



WBA0882\_v1.0

# Design Patterns e gerência de configuração: do projeto ao controle de versões



# Controle de versões na prática com ferramentas de versionamento

Ferramentas de controle.

## Bloco 1

Arthur Gonçalves Ferreira



## ➤ Introdução

- **O que são as ferramentas de controle de versão? Visão geral.**

Os sistemas de **controle de versão**:

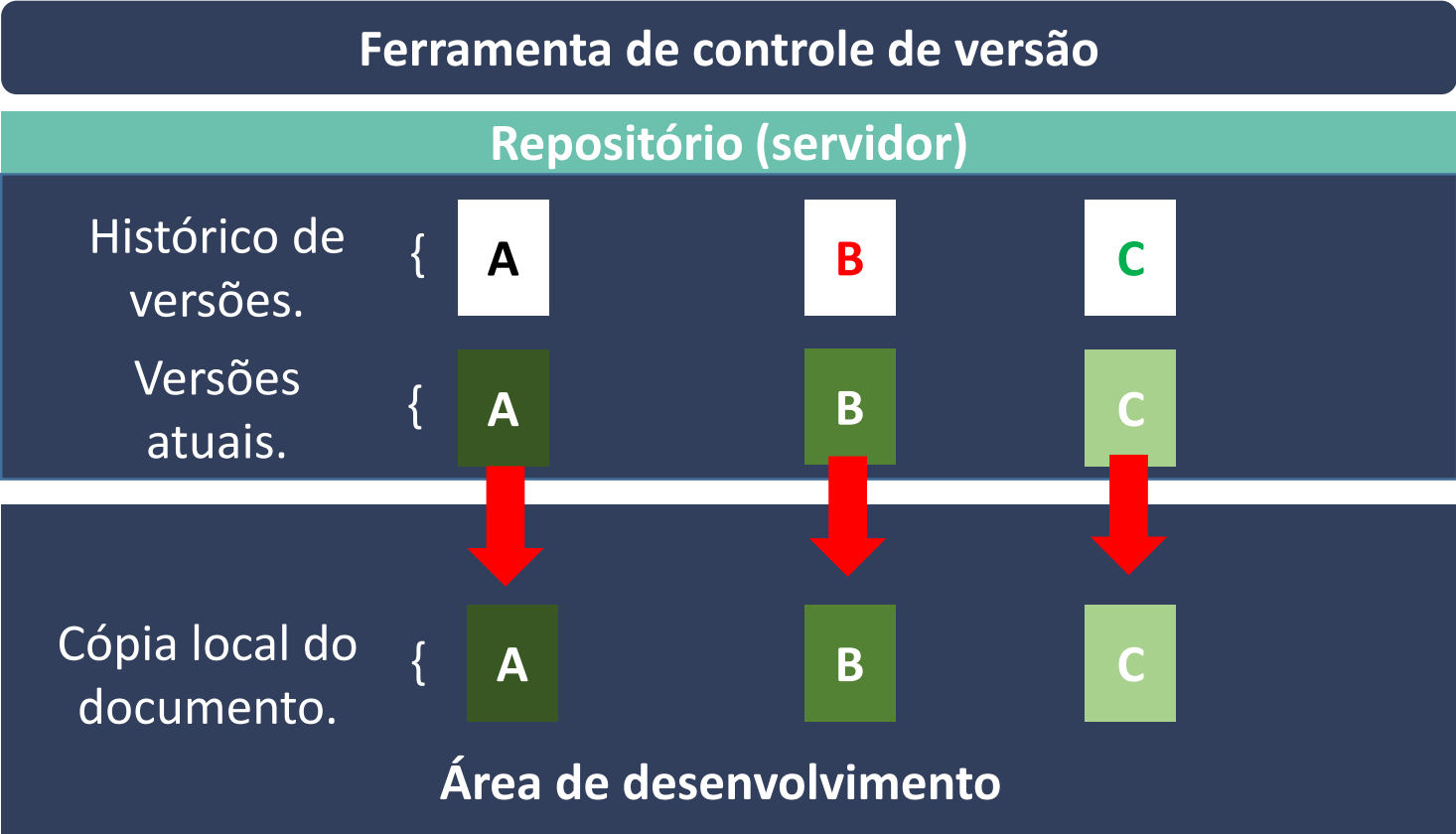
- Auxiliam equipes de desenvolvedores a gerenciar alterações no código fonte.
- Mantêm registro de todas as modificações no código.
- Permite organização.
- Oferecem segurança a um projeto.
- Permitem visualizar as evoluções do sistema.
- Eliminam ou diminui quase a 0 a probabilidade de erros grotescos de controle de sistemas.





