

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
GEDRE - INTELIGÊNCIA EM ILUMINAÇÃO



Minicurso de Git e GitHub

Gustavo Richter Kuhn

Versão 1.1

Outubro de 2025

Conteúdo

- Sistema de Controle de Versionamento
- Diferenças entre Git e GitHub
- Instalação e Setup
- Iniciando no GitHub Desktop e Git Bash
 - Criando um repositório local
 - Clonando um repositório remoto
- Lidando com conflitos e erros

O que é um sistema de controle de versionamento?

- Um arquivo, várias versões
- Monitora alterações em um arquivo ao longo do tempo: quem e quando



Trabalho de Circuitos .pdf



Trabalho de Circuitos v1.pdf



Trabalho de Circuitos v2.pdf



Trabalho de Circuitos v3 Final.pdf



Trabalho de Circuitos v3 FinalFinal.pdf

History for [klipper](#) / [src](#) / [rp2040](#) / [main.c](#) on [master](#)

Commits on Jun 2, 2025

rp2040: Prefer larger postdiv1 on rp2040 chips

KevinOConnor committed yesterday · ✓ 1 / 1

stm32: Add comments on PLL frequency requirements to clock setup code

KevinOConnor committed yesterday · ✓ 3 / 3

Commits on Feb 26, 2025

rp2040: Fix build of rp2350

KevinOConnor committed on Feb 26

rp2040: set clock to 200Mhz

nefelim4ag authored and KevinOConnor committed on Feb 26 · ✓ 1 / 1

Commits on Nov 14, 2024

rp2040: Add rp2350 bootrom based chipid and reboot to bootloader code

KevinOConnor committed on Nov 14, 2024

rp2040: Move chipid reading to bootrom.c

KevinOConnor committed on Nov 14, 2024

rp2040: Initial rp2350 support

KevinOConnor committed on Nov 14, 2024

rp2040: Use a higher USB PLL internal frequency

KevinOConnor committed on Nov 14, 2024

O que é um sistema de controle de versionamento?

- Marca e guarda versões importantes do arquivo
- Pode servir de “backup”, revertendo alterações indesejadas ou voltando para versões funcionais
- Permite a colaboração entre vários autores

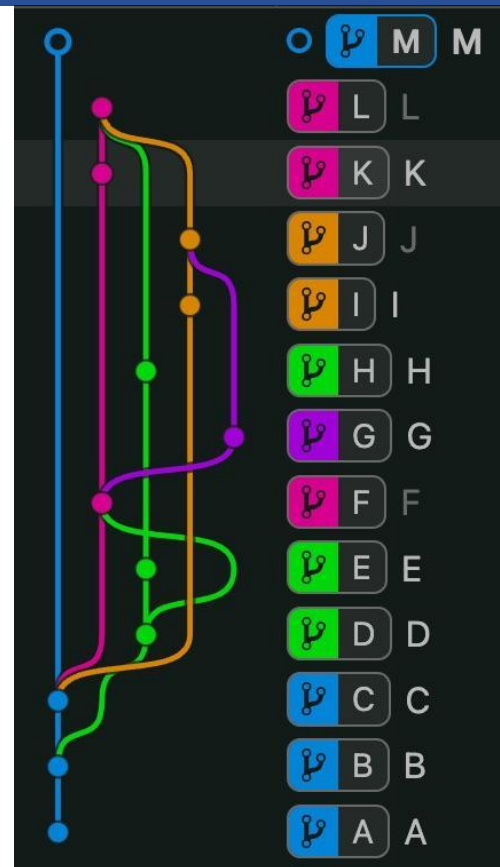
	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Contributors 452



