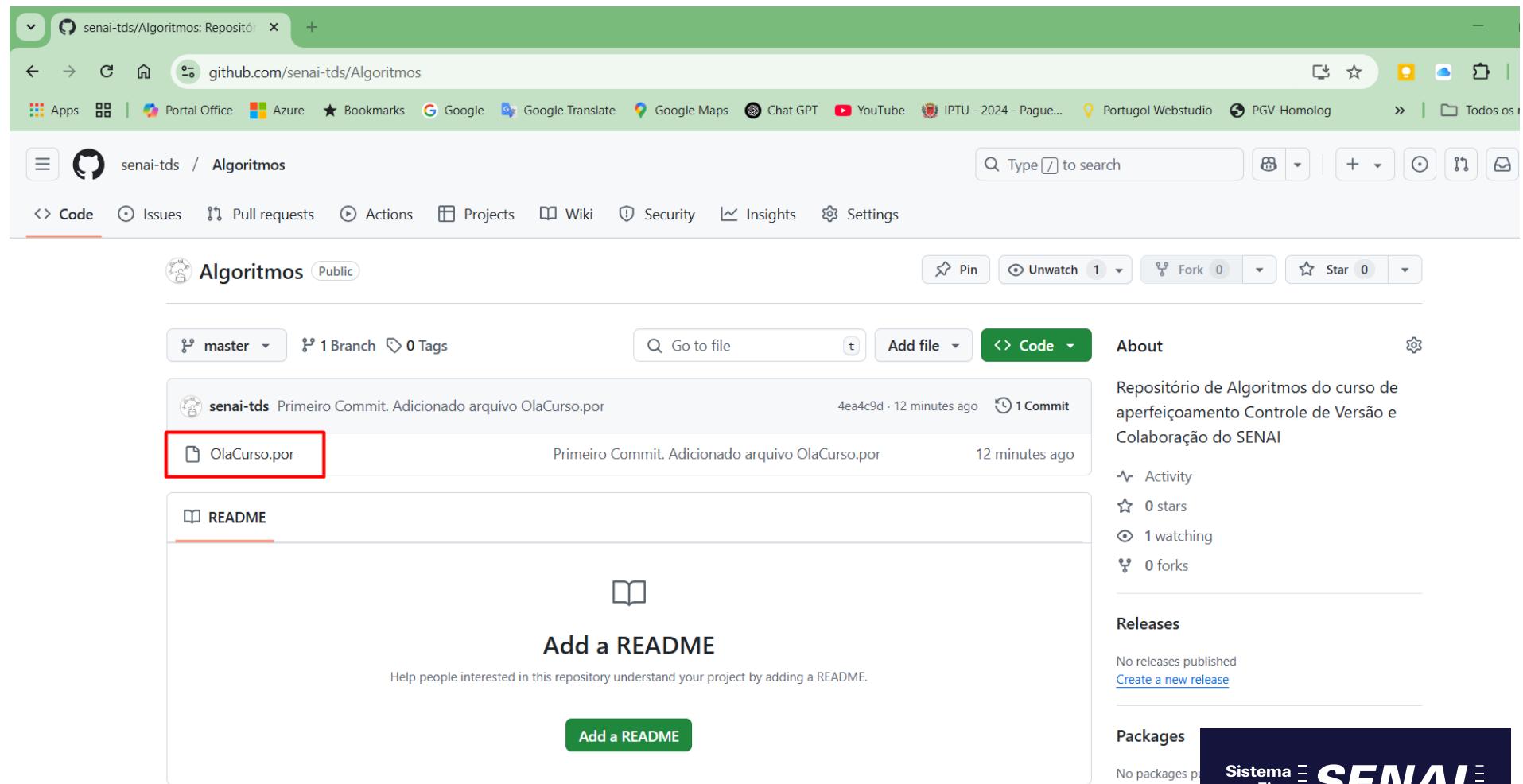
```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 366 bytes | 366.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/senai-tds/Algoritmos.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Controle de Versão e Colaboração

Verificando publicação no repositório remoto

Controle de Versão e Colaboração

Publicando repositório remoto



A screenshot of a GitHub repository page for 'senai-tds/Algoritmos'. The repository is public and contains one branch ('master') and one commit. The commit was made by 'senai-tds' and added the file 'OlaCurso.por'. The file 'OlaCurso.por' is highlighted with a red box. Below the file list, there is a section to 'Add a README'.

The GitHub interface includes a navigation bar with links like Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. On the right side, there is an 'About' section with details about the repository, such as its purpose ('Repositório de Algoritmos do curso de aperfeiçoamento Controle de Versão e Colaboração do SENAI'), activity (0 stars, 1 watching, 0 forks), releases (none published), and packages (none published).



A dark blue-tinted photograph of two people from behind, looking at a laptop screen. One person's hands are visible on the keyboard.

Controle de Versão e Colaboração

Fluxo de trabalho



Controle de Versão e Colaboração

Fluxo de trabalho

.....

A imagem a seguir mostra o fluxo atual de trabalho com as alterações feitas.



Controle de Versão e Colaboração

Alterando repositório remoto diretamente

git pull

Controle de Versão e Colaboração

Alterando o repositório remoto diretamente

.....

O nosso repositório local deve ser uma cópia do repositório remoto. Isso significa que devemos enviar nossas alterações para o repositório remoto e também baixar as alterações de lá para nossa máquina.

Então, para aprender como sincronizar e baixar as alterações quando o repositório remoto for atualizado, vamos:

- fazer uma alteração diretamente no repositório remoto; e
- baixar a alteração no repositório remoto: **git pull**.

Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



Primeira coisa: Verificar se os repositórios (local e remoto) estão iguais

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status ←
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ |
```

Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



FAZER UMA ALTERAÇÃO DIRETAMENTE NO REPOSITÓRIO REMOTO

Vamos criar um arquivo README no repositório remoto para simular uma alteração feita por outro programador.

Dica!

O README.md é um arquivo de texto utilizado para descrever, documentar ou exemplificar seu projeto. Apesar de não ser obrigatório (seu código vai funcionar perfeitamente sem ele), o README é essencial, pois mescla o cartão de visitas e a ementa do projeto, sendo o responsável por cativar interesse em seu trabalho ou relembrá-lo dos aspectos gerais e pontos relevantes daquele desenvolvimento.

