



GitHub

IoT para Sistemas Embarcados

Departamento de Engenharia de Controle e Automação
Instituto de Ciência e Tecnologia – UNESP – Campus Sorocaba

Prof. Dr. Dhiego Fernandes Carvalho

dhiego.fernandes@unesp.br

Objetivos

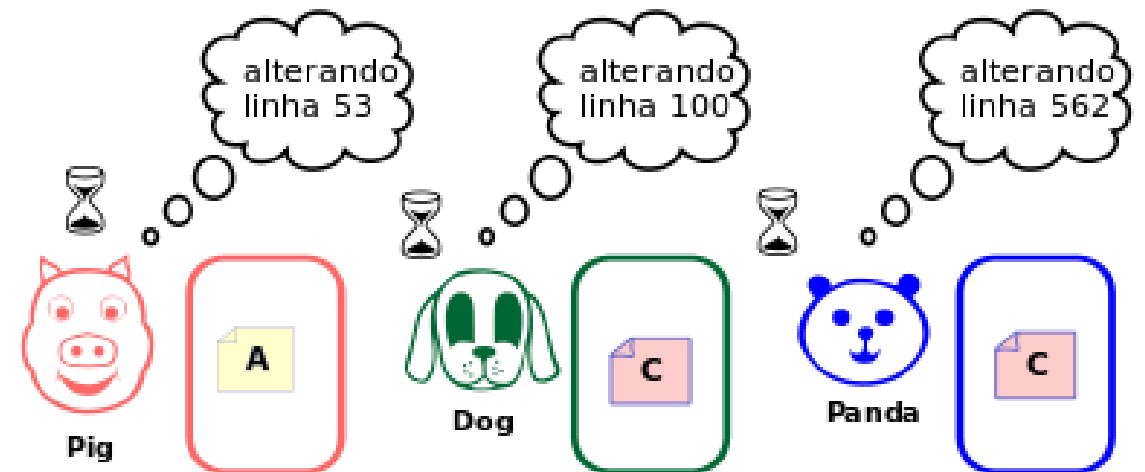
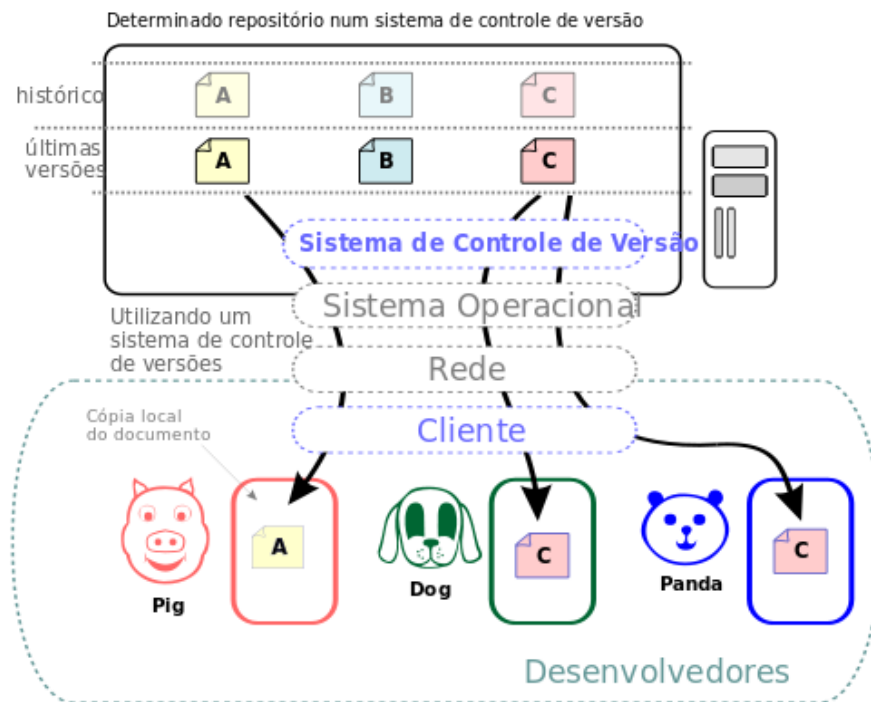
1. Entender como o GitHub foi criado.
2. Entender o que é um Sistema de Controle de Versão
3. Explorar as funcionalidades do GitHub.
4. Ensinar os principais comandos do Git.
5. Compreender a importância do Licenciamento em Projetos.
6. Utilizar o GitHub como Ferramenta de desenvolvimento e compartilhamento de Projetos.