# O que são ferramentas de controle? Visão geral

Figura 1 – Ferramentas de controle de versão



Fonte: elaborada pelo autor.

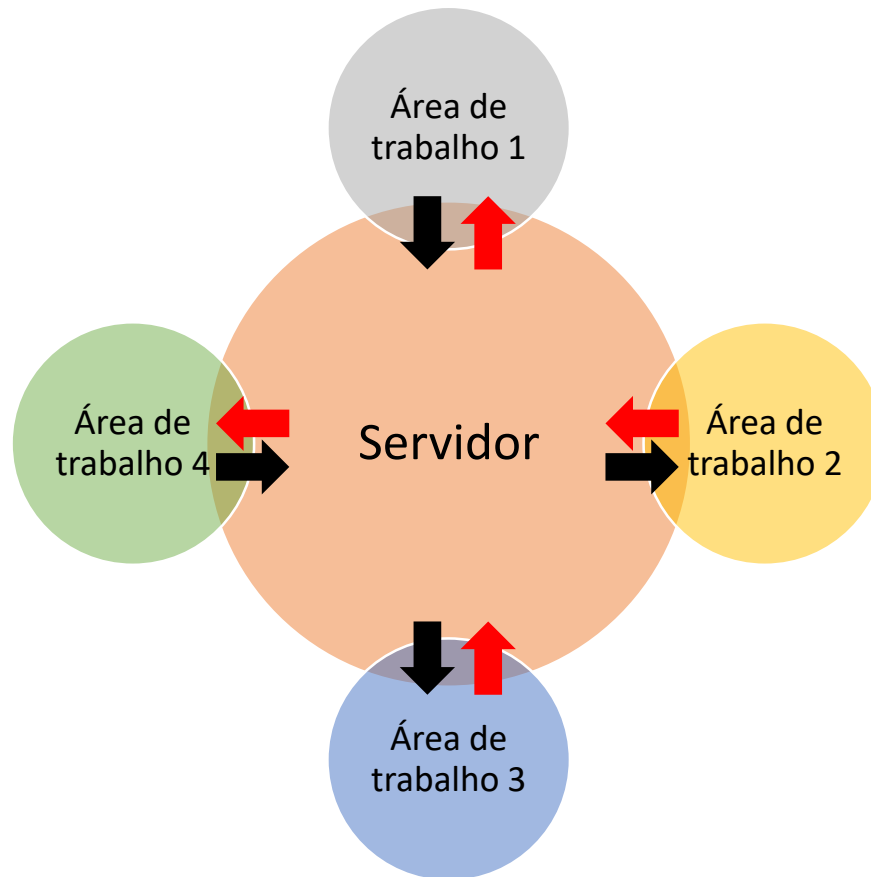
## ► Tipos de ferramentas de controle: centralizada

- Utiliza um servidor central (repositório local).
- Ideal para equipes pequenas que trabalhem em rede local.
- Utilizam arquitetura cliente-servidor.
- Possuem várias áreas de trabalho.
- As áreas de trabalho passam pelo servidor para se comunicar.



## ➤ Tipos de ferramentas de controle: centralizada

Figura 2 – Ferramenta centralizada de controle de versão



Fonte: elaborada pelo autor.

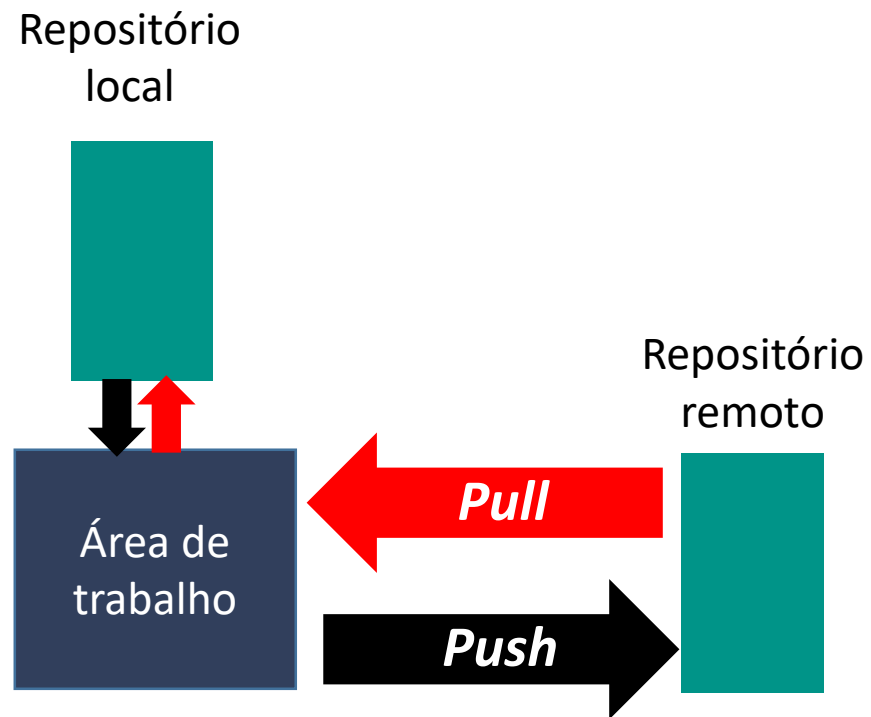
## ➤ Tipos de ferramentas de controle: distribuído