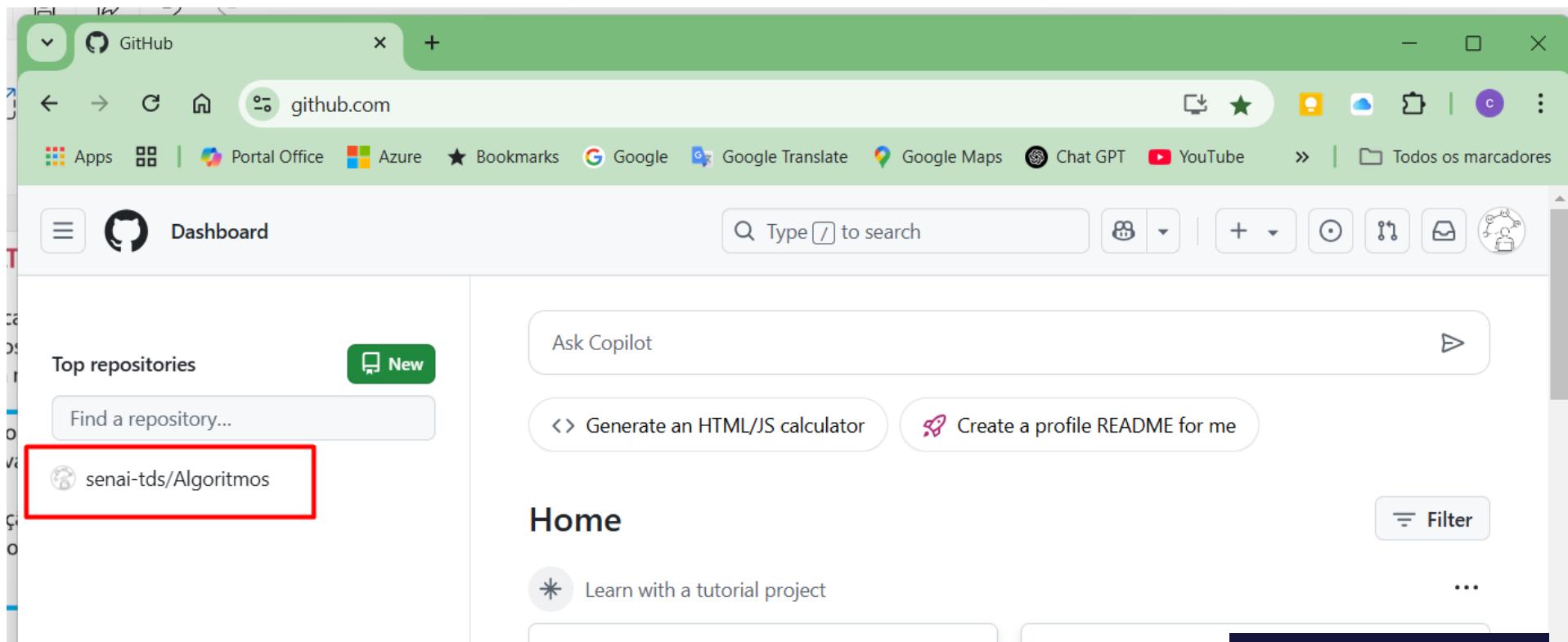


Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



Acessar o repositório remoto

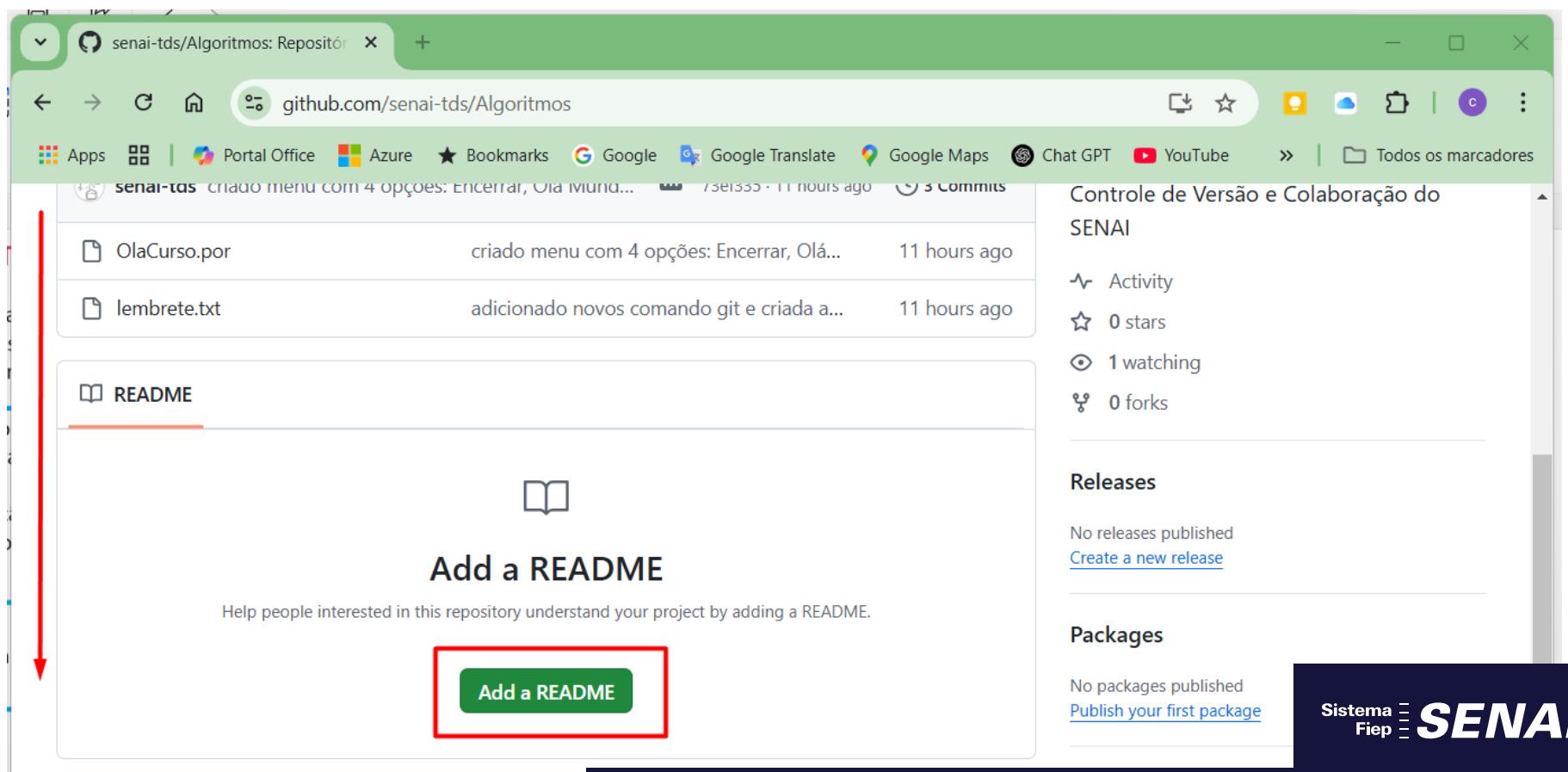


Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



Clicar no botão “Add a README”



Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



Adicionar um comentário

The screenshot shows a GitHub repository for 'senai-tds/Algoritmos'. The user is editing the 'README.md' file in the 'master' branch. The text area contains the following content:

```
1  ### CURSO DE APERFEIÇOAMENTO - GITHUB: CONTROLE DE VERSÃO E COLABORAÇÃO ###
2
3  Exemplo de arquivo README com informações sobre o curso.
4  Duração: 40h (10 dias)
5  Turma: APP-N-G00033 / 2025
6  Sala: Plataforma TEAMS
```

A red arrow points from the bottom-left of the text area to the 'Commit changes...' button in the top right corner of the editor. This button is highlighted with a red box.

Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



The screenshot shows a GitHub commit dialog box over a browser window. The commit message is "Criado arquivo README.md". The "Commit changes" button is highlighted with a red border and a red arrow points to it from the bottom right. The "Commit message" field is also highlighted with a red border.

Commit changes

Commit message

Criado arquivo README.md

Extended description

Add an optional extended description...

Commit directly to the `master` branch

Create a [new branch](#) for this commit and start a pull request [Learn more about pull requests](#)

Commit changes

Cancel

Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page.

Attach files by dragging & dropping, selecting or pasting them.

Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



The screenshot shows a web browser window with the GitHub interface. The URL in the address bar is `github.com/senai-tds/Algoritmos/tree/master`. The repository name is `senai-tds / Algoritmos`. The `Code` tab is selected. A commit history is displayed:

| Name | Last commit message | Last commit date |
|--------------|--|------------------|
| OlaCurso.por | criado menu com 4 opções: Encerrar, Olá Mundo Git, Comandos Linux e C... | 11 hours ago |
| README.md | Criado arquivo README.md. | now |
| lembrete.txt | adicionado novos comando git e criada area de comandos linux | 12 hours ago |
| README.md | | |

CURSO DE APERFEIÇOAMENTO - GITHUB: CONTROLE DE VERSÃO E COLABORAÇÃO

Exemplo de arquivo README com informações sobre o curso. Duração: 40h (10 dias) Turma: APP-N-G00033 / 2025 Sala: Plataforma T

Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



BAIXAR A ALTERAÇÃO NO REPOSITÓRIO REMOTO: *git pull*

Para baixar a atualização de seu repositório remoto para seu repositório local, execute no terminal local o comando **git pull**.