Índice

1. Sistema de Controle de Versão
2. História do GitHub
3. Funcionalidades do GitHub
4. Comandos do Git
5. Licenciamento no GitHub
6. Conclusões

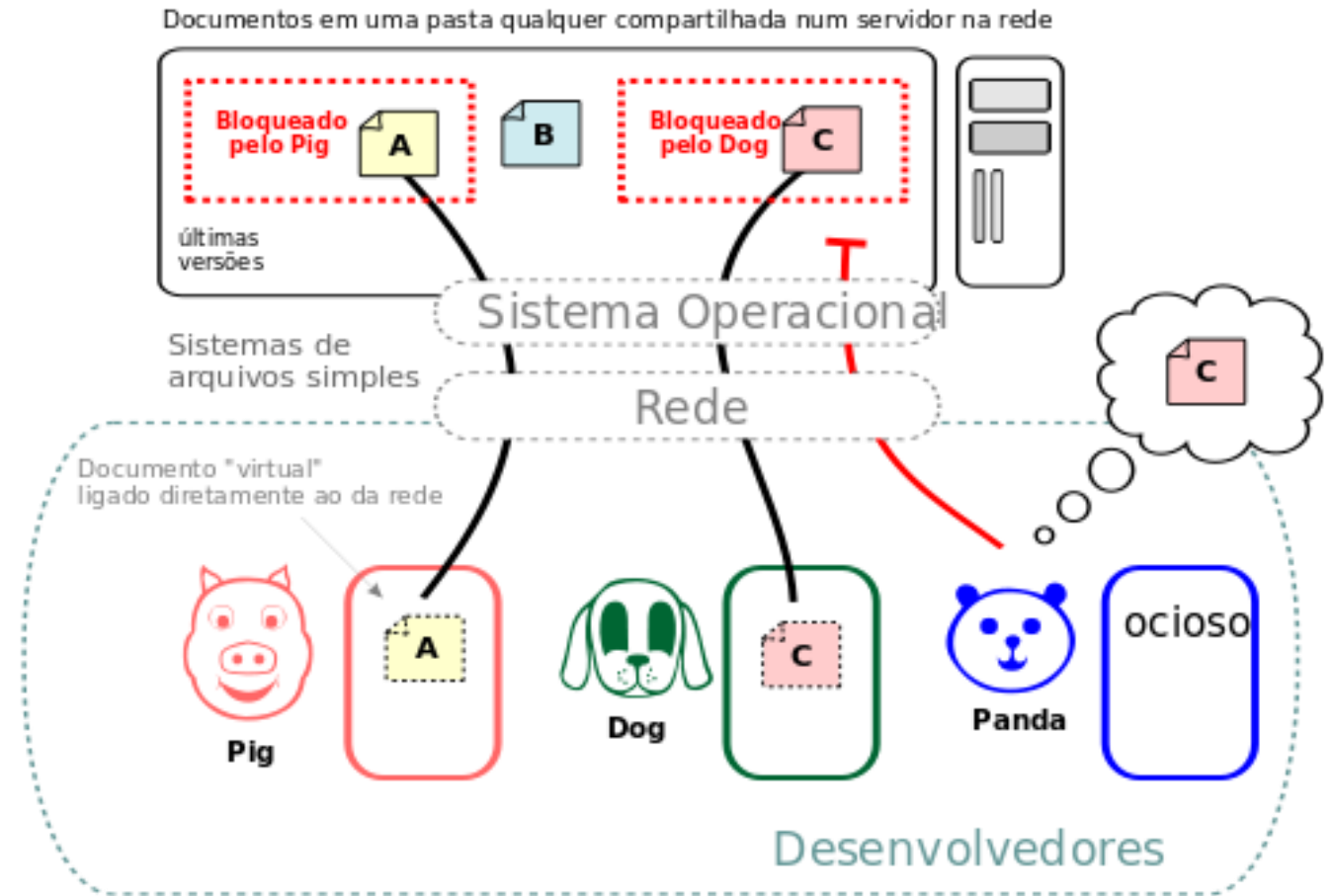
Sistema de Controle de Versão

- Um sistema de controle de versão (VCS - *Version Control System*) é uma