- Ideal para equipes grandes que trabalhem em diferentes locais.
- Cada área de trabalho possui seu próprio servidor.
- Áreas de trabalho comunicam-se entre si, sem o servidor.
- Utiliza um servidor como centro de envio de arquivos.
- Utiliza servidor local e servidor remoto.
- Utilizam as operações *pull* (Puxar) e *push* (Empurrar)
- É mais rápido que uma ferramenta de controle centralizada.
- São ferramentas mais complexas.



## ➤ Tipos de ferramentas de controle: distribuído

Figura 3 – Ferramenta distribuído de controle de versão



Fonte: elaborada pelo autor.





# Exemplos de ferramentas de controle centralizado e distribuído

Quadro 1 – Ferramenta de controle de versão

	<i>Open source</i>	Comercial
Centralizada.	Subversion e CVS.	Team Foundation Server - Microsoft.
Distribuído.	Mercurial   Git.	Team Concert - IBM/Rational.

Fonte: elaborado pelo autor.



# Gerência de configuração: atividades, ferramentas e desafios

Ferramenta de controle Git

## Bloco 2

Arthur Gonçalves Ferreira



# ➤ Introdução à ferramenta de controle de versão Git

- **O que é o Git:**
  - Ferramenta de controle distribuído.
  - Área de desenvolvimento.
  - Criado por Linus Torvalds em 2005.
- **Objetivo:**
  - Registrar as mudanças ocorridas no código fonte do projeto.
  - Permitir que os arquivos sejam alterados de forma simultânea.
  - Restaurar arquivos para de uma versão anterior.



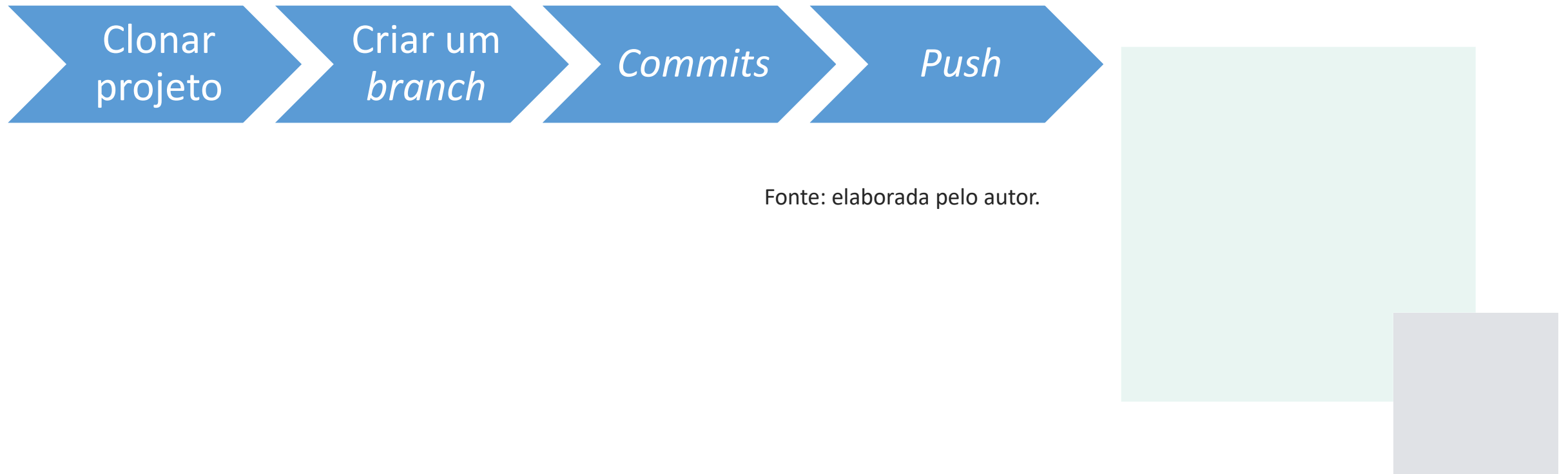
# ➤ Introdução à ferramenta de controle de versão Git

- **Ferramentas instaladas com o Git:**
  - Utiliza arquitetura em *branch* (ramificações).
  - Os arquivos ficam em um repositório remoto.
  - Os arquivos do repositório remoto são clonados para um repositório local.