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git pull ←
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.16 KiB | 62.00 KiB/s, done.
From https://github.com/senai-tds/Algoritmos
  73ef335..dc4aa31 master      -> origin/master
Updating 73ef335..dc4aa31
Fast-forward
 README.md | 6 ++++++ ←
 1 file changed, 6 insertions(+)
 create mode 100644 README.md
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos
$ |
```

Controle de Versão e Colaboração

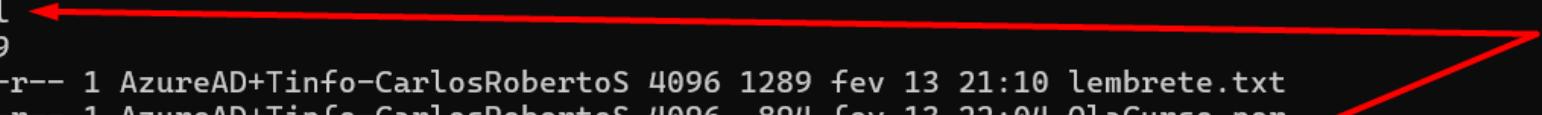
Alterando o repositório remoto diretamente



Verificando arquivos baixados

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ ls -l
total 9
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 1289 fev 13 21:10 lembrete.txt
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 894 fev 13 22:04 OlaCurso.pdf
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 221 fev 14 09:38 README.md

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```

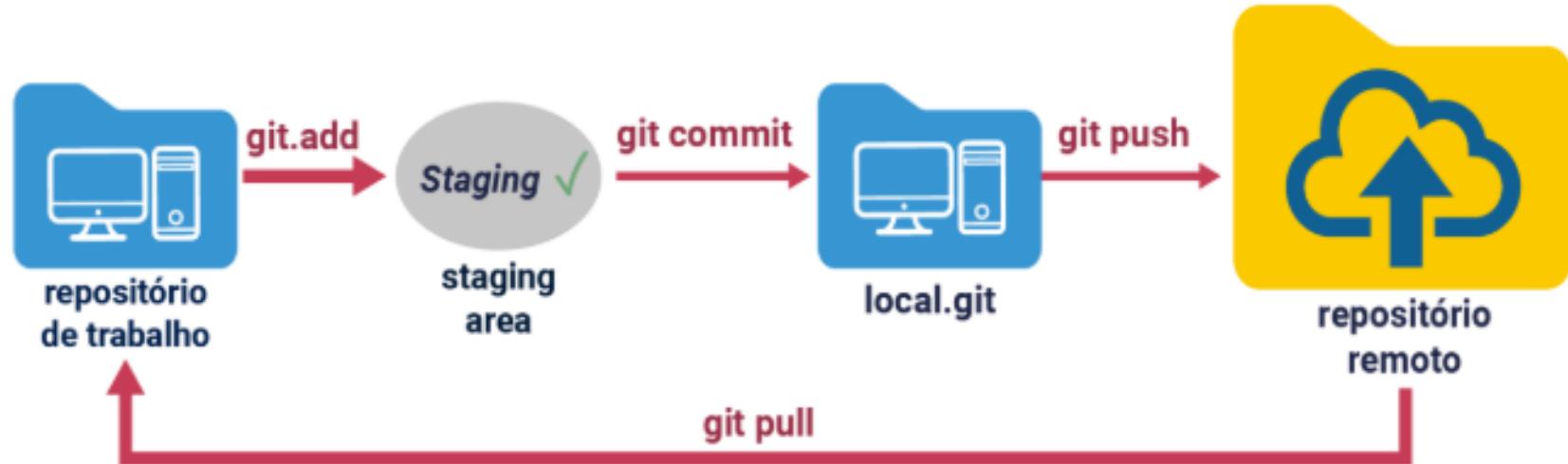


Controle de Versão e Colaboração

Alterando o repositório remoto diretamente



Alteração do fluxo de trabalho



Controle de Versão e Colaboração

Impedindo monitoramento de arquivos

.gitignore

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



IMPEDIR QUE UMA ALTERAÇÃO SEJA MONITORADA: ARQUIVO *.gitignore*

Imagine que você estava testando um trecho de código que não deu certo e, portanto, você não quer publicá-lo. Com o Git, é possível gerenciar arquivos que não desejamos publicar. Para isso, você deve criar um arquivo **.gitignore** dentro da pasta do projeto e informar quais arquivos você deseja ignorar. Podemos adicionar pastas, arquivos ou qualquer outro item que não desejamos publicar/ou adicionare em nosso repositório.

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Criar um arquivo na área “Working Directory”

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ touch arquivo-nao-monitorado.txt ←

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ echo "este arquivo não deu certo e, portanto, não queremos publicá-lo" > arquivo-nao-monitorado.txt ←

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    arquivo-nao-monitorado.txt ←

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

Observação: A partir da criação de um arquivo na área **Working Directory**, o GIT passa a monitorá-lo automaticamente

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Criar o arquivo “.gitignore”

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ touch .gitignore ←

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore ←
    arquivo-nao-monitorado.txt

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

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Inserir no arquivo “.gitignore” o arquivo a ser ignorado

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ echo "arquivo-nao-monitorado.txt" >> .gitignore
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ cat .gitignore
arquivo-nao-monitorado.txt
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

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

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

Observação: verificar através do comando **git status** que o **arquivo-nao-monitorado.txt** a partir deste momento não é mais reconhecido com modificado, inserido ou editado.

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Mostrar arquivos ocultos

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ ls -l -a
```

```
total 22
drwxr-xr-x 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 20:49 .
drwxr-xr-x 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 19:54 ../
drwxr-xr-x 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 20:51 .git/
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 27 fev 14 20:50 .gitignore
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 20:48 arquivo-nao-monitorado.txt
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 1289 fev 14 19:54 lembrete.txt
drwxr-xr-x 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 19:54 material/
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 894 fev 14 19:54 OlaCurso.por
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 210 fev 14 19:54 README.md
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ ls -l -A
```

```
total 18
drwxr-xr-x 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 20:51 .
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 27 fev 14 20:50 .gitignore
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 20:48 arquivo-nao-monitorado.txt
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 1289 fev 14 19:54 lembrete.txt
drwxr-xr-x 1 AzureAD+Tinfo-CarlosRobertoS 4096 0 fev 14 19:54 material/
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 894 fev 14 19:54 OlaCurso.por
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 210 fev 14 19:54 README.md
```

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Adicionar as alterações em “staging area”

git add .

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

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

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it ←

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore ←

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Confirmar as alterações para repositório local “.git/”

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git commit -m "Adicionado o arquivo .gitignore"
[master 6c49421] Adicionado o arquivo .gitignore
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ |
```

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Transferir alterações para repositório remoto

git push -u origin master

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/senai-tds/Algoritmos.git
  5acbd24..6c49421  master -> master
branch 'master' set up to track 'origin/master'.
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ |
```

Controle de Versão e Colaboração

Impedindo monitoramento de arquivos



Verificar alterações no repositório remoto
acessar github.com

The screenshot shows a GitHub repository page for 'senai-tds/Algoritmos'. The 'Code' tab is selected, displaying the 'master' branch with one commit. The commit details are as follows:

| File | Message | Date |
|--------------|--|------------|
| .gitignore | Adicionado o arquivo .gitignore | 3 days ago |
| OlaCurso.por | criado menu com 4 opções: Encerrar, Olá Mundo Git, Coman... | 4 days ago |
| README.md | Inserido o arquivo README.md com informações do curso | 3 days ago |
| lembrete.txt | adicionado novos comando git e criada area de comandos li... | 4 days ago |

The commit for '.gitignore' is highlighted with a red box.

Controle de Versão e Colaboração

Baixando repositório remoto

`git clone`

Controle de Versão e Colaboração

Baixando o repositório remoto



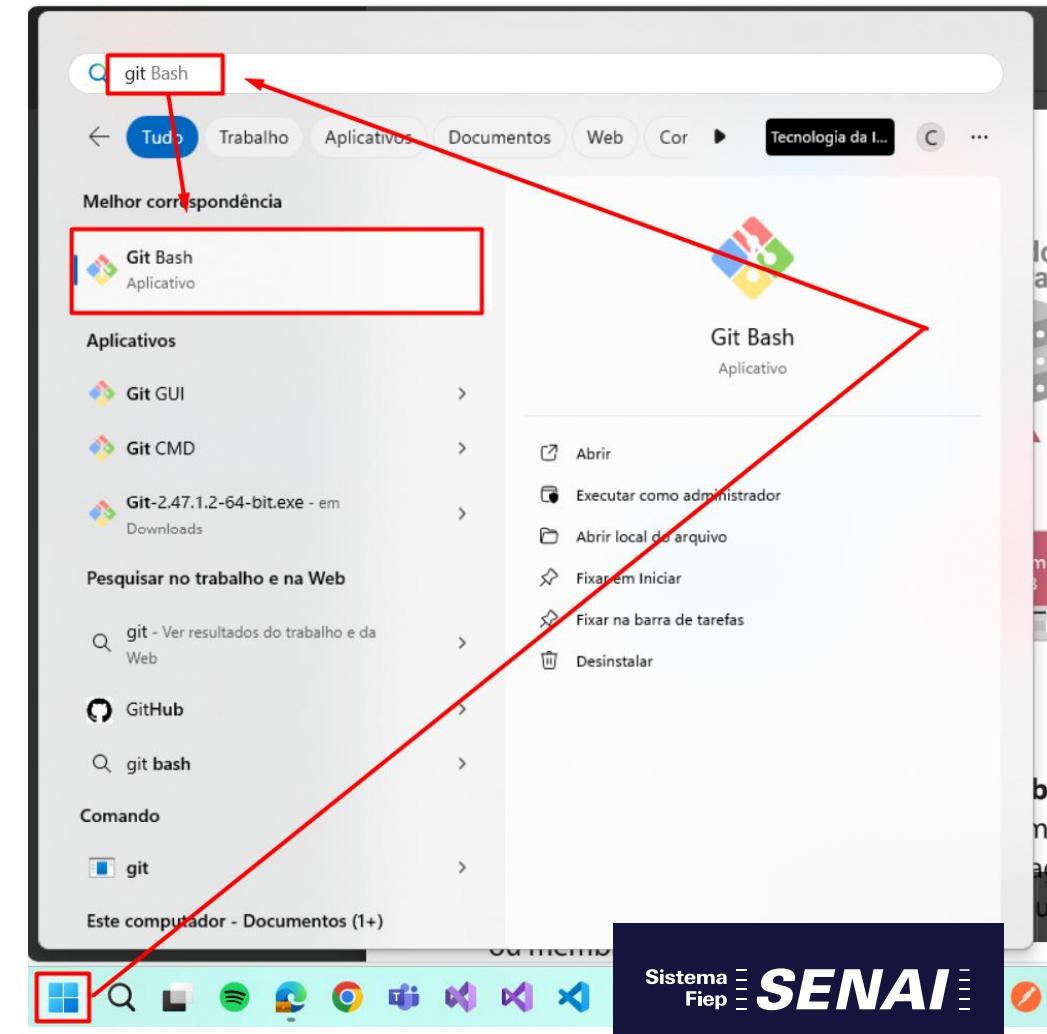
Criar estrutura de Pastas

Abrir o “Git Bash” na área do usuário

- Na barra de tarefas do Windows, clicar no botão iniciar;
- Na caixa de edição digitar “git”;
- Clicar no aplicativo “Git Bash” que deverá aparecer automaticamente conforme indicado na figura ao lado;



Git Bash
Aplicativo



Controle de Versão e Colaboração

Baixando o repositório remoto



Criar estrutura de pastas

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ pwd
/c/Users/Tinfo-CarlosRobertoS ← area do usuário

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~
$ cd Documents/ ← acessar pasta Documents

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents
$ mkdir senai-tds ← criar diretório senai-tds

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents
$ cd senai-tds/ ← acessar pasta senai-tds