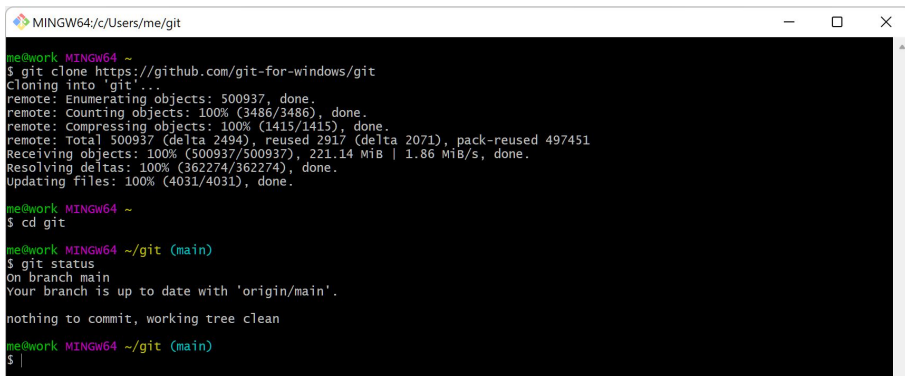
[+ 438 contributors](#)





Diferenças entre Git e GitHub

- Git é o sistema de controle de versionamento
- Git sozinho é uma CLI, acessada pelo terminal
- Repositórios locais e remotos
- GitHub é uma plataforma de repositórios remotos



```
MINGW64/c/Users/me/git
me@work MINGW64 ~
$ git clone https://github.com/git-for-windows/git
Cloning into 'git'...
remote: Enumerating objects: 500937, done.
remote: Counting objects: 100% (3486/3486), done.
remote: Compressing objects: 100% (1415/1415), done.
remote: Total 500937 (delta 2494), reused 2917 (delta 2071), pack-reused 497451
Receiving objects: 100% (500937/500937), 221.14 MiB | 1.86 MiB/s, done.
Resolving deltas: 100% (362274/362274), done.
Updating files: 100% (4031/4031), done.

me@work MINGW64 ~
$ cd git

me@work MINGW64 ~/git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

me@work MINGW64 ~/git (main)
$ |
```

Git Cheat Sheet

Git: configurations

```
$ git config --global user.name "FirstName LastName"
$ git config --global user.email "your-email@email-provider.com"
$ git config --global color.ui true
$ git config --list
```

Git: starting a repository

```
$ git init
$ git status
```

Git: staging files

```
$ git add <file-name>
$ git add <file-name> <another-file-name> <yet-another-file-name>
$ git add .
$ git add --all
$ git add -A
$ git rm --cached <file-name>
$ git reset <file-name>
```