# ➤ Estrutura de funcionamento da ferramenta de controle de versão Git

Figura 4 – Estrutura de funcionamento da ferramenta de controle de versão Git



Fonte: elaborada pelo autor.

# ➤ Introdução à ferramenta de controle de versão Git

- **Ferramentas instaladas com o Git:**
  - Git Bash.
  - Git cmd.
  - Git GUI.





# Exemplo de controle de versão com Git

Figura 5 – Git bahs, primeiros comandos

```
MINGW64:/C/Users/usuario

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site
$ git config --global user.name "user"

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site
$ git config --global user.email "user@email.com"

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site
$ git init
Initialized empty Git repository in C:/Users/user/Desktop/Site/.git/

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)
$ git status
On branch master

No commits yet

Untracked files:
(use "git add <file>..." to include in what will be committed)
  Aula Word 1.pdf
  "Exerc\303\255cio Word.pdf"

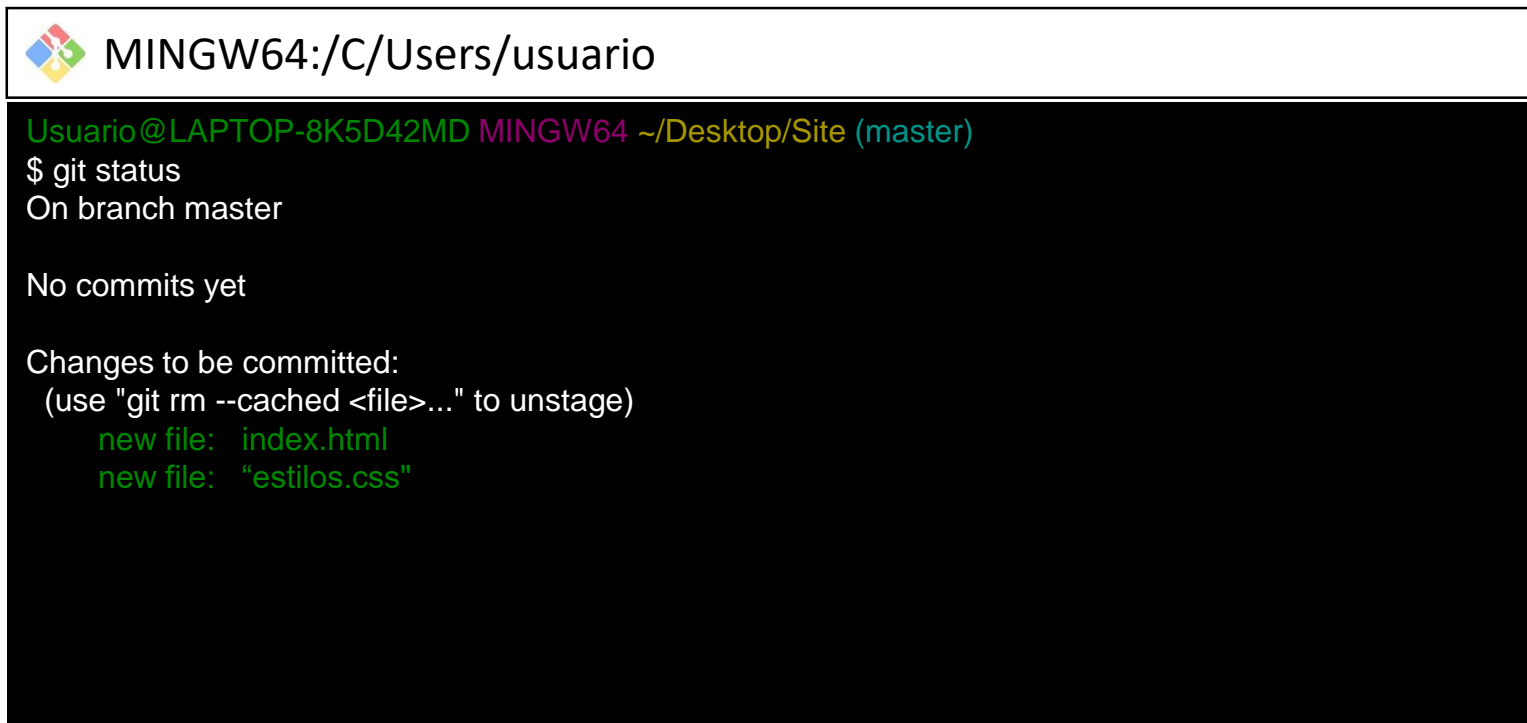
nothing added to commit but untracked files present (use "git add" to track)

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site master)
$ git add .
```