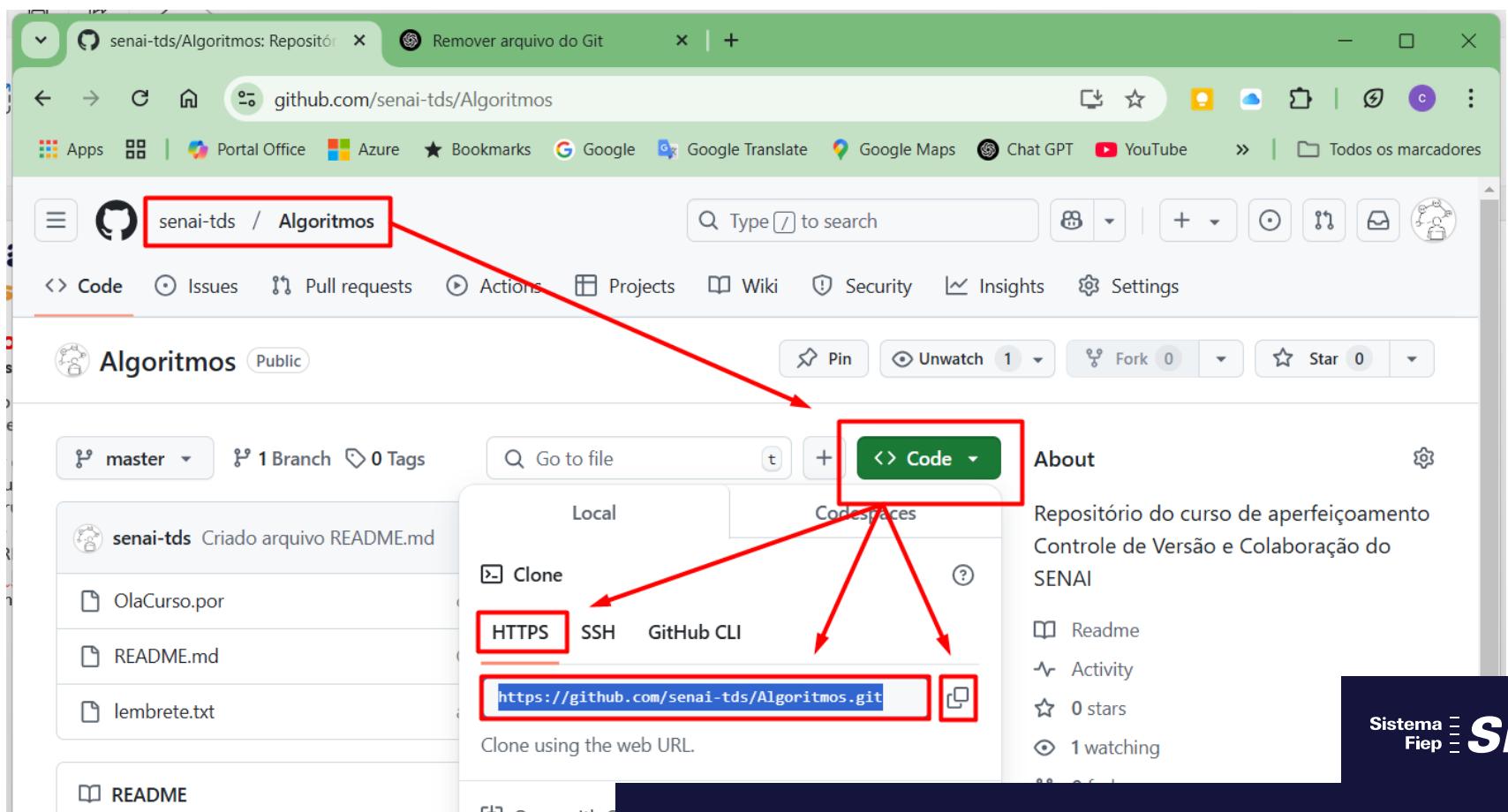
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds
$ pwd
/c/Users/Tinfo-CarlosRobertoS/Documents/senai-tds
```

Controle de Versão e Colaboração

Baixando o repositório remoto



Obter a URL Web do repositório remoto



Controle de Versão e Colaboração

Baixando o repositório remoto



Usar comando para clonagem do repositório remoto

```
git clone https://github.com/senai-tds/Algoritmos.git
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds
$ git clone https://github.com/senai-tds/Algoritmos.git
Cloning into 'Algoritmos'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 13 (delta 3), reused 7 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (3/3), done.
```

Controle de Versão e Colaboração

Baixando o repositório remoto



Verificando conteúdo do projeto recém clonado

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds
$ cd Algoritmos/

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ ls -l
total 9
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 1289 fev 14 11:45 lembrete.txt
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 894 fev 14 11:45 OlaCurso.por
-rw-r--r-- 1 AzureAD+Tinfo-CarlosRobertoS 4096 221 fev 14 11:45 README.md

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```

Controle de Versão e Colaboração

Criando atalhos

git config --global

Controle de Versão e Colaboração

Criando atalhos



O Git permite criar atalhos para comandos tornando o uso mais rápido e eficiente

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global alias.st status

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global --list
user.name=senai-tds
user.email=carlos.stunitz@outlook.com
gui.recentrepo=C:/Projetos/VS2017/TInfo/Seg2C
alias.npp=!"C:/Program Files/Notepad++/notepad++.exe" -multiInst -nosession
alias.st=status

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git st
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Controle de Versão e Colaboração

Criando atalhos



Conhecendo o git log no formato longo

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git log --oneline --all --decorate --graph
* 6c49421 (HEAD -> master, origin/master, origin/HEAD) Adicionado o arquivo .gitignore
* 5acbd24 Adicionado material utilizado em sala de aula
* 7104237 Inserido o arquivo README.md com informações do curso
* 1a26784 Removido arquivo README.md
* dc4aa31 Criado arquivo README.md
* 73ef335 criado menu com 4 opções: Encerrar, Olá Mundo Git, Comandos Linux e Comandos Git
* 9b88861 adicionado novos comando git e criada area de comandos linux
* 9abc6b6 Primeiro commit
```

Controle de Versão e Colaboração

Criando atalhos



Criando um atalho para o git log de forma resumida

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global alias.hist "log --oneline --all --decorate --graph" ←

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git hist
* 6c49421 (HEAD -> master, origin/master, origin/HEAD) Adicionado o arquivo .gitignore
* 5acbd24 Adicionado material utilizado em sala de aula
* 7104237 Inserido o arquivo README.md com informações do curso
* 1a26784 Removido arquivo README.md
* dc4aa31 Criado arquivo README.md
* 73ef335 criado menu com 4 opções: Encerrar, Olá Mundo Git, Comandos Linux e Comandos Git
* 9b88861 adicionado novos comando git e criada area de comandos linux
* 9abc6b6 Primeiro commit
```

Controle de Versão e Colaboração

Criando atalhos



Criando um atalho para o git log de forma resumida

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global alias.hist "log --oneline --all --decorate --graph" ←

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git hist
* 6c49421 (HEAD -> master, origin/master, origin/HEAD) Adicionado o arquivo .gitignore
* 5acbd24 Adicionado material utilizado em sala de aula
* 7104237 Inserido o arquivo README.md com informações do curso
* 1a26784 Removido arquivo README.md
* dc4aa31 Criado arquivo README.md
* 73ef335 criado menu com 4 opções: Encerrar, Olá Mundo Git, Comandos Linux e Comandos Git
* 9b88861 adicionado novos comando git e criada area de comandos linux
* 9abc6b6 Primeiro commit
```