ferramenta essencial no desenvolvimento de software, utilizada para gerenciar as alterações feitas em documentos, programas, websites etc.



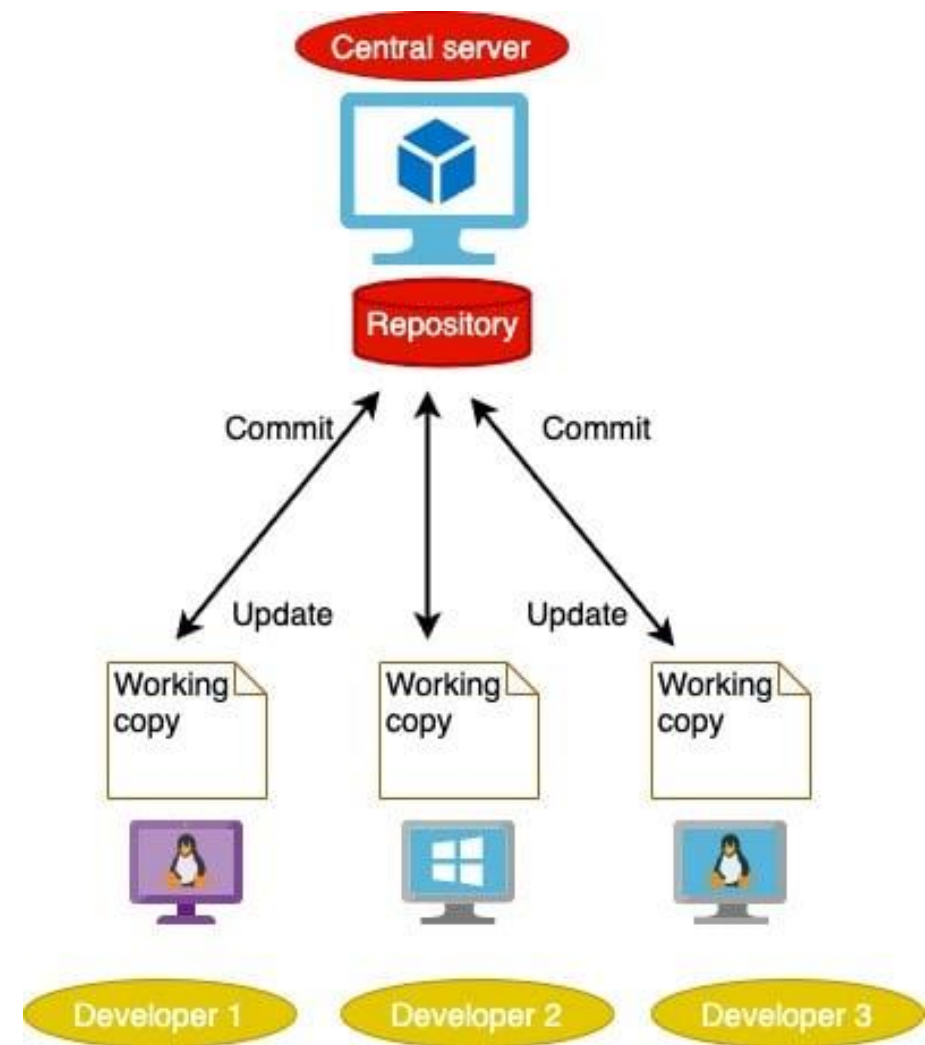
Sistema de Controle de Versão

Sem o VCS, os arquivos ficam bloqueados enquanto estão sendo utilizados ou modificados.