Fonte: captura de tela de Git.

# Exemplo de controle de versão com Git

Figura 6 – Git Bash, verificando arquivos adicionados



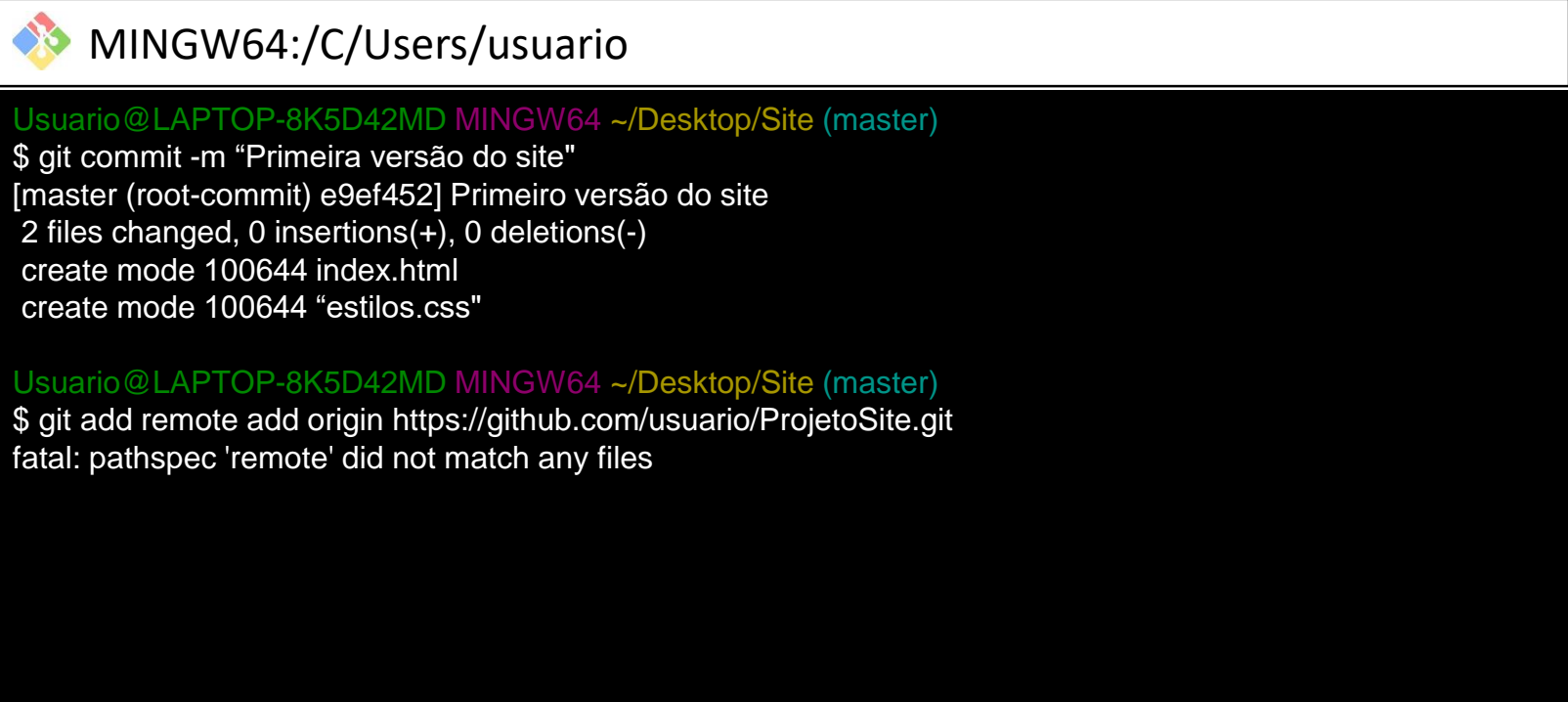
```
MINGW64:/C/Users/usuario  
  
Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)  
$ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
    new file:   index.html  
    new file:   "estilos.css"
```

Fonte: captura de tela de Git.