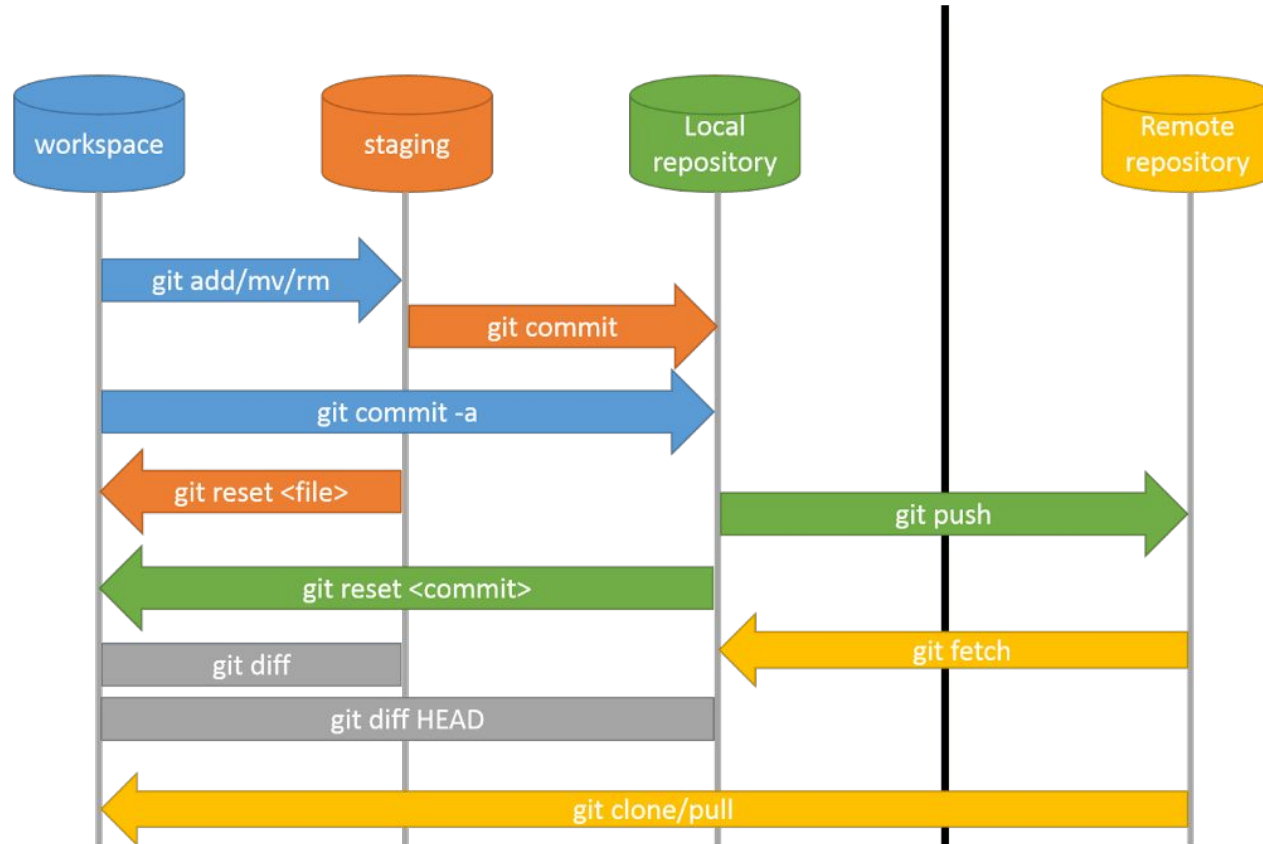
Git: committing to a repository

```
$ git commit -m "Add three files"
$ git reset --soft HEAD^
$ git commit --amend -m <enter your message>
```

Git: pulling and pushing from and to repositories

```
$ git remote add origin <link>
$ git push -u origin master
$ git clone <clone>
$ git pull
```

Workflow usando o Git



Criar uma conta no GitHub

- O GitHub será a plataforma de repositórios remotos usada nesse minicurso.
- Outros projetos podem usar servidores proprietários ou outras plataformas abordadas anteriormente.

Sign up to GitHub

Email*

Password*

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username*

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region*

For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

Email preferences

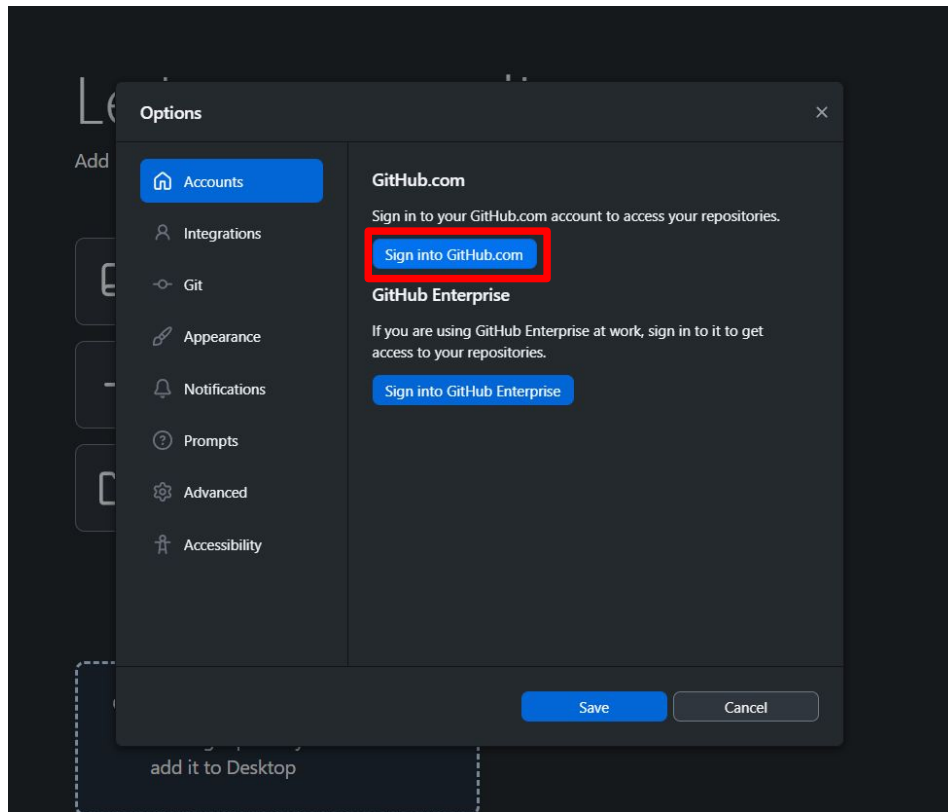
☐ Receive occasional product updates and announcements