Controle de Versão e Colaboração

Removendo configurações globais

git config --global --unset

Controle de Versão e Colaboração

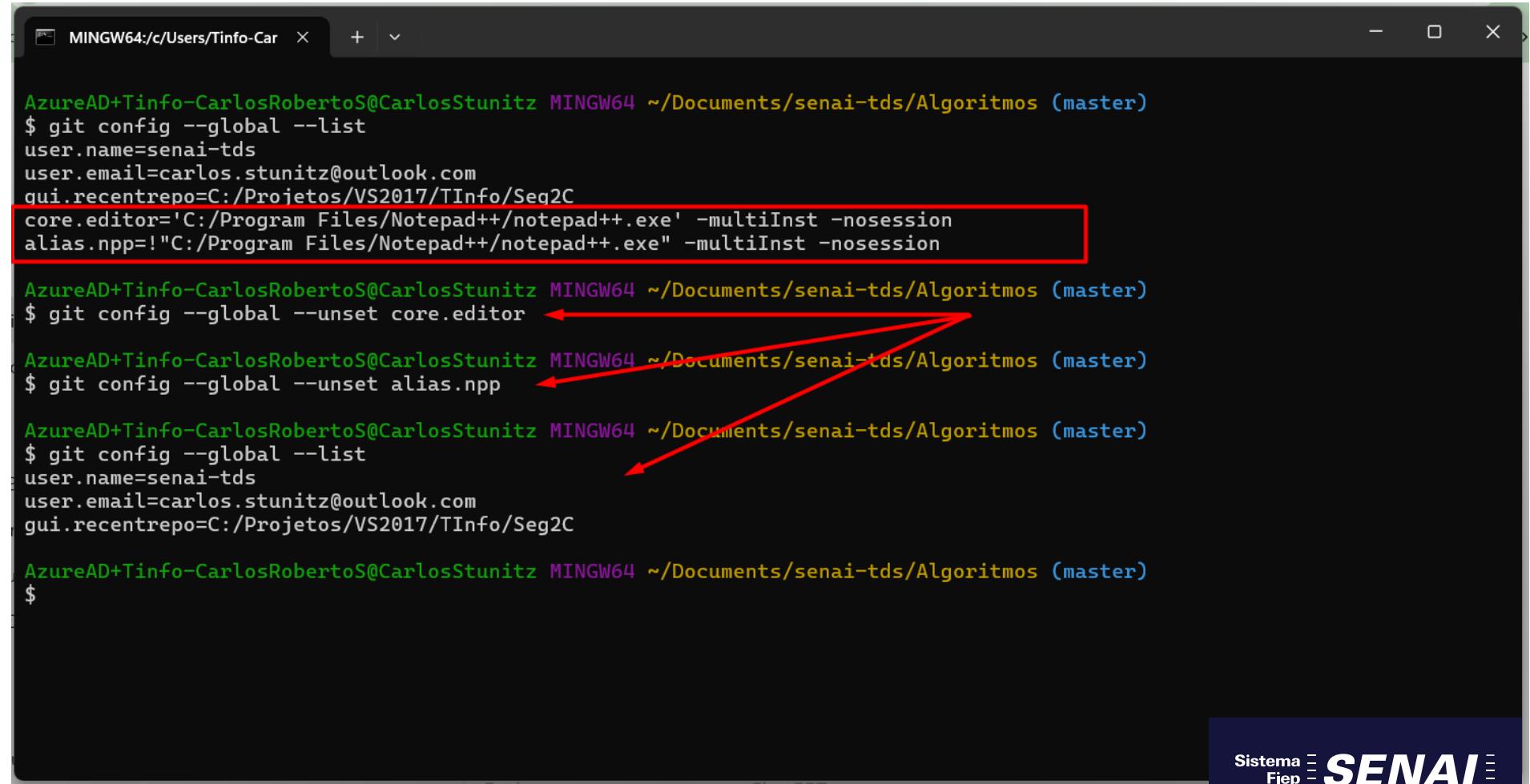
Removendo configurações globais



```
MINGW64:/c/Users/Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global --list
user.name=senai-tds
user.email=carlos.stunitz@outlook.com
gui.recentrepo=C:/Projetos/VS2017/TInfo/Seg2C
core.editor='C:/Program Files/Notepad++/notepad++.exe' -multiInst -nosession
alias.npp=!"C:/Program Files/Notepad++/notepad++.exe" -multiInst -nosession

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global --unset core.editor
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global --unset alias.npp
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global --list
user.name=senai-tds
user.email=carlos.stunitz@outlook.com
gui.recentrepo=C:/Projetos/VS2017/TInfo/Seg2C

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$
```



Controle de Versão e Colaboração

Configurando editor padrão com atalhos

git config

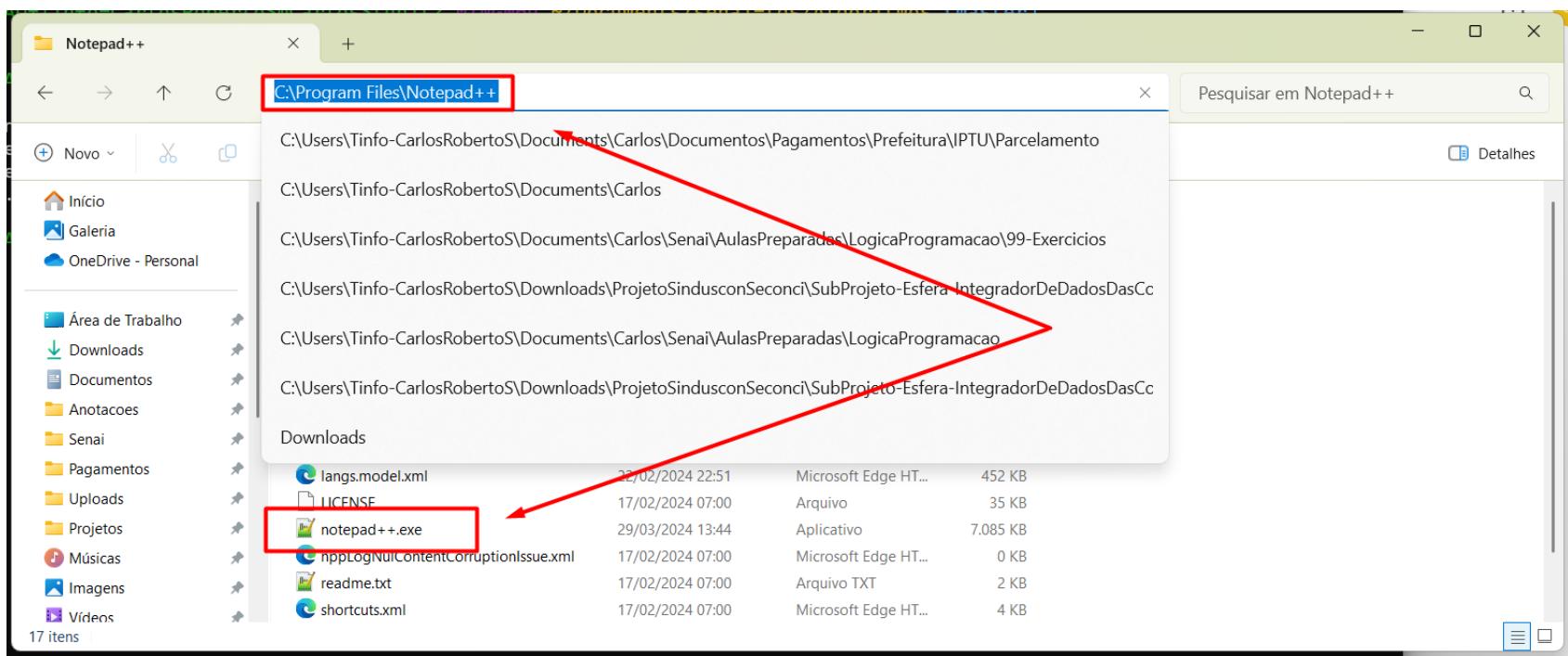


Controle de Versão e Colaboração

Configurando editor padrão com atalhos



Obtendo diretório e nome do arquivo

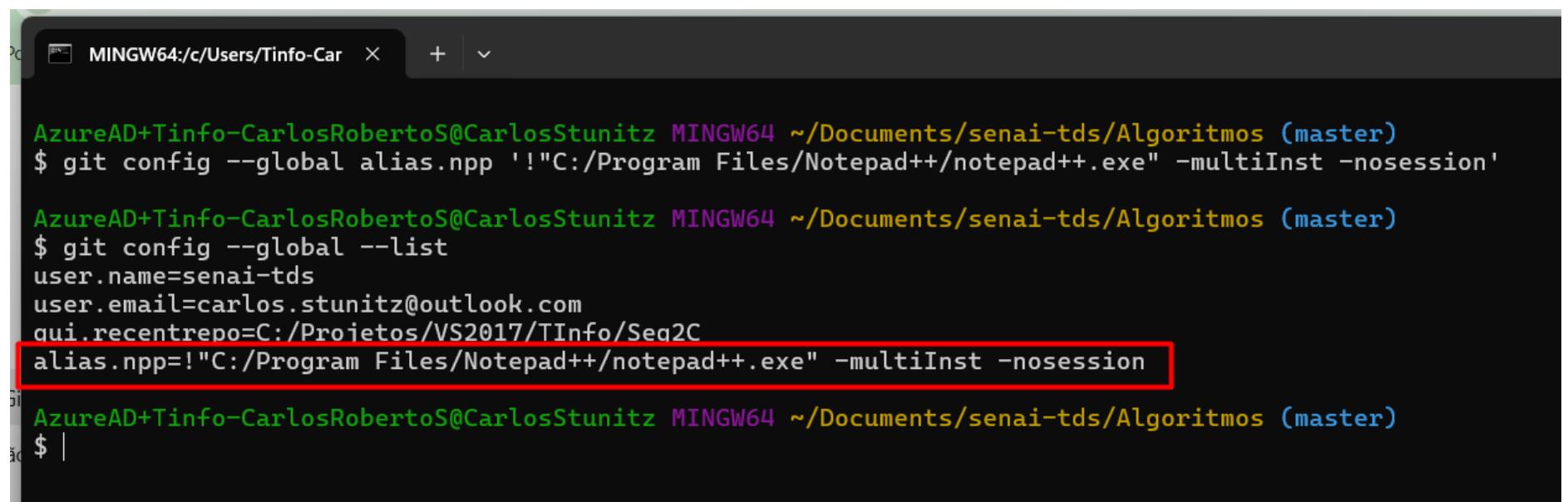


Controle de Versão e Colaboração

Configurando editor padrão com atalhos



Configurando editor padão



```
MINGW64:/c/Users/Tinfo-Car  X + ▾
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global alias.npp '!"C:/Program Files/Notepad++/notepad++.exe" -multiInst -nosession'

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ git config --global --list
user.name=senai-tds
user.email=carlos.stunitz@outlook.com
qui.recentrepo=C:/Projetos/VS2017/TInfo/Seq2C
alias.npp=!"C:/Program Files/Notepad++/notepad++.exe" -multiInst -nosession

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
$ |
```