# Exemplo de controle de versão com Git

Figura 7 – Git Bash, conectando com o Github

A screenshot of a Git Bash terminal window. The title bar at the top shows the Windows logo icon and the path 'MINGW64:/C/Users/usuario'. The terminal background is black with text in various colors. The first prompt shows the user is in the directory ~/Desktop/Site on the master branch. They run 'git commit -m "Primeira versão do site"', which results in a successful commit with hash e9ef452, creating two files: index.html and estilos.css. The second prompt shows the user running 'git add remote add origin https://github.com/usuario/ProjetoSite.git', which results in a fatal error because the pathspec 'remote' does not match any files.

```
MINGW64:/C/Users/usuario

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)
$ git commit -m "Primeira versão do site"
[master (root-commit) e9ef452] Primeiro versão do site
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
create mode 100644 "estilos.css"

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)
$ git add remote add origin https://github.com/usuario/ProjetoSite.git
fatal: pathspec 'remote' did not match any files
```

Fonte: captura de tela de Git.

# Exemplo de controle de versão com Git

Figura 8 – Git Bash, acessando o Github

```
MINGW64:/C/Users/usuario

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)
$ git remote add origin https://github.com/usuario/ProjetoSite.git
Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 159.28 KiB | 9.95 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/usuario/ProjetoSite.git
 * [new branch]    master -> master
```

Fonte: captura de tela de Git.