Principais Funcionalidades do VCS

- **Controle de Mudanças:** registro de cada alteração feita em um arquivo ou conjunto de arquivos.
- **Rastreamento de Histórico:** histórico completo de alteração, facilitando a evolução do projeto e permitindo melhor a localização de bugs.
- **Ramificação do Projeto:** utilizando recursos como *Branching* e *Merging*, permite que diversos desenvolvedores trabalhem em diferentes partes do projeto, sem interferência.



Exemplos práticos do VCS



DESENVOLVIMENTO
COLABORATIVO DE SOFTWARE



CONTROLE DE VERSÃO DE
DOCUMENTAÇÃO



PESQUISA ACADÊMICA E
REDAÇÃO DE ARTIGOS.

Sistemas de Controle de Versão

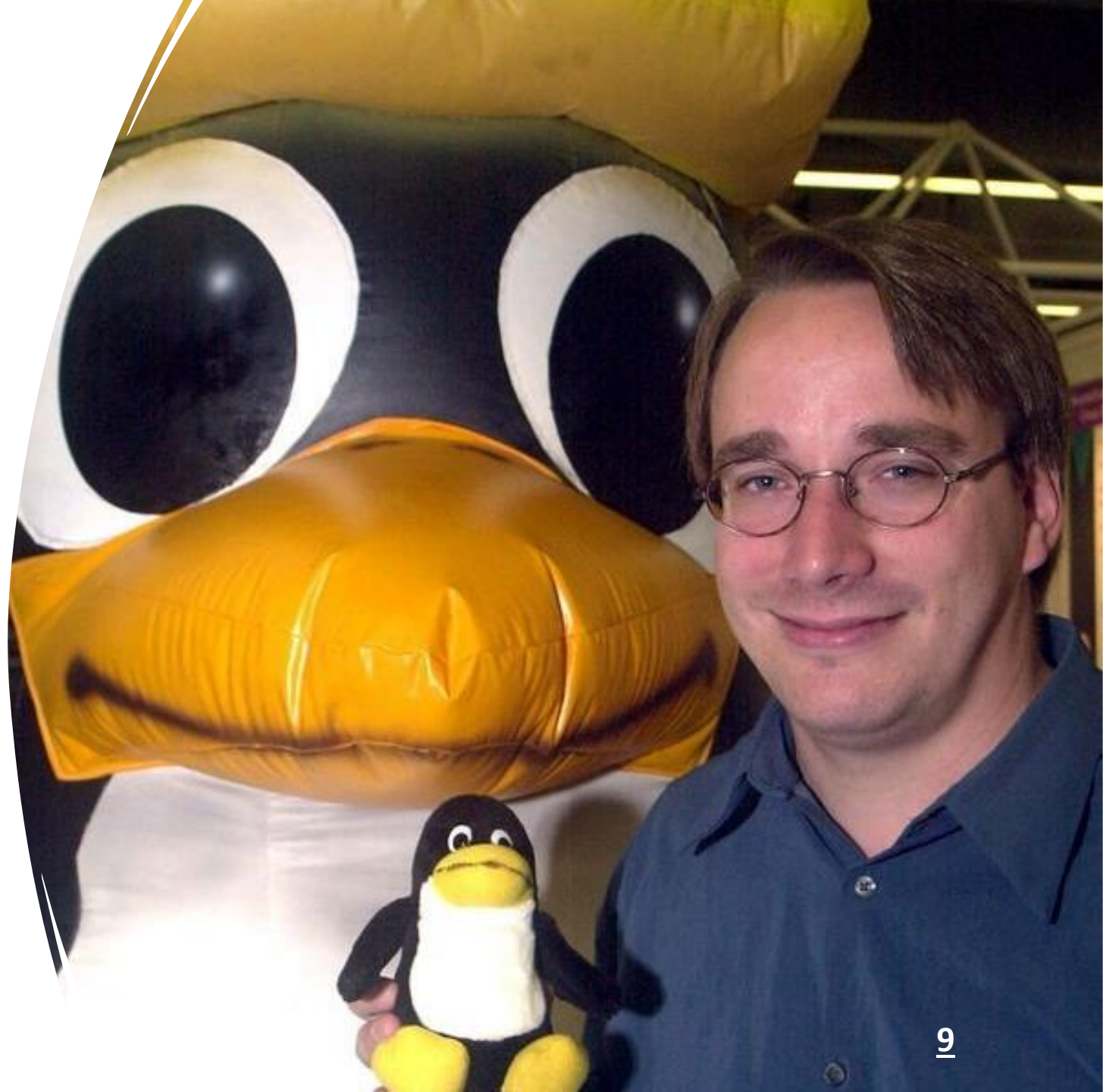
- **Subversion (SVN):** antes do git se tornar popular, era o mais utilizado.
- **Git:** o VCS mais usado. O Git é a base para plataformas como o GitHub, GitLab, e Bitbucket.
- **Mercurial:** semelhante ao Git em muitos aspectos, conhecido pela sua simplicidade e facilidade de uso.