Controle de Versão e Colaboração

Configurando editor padrão com atalhos



Observar que enquanto o aplicativo configurado estiver sendo executado o prompt do terminal não fica disponível para uso

Usando editor padrão configurado

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmos (master)
\$ git npp lembrete.txt

C:\Users\Tinfo-CarlosRobertoS\Documents\senai-tds\Algoritmos\lembrete.txt - Notepad++

```
1 Anotações:  
2  
3 ### COMANDOS LINUX ###  
4 ==> touch [nomeDoArquivo.ext]: cria arquivo em branco  
5 ==> echo "conteúdo" > [nomeDoArquivo.ext]: escreve no arquivo apagando todo o  
6 ==> echo "conteúdo" >> [nomeDoArquivo.ext]: estende (append) no arquivo prese:  
7 ==> cat [nomeDoArquivo.ext] exibe arquivo na tela  
8 ==> ls -d .*/ exibe arquivos e diretórios ocultos no git bash  
9 ==> mkdir [nomeDiretorio]: criar um novo diretório  
10 ==> cd [nomeDiretorio]/: navegar entre diretórios  
11 ==> rm -rf [nomeDiretorio]: apaga diretórios sem confirmação
```

Normal text file length : 1.289 lines : 25 Ln : 1 Col : 1 Pos : 1 Windows (CR L)

Controle de Versão e Colaboração

Visualizando e comparando alterações

git difftool



Controle de Versão e Colaboração

Visualizando e comparando alterações realizadas

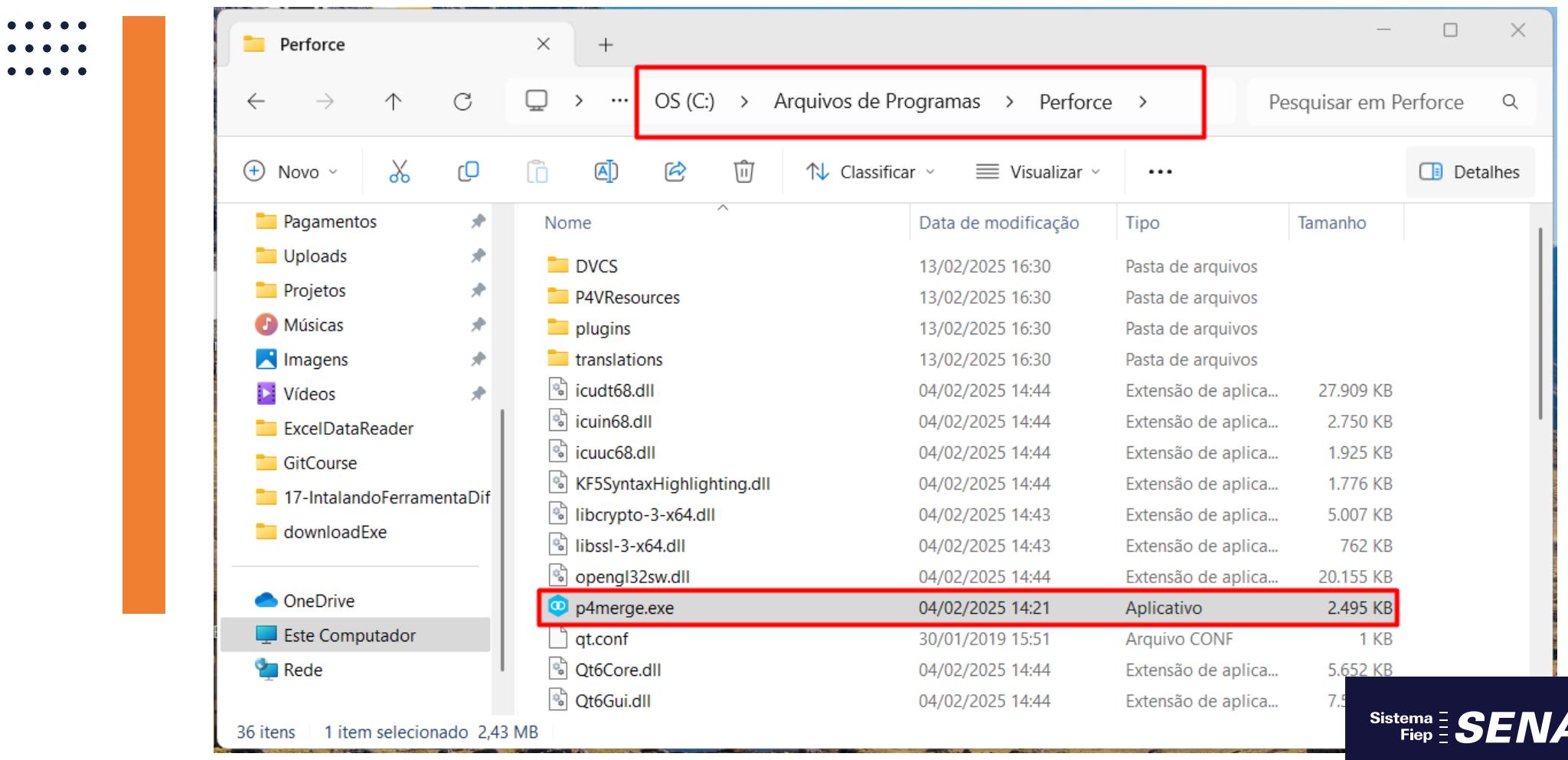


Instalar aplicativo p4Merge.exe da Perforce

Verificar documento específico para
download e instalação do aplicativo
p4merge.exe da Perforce

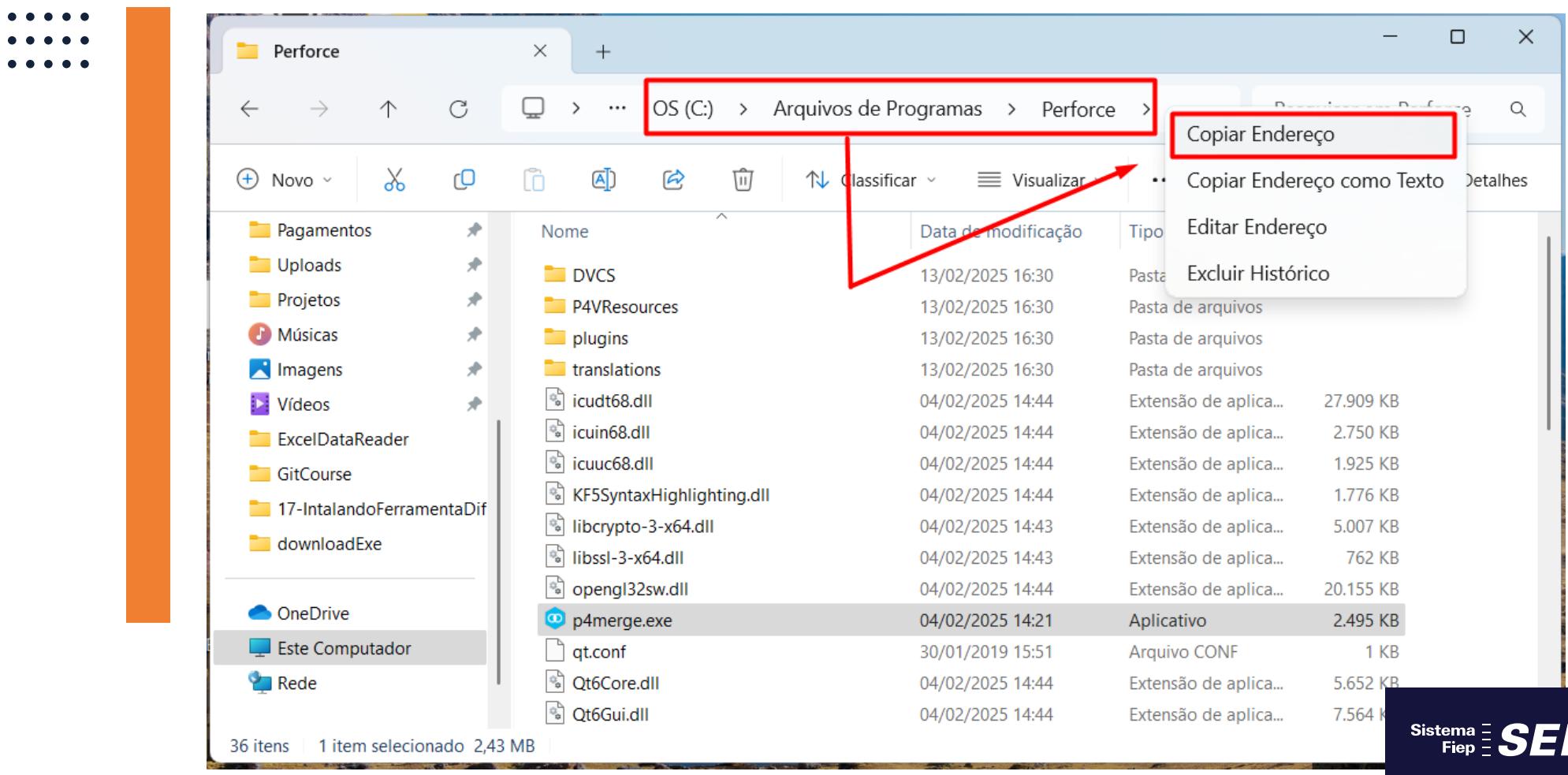
Controle de Versão e Colaboração

Visualizando e comparando alterações realizadas



Controle de Versão e Colaboração

Visualizando e comparando alterações realizadas



Controle de Versão e Colaboração

Visualizando e comparando alterações realizadas



The terminal window shows the following configuration steps:

```
MINGW64:/c/Users/Tinfo-Car... X + v configurando P4Merge para ferramenta diff tool
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global diff.tool p4merge
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global difftool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global difftool.prompt false
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global --list
user.name=senai-tds
user.email=carlos.stunitz@outlook.com
gui.recentrepo=C:/Projetos/VS2017/TInfo/Seg2C
alias.hist=log --oneline --all --decorate --graph
diff.tool=p4merge
difftool.p4merge.path=C:/Program Files/Perforce/p4merge.exe
difftool.prompt=false
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$
```