# Gerência de configuração: atividades, ferramentas e desafios

Ferramenta Github

**Bloco 3**

Arthur Gonçalves Ferreira





# Introdução ao Github

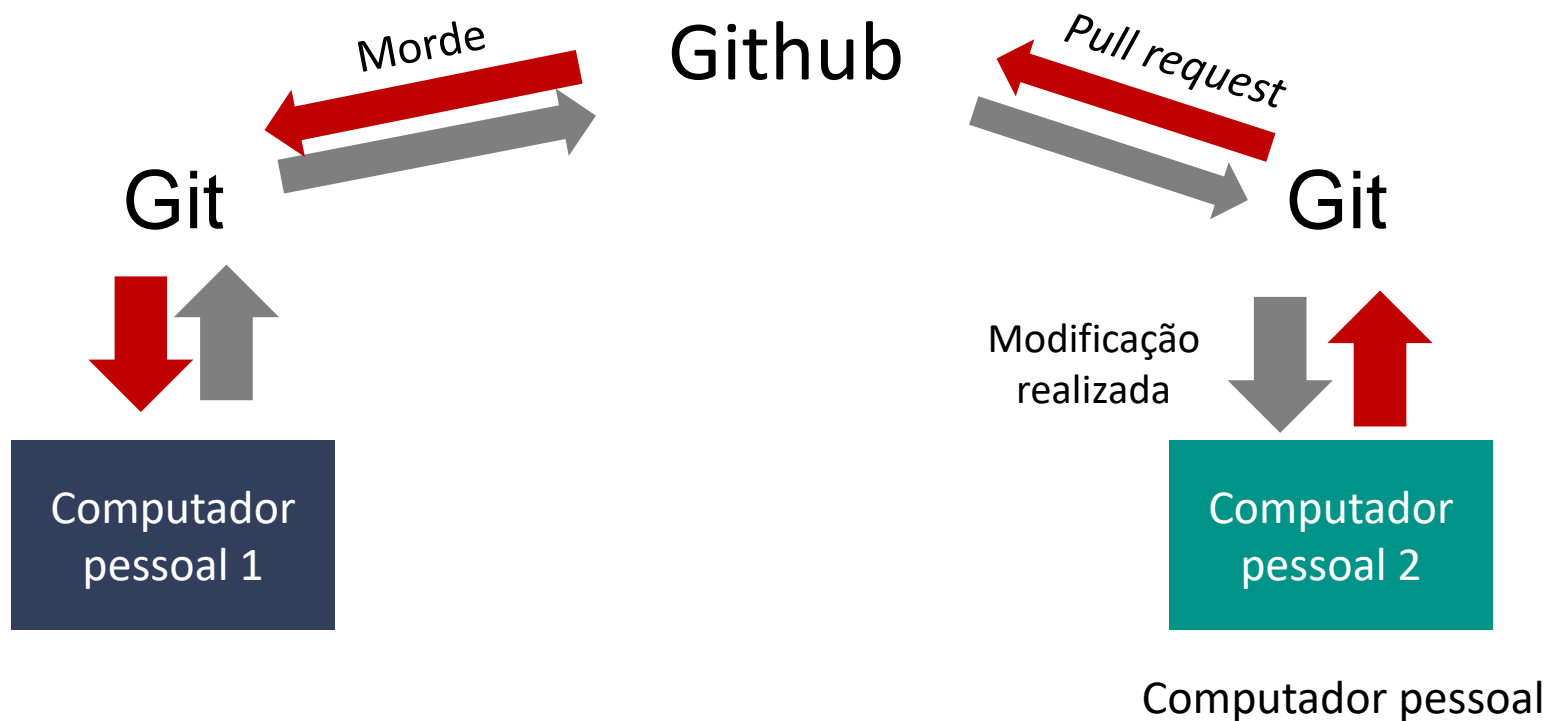
- **O que é o Github?**

- Não é a mesma coisa que Git.
- Atua com a ferramenta Git.
- Criado em 2008 por Chris Wanstrath, Tom Preston, Scott chacon e P. J Hyett.
- Atualmente pertence a Microsoft.
- É uma plataforma de hospedagem de repositório Git.
- É onde ocorre o compartilhamento de código.
- É onde o código é centralizado.



# ➤ Funcionamento do Github

Figura 9 – Github



Fonte: elaborada pelo autor.



## Cadastro no Github

- Acessar o site.
- Clicar em *sign up*.
- Realizar cadastro.





# Cadastro no Github

Figura 10 – Cadastro Github

Join GitHub

## Create your account

Username \*

Email address \*

Password \*

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.  
[Learn more.](#)

Email preferences

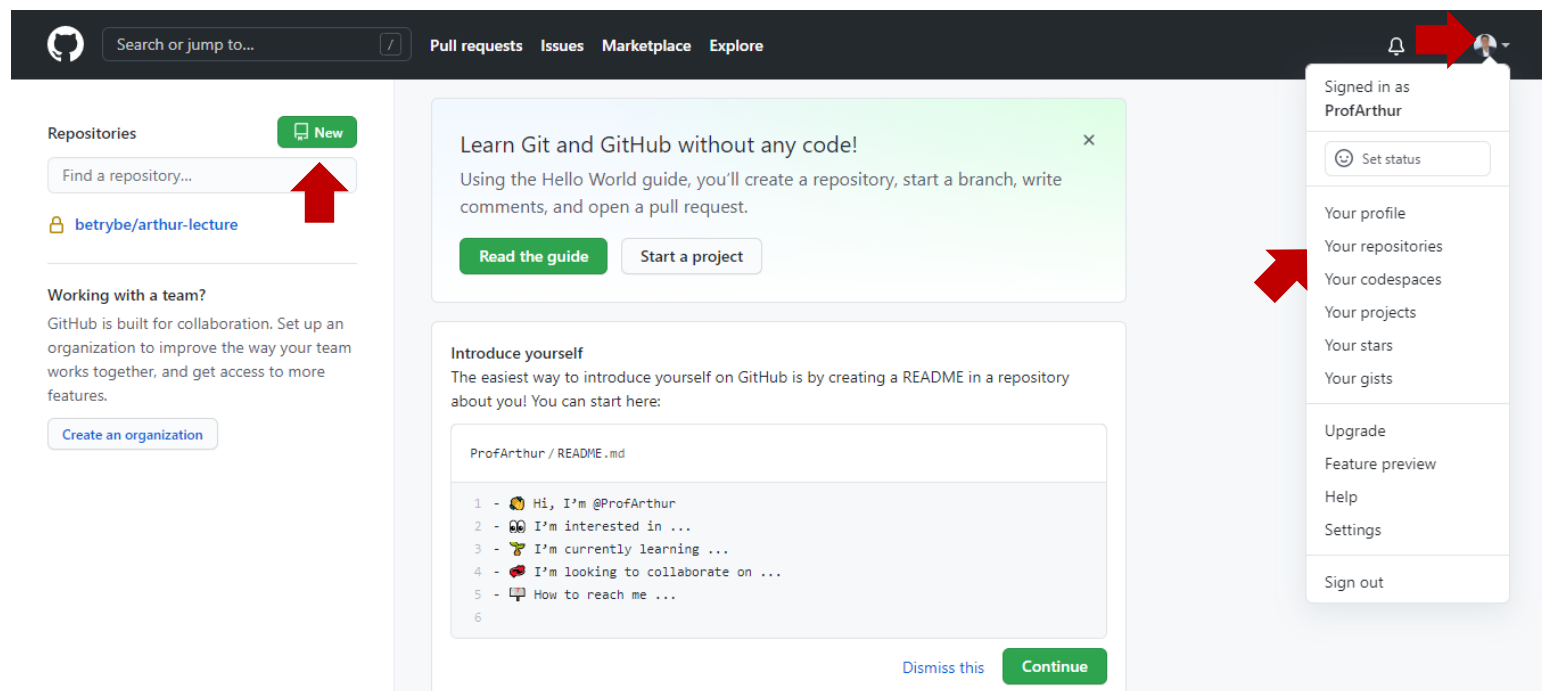
☐ Send me occasional product updates, announcements, and offers.