git

História do GitHub

- O Git é um programa de VCS, sendo projeto e desenvolvido por Linus Torvalds em 2005 para o desenvolvimento do Kernel Linux.



História do GitHub

- O GitHub foi co-fundado por Chris Wanstrath, PJ Hyett, e Tom Preston-Werner em abril de 2008.
- A ideia era criar uma plataforma baseada na web que utilizasse o Git para controle de versão.



História do GitHub

- Infelizmente, em Junho de 2018, a Microsoft comprou o GitHub por \$7,5 bilhões.
- A aquisição gerou preocupações, mas a Microsoft optou por manter o GitHub de forma independente.
- O GitHub é a maior plataforma de hospedagem do mundo com mais de 83 milhões de desenvolvedores e 200 milhões de repositórios.



Funcionalidades do GitHub

- **Repositório Web:**

- Espaços de trabalho digitais onde os desenvolvedores podem armazenar, organizar, compartilhar e colaborar em projetos de software.
- Você pode adicionar manuais para a instalação e configuração do projeto.

The screenshot shows a GitHub repository page for 'IoT_for_Embedded_Devices' by user 'DhiegoFC'. The repository is public and has 1 branch (main) and 0 tags. It shows a file list with folders 'images' and 'lessons', and a file 'README.md'. The 'About' section describes it as a public repository for a graduate course at UNESP, Sorocaba campus. It also shows 0 stars and 1 watching.

IoT_for_Embedded_Devices Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file t Add file <> Code

DhiegoFC README f1d125e · 2 days ago 61 Commits

images	readme	last week
lessons	week 2	2 days ago
README.md	README	2 days ago

About

Public Repository for 'IoT for Embedded Devices', a graduate course offered by the Department of Control and Automation Engineering at the São Paulo State University (UNESP), Sorocaba campus

Readme Activity 0 stars 1 watching

Funcionalidades do GitHub

Criação de Portifólios: vários repositórios públicos pode ser uma vitrine do trabalho desenvolvido por um programador, pesquisador, professor, etc.



Dhiego Fernandes Carvalho

DhiegoFC · he/him

Professor at State University of São Paulo (UNESP), Sorocaba, Brazil.

Edit profile

State University of São Paulo (UNESP), Sorocaba, Brazil

<https://www.sorocaba.unesp.br/#/graduacao/engenharia-de-controle-e-automacao/docentes/dhiego/>

in/dhiegofc

Popular repositories

Customize your pins

TVBOX_TX9

Public

This is a manual explaining how to install restored firmwares on a TVBox , model TX9, using multitool software.

Industrial_Process_Control

Public

The public repository for the Industrial Process Control course, taught by Prof. Dr. Dhiego Fernandes Carvalho (UNESP), at State University of São Paulo, Sorocaba, Brazil.