Controle de Versão e Colaboração

Visualizando e comparando alterações realizadas



The screenshot shows a terminal window titled "MINGW64:c/Users/Tinfo-Car" with the command line "configurando P4Merge para ferramenta merge tool". The window displays the following git configuration commands:

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global merge.tool p4merge
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global mergetool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global mergetool.prompt false
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git config --global --list
user.name=senai-tds
user.email=carlos.stunitz@outlook.com
gui.recentrepo=C:/Projetos/VS2017/TInfo/Seg2C
alias.hist=log --oneline --all --decorate --graph
diff.tool=p4merge
difftool.p4merge.path=C:/Program Files/Perforce/p4merge.exe
difftool.prompt=false
merge.tool=p4merge
mergetool.p4merge.path=C:/Program Files/Perforce/p4merge.exe
mergetool.prompt=false
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ |
```

Red boxes highlight the first four configuration lines, and another red box highlights the last three configuration lines.

Controle de Versão e Colaboração

Visualizando e comparando alterações realizadas



```
MINGW64:/c/Users/Tinfo-Car X + - X
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git status
On branch master
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:   lembrete.txt

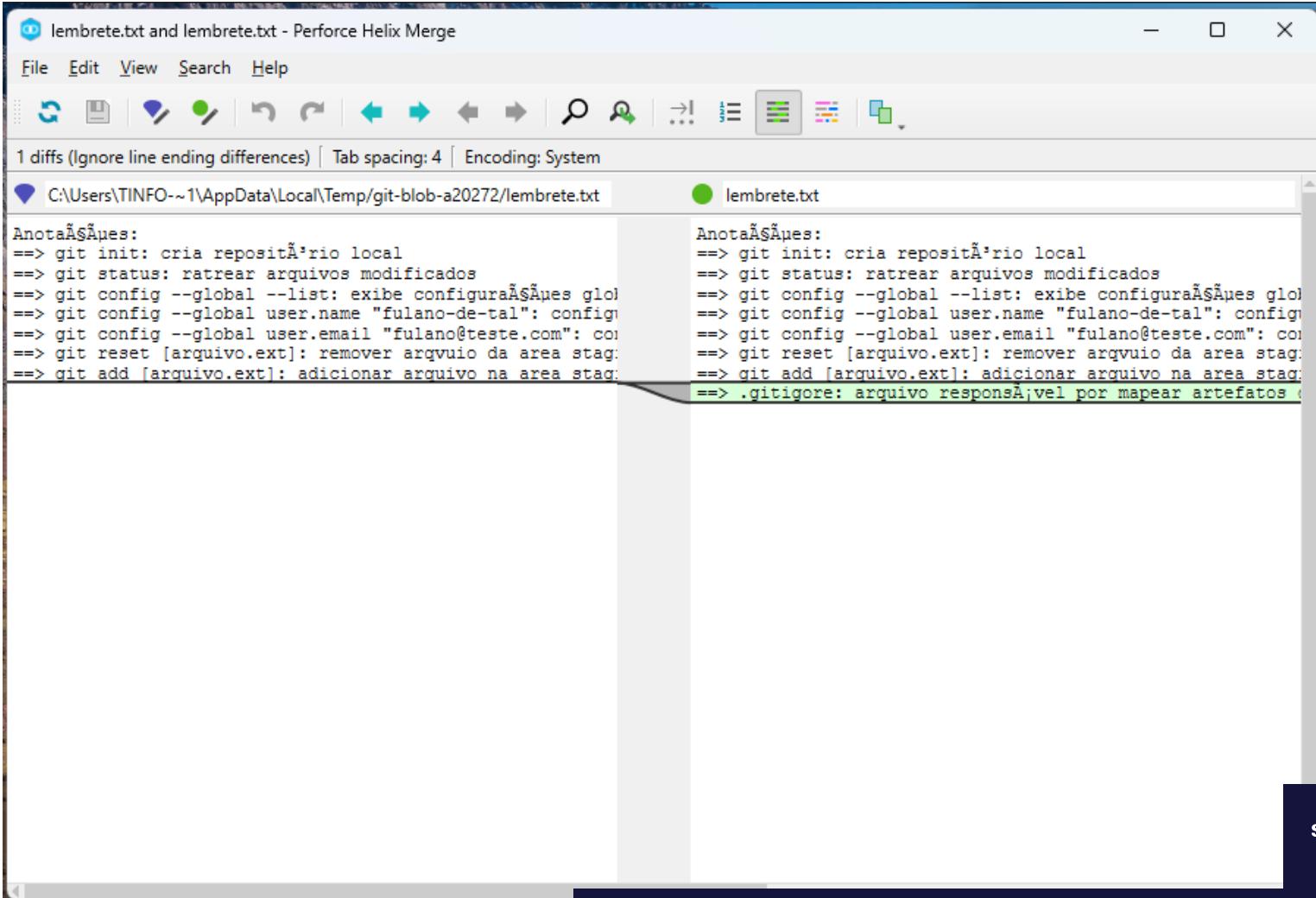
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

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

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Algoritmo (master)
$ git difftool lembrete.txt|
```

Controle de Versão e Colaboração

Visualizando e comparando alterações realizadas



The screenshot shows a file comparison interface titled "lembrete.txt and lembrete.txt - Perforce Helix Merge". The left pane displays the local file content:

```
Anotações:  
==> git init: cria repositório local  
==> git status: listar arquivos modificados  
==> git config --global --list: exibe configurações globais  
==> git config --global user.name "fulano-de-tal": configuração de nome do usuário global  
==> git config --global user.email "fulano@teste.com": configuração de e-mail global  
==> git reset [arquivo.ext]: remover arquivo da área stage  
==> git add [arquivo.ext]: adicionar arquivo na área stage
```

The right pane displays the remote file content:

```
Anotações:  
==> git init: cria repositório local  
==> git status: listar arquivos modificados  
==> git config --global --list: exibe configurações globais  
==> git config --global user.name "fulano-de-tal": configuração de nome do usuário global  
==> git config --global user.email "fulano@teste.com": configuração de e-mail global  
==> git reset [arquivo.ext]: remover arquivo da área stage  
==> git add [arquivo.ext]: adicionar arquivo na área stage  
==> .gitignore: arquivo responsável por mapear artefatos
```

The interface includes a toolbar with various file operations and a status bar indicating "1 diff" and "Tab spacing: 4".

Controle de Versão e Colaboração

Branches

Controle de Versão e Colaboração

Branches



Por que usar branches?

- Permite trabalhar em novas funcionalidades sem afetar o código estável;
- Facilita a colaboração em equipe;
- Ajuda a organizar o fluxo de trabalho, permitindo múltiplos desenvolvedores trabalharem simultaneamente;
- Garante que apenas o código testado e aprovado seja mesclado à Branch principal;

Controle de Versão e Colaboração

Branches



Trabalho de Equipe

Um dos pontos importantes do Git é a sua flexibilidade de trabalho, seja para um trabalho individual, em equipes pequenas ou grandes. Por meio do versionamento e rastreamento, é possível trabalharmos de maneira paralela em tarefas distintas e entregarmos itens de modo mais rápido durante o processo. Para trabalhar várias tarefas e trechos de códigos simultaneamente usamos o recurso **branch**.



Controle de Versão e Colaboração

Branches

.....

Até agora, todas as alterações estavam sendo feitas no mesmo ramo de trabalho (**master**). Agora conheceremos mais a fundo o que acontece quando vários programadores editam o mesmo arquivo.

Considerando um determinado ponto do código fonte, nós podemos criar uma cópia deste trabalho e atuarmos em tarefa de maneira paralela.



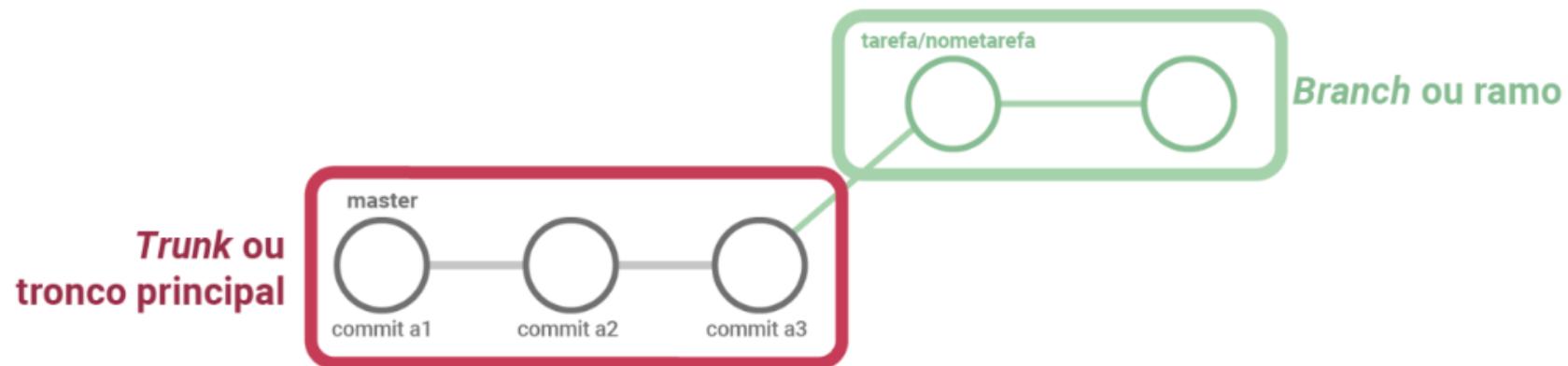
Controle de Versão e Colaboração

Branches

.....



Na prática, para cada atividade que formos executar, atuaremos em um novo ramo (**branch**) distinto derivado da branch principal, posteriormente uniremos o trabalho dos ramos com o tronco principal.



Controle de Versão e Colaboração

Branches



Como as branches são utilizadas?

- No Git, a branch principal costuma se chamar *main* ou *master*;
- Outras branches são criadas para:
 - Adicionar funcionalidades
 - Corrigir bugs,
 - Testar novas ideias antes de serem integradas ao código principal;

Controle de Versão e Colaboração

Branches



Listar branches

Listar todas as branches locais

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git branch ←
* master
```