Verify your account

Fonte: captura de tela de Github.

# Criando repositório no Github

Figura 11 – Criando repositório no Github



Fonte: captura de tela de Github.




# ▶ Criando repositório no Github

Figura 12 – Configuração de repositório no Github

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

 ProfArthur ▾

Repository name \*

Great repository names are short and memorable. Need inspiration? How about [studious-octo-robot?](#)

Description (optional)



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:

Skip this step if you're importing an existing repository.



Add a README file

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



Add .gitignore

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



Choose a license

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


Create repository

Fonte: captura de tela de Github.

# ➤ Criando repositório no Github

**Figura 13 – Configuração de repositório no Github**

Quick setup — if you've done this kind of thing before

 Set up in Desktop or ☐ HTTPS ☐ SSH

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# ProjetoSite" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ProfArthur/ProjetoSite.git
git push -u origin main
```

Fonte: captura de tela de Github.

# Teoria em Prática

## Bloco 4

Arthur Gonçalves Ferreira



## ➤ Reflita sobre a seguinte situação

- Um desenvolvedor de sistemas está desenvolvendo um projeto de software e utilizando o Git e Github para controlar as versões do Sistema.
- Ele utiliza o Gi bash e, ao tentar aplicar o commit -m, surge a mensagem de erro: *nothing added to commit but untracked files present*.
- O que aconteceu?



# ► Norte para a resolução...

Figura 14 – Git

```
MINGW64:/C/Users/usuario

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site
$ git config --global user.name "Profarthur"

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site
$ git config --global user.email "bacharelarthur@gmail.com"

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site
$ git init
Initialized empty Git repository in C:/Users/bacha/Desktop/Informática/.git/

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Aula Word 1.pdf
  "Exerc\303\255cio Word.pdf"

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

Usuario@LAPTOP-8K5D42MD MINGW64 ~/Desktop/Site (master)
$ git add .
```

Fonte: captura de tela de Git.

# Dica do(a) Professor(a)

## Bloco 5

Arthur Gonçalves Ferreira





## Dica do professor

- **Vídeo 1.** Dicionário do programador – Git.
  - Neste vídeo, você vai conhecer um pouco mais sobre a ferramenta Git, desde a sua criação até o seu funcionamento.
  - Para assistir a este vídeo, pesquise pelo título no YouTube.
- **Vídeo 2.** Dicionário do programador – Github.
  - Neste vídeo, você vai conhecer um pouco mais sobre a ferramenta Github, desde a sua criação até o seu funcionamento.
  - Para assistir a este vídeo, pesquise pelo título no YouTube.
- **Vídeo 3.** DevMedia– Entenda o Git em 10 minutos.
  - Neste vídeo, você vai conhecer um pouco mais sobre o funcionamento da a ferramenta Git.
  - Para assistir a este vídeo, pesquise pelo título no YouTube.







# Referências

AQUILES, A.; FERREIRA, R. **Controle de versões com Git e GitHub**. v. 1. São Paulo: Casa do código, 2014.

BELL, P.; BEER, B. **Introdução ao GitHub**. São Paulo: Novatec, 2014.

BORGES, Simone de S. **Apoio de gerência de configuração de software ao ARAGAwCRe**. 2008. 102 f. Dissertação (Mestrado em Ciência da Computação) - UFSCar, São Carlos, 2008. Disponível em:  
<https://repositorio.ufscar.br/bitstream/handle/ufscar/411/2182.pdf?sequence=1&isAllowed=y>. Acesso em: 8 out. 2021.

MORAIS, Izabely S. de; ZANI, Aline. **Engenharia de software**. Porto Alegre: SAGAH, 2017.

PRESSMAN, R. S. **Engenharia de Software, uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016.

SILVERMAN, R. E. **Working introduction Git pocket guide**. [s.l.]: O'reilly, 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education do Brasil, 2018.



Bons estudos!