Continue >

Instalação e Setup - GitHub Desktop

- Instalar GitHub Desktop
 - <https://desktop.github.com/download/>
- Logar com a conta no GitHub Desktop
 - File > Options > Accounts >
> Sign into GitHub.com

Instalação e Setup - GitHub Desktop



Instalação e Setup - Git

Baixar o Git pela CLI

- MacOS: `$ brew install git`
- Linux: `apt-get install git`
- Windows: `winget install --id Git.Git -e --source winget`

Usando o Git Bash

- `cd “pasta”` - trocar de pasta
- `mkdir “nome”` - criar uma pasta/diretório novo
- `ls` - lista os arquivos e pastas do diretório atual

- `git init` - inicia um repositório
- `git status` - mostra os arquivos em edição atualmente
- `git branch` - mostra o *branch* atual ou cria um novo
- `git checkout` - usado para transicionar entre *commits* e *branches*

- `git add` - adiciona arquivos à etapa de *Staging*
- `git commit` - salva arquivos ao repositório local

- `git push` - envia os *commits* realizados à um repositório remoto

Instalação e Setup - Git

Associar suas credenciais da conta GitHub

- Abrir o Git Bash
- `git config --global user.name "Seu Nome"`
- `git config --global user.email "seu.email@suainstituicao.com"`

Adicionando uma publickey ao Git Bash

Como o GitHub pode garantir que um push vindo de uma máquina é confiável e que é realmente você que está fazendo essa ação?

É criada uma chave encriptada, que é compartilhada entre o Git Bash e o GitHub.

É necessários que ambos saibam sua parte da chave.

Criar e adicionar a chave no Git Bash e adicionar no GitHub

Criando e adicionando uma publickey ao Git Bash

- Abrir o Git Bash:
 - `cd ~/.ssh && ssh-keygen`
 - Clicar enter
 - Adicionar senha (essa será a senha usada toda vez que você realizar um push ou clone)
- Copiar essa chave para o ctrl+V
 - `cat "nome_id".pub | clip`
 - "nome_id" será diferente para cada um, mas deve terminar com .pub
- Adicionar essa chave ao GitHub
 - Acessar <https://github.com/settings/keys>

Adicionando uma publickey ao GitHub

- Acessar a sua conta no GitHub e ir para suas chaves:
 - Ícone de Usuário > Configurações > Chaves SSH e GPG
- Clicar em “Nova chave”
 - Adicionar título
 - Colar a chave copiada no slide anterior
 - Salvar

Iniciando um novo projeto próprio: edição local

- Criar um novo repositório local (git init)
- Escolher um editor de texto: Sublime, VScode, NotePad++, Vim, etc...
 - Criar documentos, pastas, códigos, imagens, entre outros arquivos...
 - O git não irá registrar automaticamente esses novos arquivos.
- Recomendado criar uma *branch* relacionado ao recurso a ser implementado (git branch “nome do branch” + git checkout “nome do branch”)
- Criar um arquivo ou editar um existente
- Adicionar novos arquivos à etapa de *Staging* (git add “nome_do_arquivo”, ou git add .)