Listar todas as branches remotas

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git branch -r ←
  origin/HEAD -> origin/master
  origin/master
```

Listar todas as branches

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
```

Controle de Versão e Colaboração

Branches



Criar uma nova branch

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git branch nova-branch
```

Verificar ou listar todas as branches locais

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git branch
* master
  nova-branch
```

Controle de Versão e Colaboração

Branches



Mudar ou acessar uma branch específica

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git checkout nova-branch
Switched to branch 'nova-branch'

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (nova-branch)
$
```

```
graph TD; master["(master)"] --> novaBranch["(nova-branch)"]
```

OU

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git switch nova-branch
Switched to branch 'nova-branch'

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (nova-branch)
$
```

```
graph TD; master["(master)"] --> novaBranch["(nova-branch)"]
```

Controle de Versão e Colaboração

Branches



Criar uma nova branch e acessá-la simultaneamente

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git checkout -b outra-branch ←
Switched to a new branch 'outra-branch'

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (outra-branch)
$
```

OU

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git switch -c mais-uma-branch ←
Switched to a new branch 'mais-uma-branch'

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$
```

Controle de Versão e Colaboração

Branches



Alterar um artefato na nova branch

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ echo -e "### Aprendi o comando: git branch ###\n" >> README.md
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ cat README.md
### CURSO DE APERFEIÇOAMENTO – GITHUB: CONTROLE DE VERSÃO E COLABORAÇÃO ###

Exemplo de arquivo README com informações sobre o curso.
Duração: 40h (10 dias)
Turma: APP-N-G00033 / 2025
Sala: Plataforma TEAMS

### Lembrei do comando: git status ###
### Lembrei do comando: git commit ###
### Aprendi o comando: git branch ###
```

Controle de Versão e Colaboração

Branches



Rastreando alterações realizadas na branch

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ git status
On branch mais-uma-branch
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:   README.md

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

Controle de Versão e Colaboração

Branches



Visualizar alterações efetuadas

The screenshot shows a diff tool interface with two panes. The left pane shows the initial state of the file, and the right pane shows the updated state after a merge. The changes are highlighted in green.

Left Pane (Initial State):

```
### CURSO DE APERFEIÇOAMENTO - GITHUB ###
Nome do Curso: CONTROLE DE VERSÃO E COLABORAÇÃO
Duração.....: 40h ou 10 dias
Turma.....: APP-N-G00033 / 2025
Sala.....: Plataforma Microsoft Teams

### Aprendi os comandos: git branch, git checkout e git s...
```

Right Pane (Updated State):

```
### CURSO DE APERFEIÇOAMENTO - GITHUB ###
Nome do Curso: CONTROLE DE VERSÃO E COLABORAÇÃO
Duração.....: 40h ou 10 dias
Turma.....: APP-N-G00033 / 2025
Sala.....: Plataforma Microsoft Teams

### Aprendi os comandos: git branch, git checkout e git s...
### Aprendi mais um comando: git merge ###
```

Controle de Versão e Colaboração

Branches



Adicionar alterações à staging area

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
```

consultando “status” novamente

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ git status
On branch mais-uma-branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md
```

Controle de Versão e Colaboração

Branches



Confirmando alterações na branch mais-uma-branch

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ git commit -m "adicionado texto: aprendi o git branch"
[mais-uma-branch 2e79a52] adicionado texto: aprendi o git branch
 1 file changed, 2 insertions(+)
```

consultando “status” novamente

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ git status
On branch mais-uma-branch
nothing to commit, working tree clean
```

Controle de Versão e Colaboração

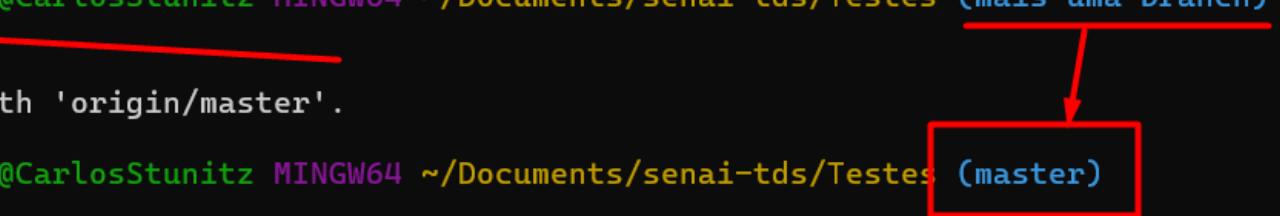
Branches



Acessando a branch master (principal)

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (mais-uma-branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$
```



consultando “status” na Branch principal

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Controle de Versão e Colaboração

Branches



Consultar artefato alterado

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ cat README.md
### CURSO DE APERFEIÇOAMENTO - GITHUB: CONTROLE DE VERSÃO E COLABORAÇÃO ###
```

Exemplo de arquivo README com informações sobre o curso.
Duração: 40h (10 dias)
Turma: APP-N-G00033 / 2025
Sala: Plataforma TEAMS

```
### Lembrei do comando: git status ###
### Lembrei do comando: git commit ###
```

O arquivo encontra-se sem a alteração efetuada
na branch que estávamos trabalhando

Controle de Versão e Colaboração

Branches



Mesclar alterações da branch **mais-uma-branch** com a
branch **master**

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git merge mais-uma-branch -m "alterações no arquivo README.md"
Updating 836bc98..2e79a52
Fast-forward (no commit created; -m option ignored)
 README.md | 2 ++
 1 file changed, 2 insertions(+)

AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ |
```

Controle de Versão e Colaboração

Branches



Consultando status

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Observar a indicação que o comando “**git push**” deverá ser utilizado para publicação das alterações realizada localmente no repositório remoto (**origin/master**)

Controle de Versão e Colaboração

Branches



Publicando alterações no repositório remoto

```
AzureAD+Tinfo-CarlosRobertoS@CarlosStunitz MINGW64 ~/Documents/senai-tds/Testes (master)
$ git push origin master
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 407 bytes | 203.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/senai-tds/Testes.git
  836bc98..2e79a52  master -> master
```

Controle de Versão e Colaboração

Branches

Verificando alterações no repositório remoto

The screenshot shows a GitHub repository page for 'senai-tds/Testes'. The repository has 1 branch and 0 tags. The commit history shows three commits:

- senai-tds adicionado texto: aprendi o git branch (2e79a52 · 29 minutes ago)
- OlaCurso.por Primeiro Commit (4 days ago)
- README.md adicionado texto: aprendi o git branch (29 minutes ago)

The 'README' file content is as follows:

```
CURSO DE APERFEIÇOAMENTO - GITHUB: CONTROLE DE VERSÃO E COLABORAÇÃO
Exemplo de arquivo README com informações sobre o curso. Duração: 40h (10 dias) Turma: APP-N-G00033 / 2025
Sala: Plataforma TEAMS

Lembrei do comando: git status
Lembrei do comando: git commit
Aprendi o comando: git branch
```

About

- Testes
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Languages

Portuguese 100.0%

OBRIG@DO

Carlos Roberto Stunitz

SISTEMA FIEP