Iniciando um novo projeto próprio: edição local

- Fazer um *commit* para o repo local (git commit -m “mensagem relevante”)
- Na CLI, para visualizar arquivos editados, em *Staging*, entre outros..., é possível usar o comando git status
- O código sendo escrito e salvo será apenas salvo localmente, para conseguir salvar o código na nuvem e permitir colaboração, é necessário adicionar um *remote*.

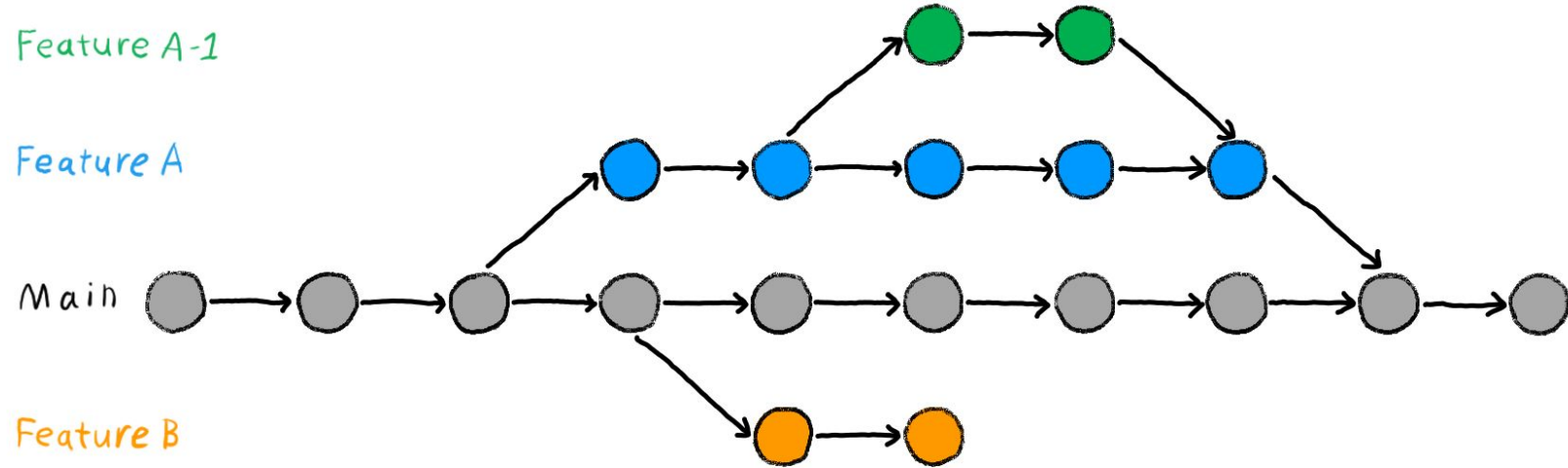
Iniciando um novo projeto: adicionado um *remote*

- No GitHub Desktop, é apenas necessário adicionar sua conta GitHub

No Git Bash, é necessário adicionar autenticação ao agente da CLI, procedimento em <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

- Adicionar um *remote*:
 - Criar um repo pelo site GitHub
 - CLI: `git remote add origin "link do repo já criado pelo GitHub"`
- Fazer um *push* para o repo remoto
 - CLI: `git push origin "branch"` ou apenas `git push origin`
- Fazer uma *pull request* à um projeto de terceiros pelo GitHub

Revisitando a Workflow de branching

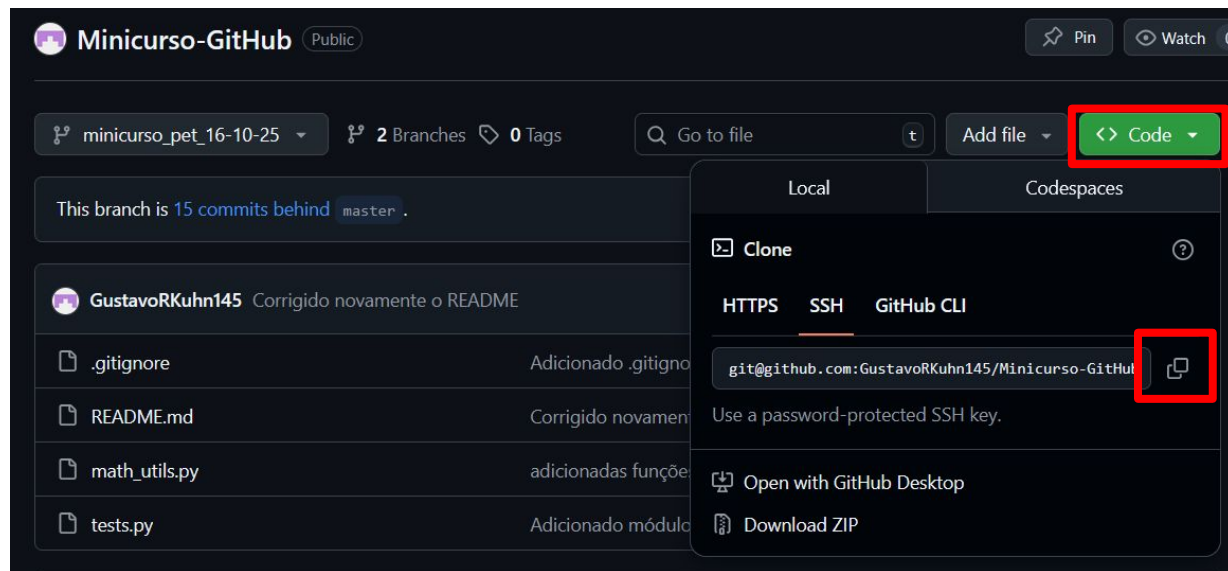


Iniciando no Git: Clonando um repositório

- Qualquer repositório público no GitHub pode ser clonado
- Pode ser usado (softwares e arquivos de build)
- Modificado (alterar algum parâmetro, renomear arquivos)
- Melhorado (contribuição open-source, implementar novos recursos e funções, criar novos arquivos)

Clonando um repositório

- Acessar o repositório no GitHub
 - https://github.com/GustavoRKuhn145/Minicurso-GitHub/tree/minicurso_pet_16-10-25



Clonando um repositório

- Code > Open with GitHub Desktop > Clone
 - CLI: `git clone “link”`
- Criar um *branch* novo
 - CLI: `git branch “nome-branch”` +
 - `git checkout “nome-branch”`

- Modificar o código-fonte usando seu editor favorito
- Adicionar as modificações à etapa de *Stating*
 - CLI: `git add “arquivo”`
- Fazer o *commit* das alterações
 - CLI: `git commit -m “mensagem relevante”`
- Publicar as mudanças ao seu branch no GitHub
 - CLI: `git push origin`
- Fazer um Pull Request para o repo original

Lidando com conflitos e erros

- Adicionar `.gitignore` para arquivos confidenciais, ferramentas de build, assets de compilação, enfim, arquivos não-essenciais
- Rebase: Volta para um *commit* específico, via SHA do *commit*
- Revert: Altera apenas as modificações feitas em um *commit* específico
- Cherry-pick: Escolhe as mudanças feitas em *commits* específicos em um *branch* para implementar em outro



gustavo.kuhn@acad.ufsm.br

Av. Roraima 1000 - Prédio 10 - Sala 440 - Santa Maria - RS



www.ufsm.br/gedre



+55 55 3220 9492



[gedre.ufsm](https://www.instagram.com/gedre.ufsm)