Hydraulic_and_Pneumatic_Circuits

Public

The public repository for the Hydraulic and Pneumatic Circuits course, taught by Prof. Dr. Dhiego Fernandes Carvalho, at State University of São Paulo (UNESP), Sorocaba, Brazil.

IoT_for_Embedded_Devices

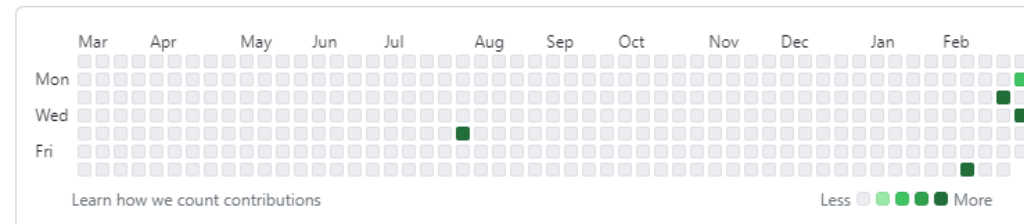
Public

Public Repository for 'IoT for Embedded Devices', a graduate course offered by the Department of Control and Automation Engineering at the São Paulo State University (UNESP), Sorocaba campus

87 contributions in the last year

Contribution settings

2024



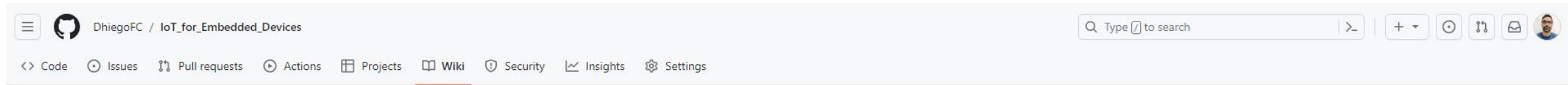
Contribution activity

March 1, 2024

DhiegoFC has no activity yet for this period.

Funcionalidades do GitHub

O GitHub oferece uma página **wiki** para criar uma documentação mais extensa do projeto.



Welcome to the IoT_for_Embedded_Devices wiki!

Wikis provide a place in your repository to lay out the roadmap of your project, show the current status, and document software better, together.


Create the first page


Aqui você pode iniciar uma página wiki


Funcionalidades do GitHub


- **Controles de Versão do Git:** Permite aos usuários rastrear e gerenciar mudanças nos códigos fonte, comparar versões, e reverter para estados anteriores se necessário.

Commits


 main ▾



 All users ▾

 All time ▾


 Commits on Feb 26, 2024



README

 DhiegoFC committed 2 days ago


f1d125e  



Merge branch 'main' of https://github.com/DhiegoFC/IoT_for_Embedded_Devices

 DhiegoFC committed 2 days ago


d690dfa  

week 2



 DhiegoFC committed 2 days ago


6499f94  

Add files via upload

 DhiegoFC committed 2 days ago

Verified

562ac1e  

 Commits on Feb 20, 2024

Funcionalidades do GitHub

- **Branches e Merge:** facilitam o desenvolvimento paralelo, sem afetar o código principal. As mudanças podem ser mescladas (merge) de volta ao branch principal após revisão.

Branches

[Overview](#) [Yours](#) [Active](#) [Stale](#) [All](#)

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	5 minutes ago			Default	

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
teste2	now		0	1	

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
teste2	now		0	1	

New branch

Funcionalidades do GitHub

- **Pull Request:** é uma solicitação para mesclar um conjunto de mudanças de um branch para outro dentro de um repositório, ou entre forks de um repositório. Serve como um mecanismo de revisão de código, permitindo que outros colaboradores revisem, discutam, e sugiram alterações antes que o código seja integrado.

The image shows a GitHub repository interface. On the left, a yellow banner indicates 'teste2 had recent pushes 1 minute ago'. Below it, a green button labeled 'Compare & pull request' is circled in red. A red arrow points from this button to the 'Open a pull request' form on the right.

Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comparisons here](#).

base: main ← compare: teste2 ✓ Able to merge. These branches can be automatically merged.

Add a title
Update README.md

Add a description
Write Preview H B I ≡ < > 🔗 | ☰ ☷ ☹ | 📎 @ ↶ ↷
Teste 2 ADD

Markdown is supported Paste, drop, or click to add files

Reviewers
No reviews

Assignees
No one—[assign yourself](#)

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Use [Closing keywords](#) in the description to automatically close issues

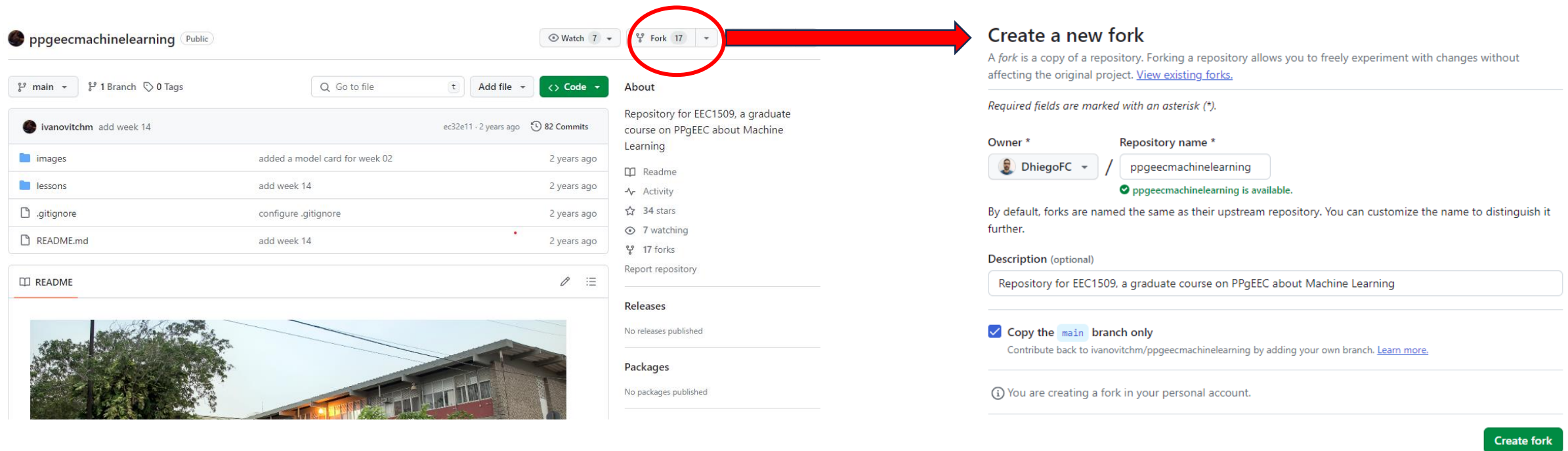
Helpful resources
[GitHub Community Guidelines](#)

[Remember, contributions to this repository should follow our \[GitHub Community Guidelines\]\(#\).](#)

[Create pull request](#)

Funcionalidades do GitHub

- **Fork:** é uma cópia exata desse repositório sob a nova conta, incluindo todo o código, branches, commits, e histórico de alterações. O repositório forkado é completamente independente do original. Após realizar as alterações no fork, pode-se enviar um pull request para o repositório original.



The screenshot shows the GitHub interface for the repository 'ppgeecmachinelearning'. The 'Fork' button is circled in red, and a red arrow points from it to the 'Create a new fork' dialog on the right.

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * DhiegoFC / Repository name * ppgeecmachinelearning

✓ ppgeecmachinelearning is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Repository for EEC1509, a graduate course on PPgEEC about Machine Learning

☒ Copy the `main` branch only

Contribute back to ivanovitchm/ppgeecmachinelearning by adding your own branch. [Learn more.](#)

ⓘ You are creating a fork in your personal account.

Create fork

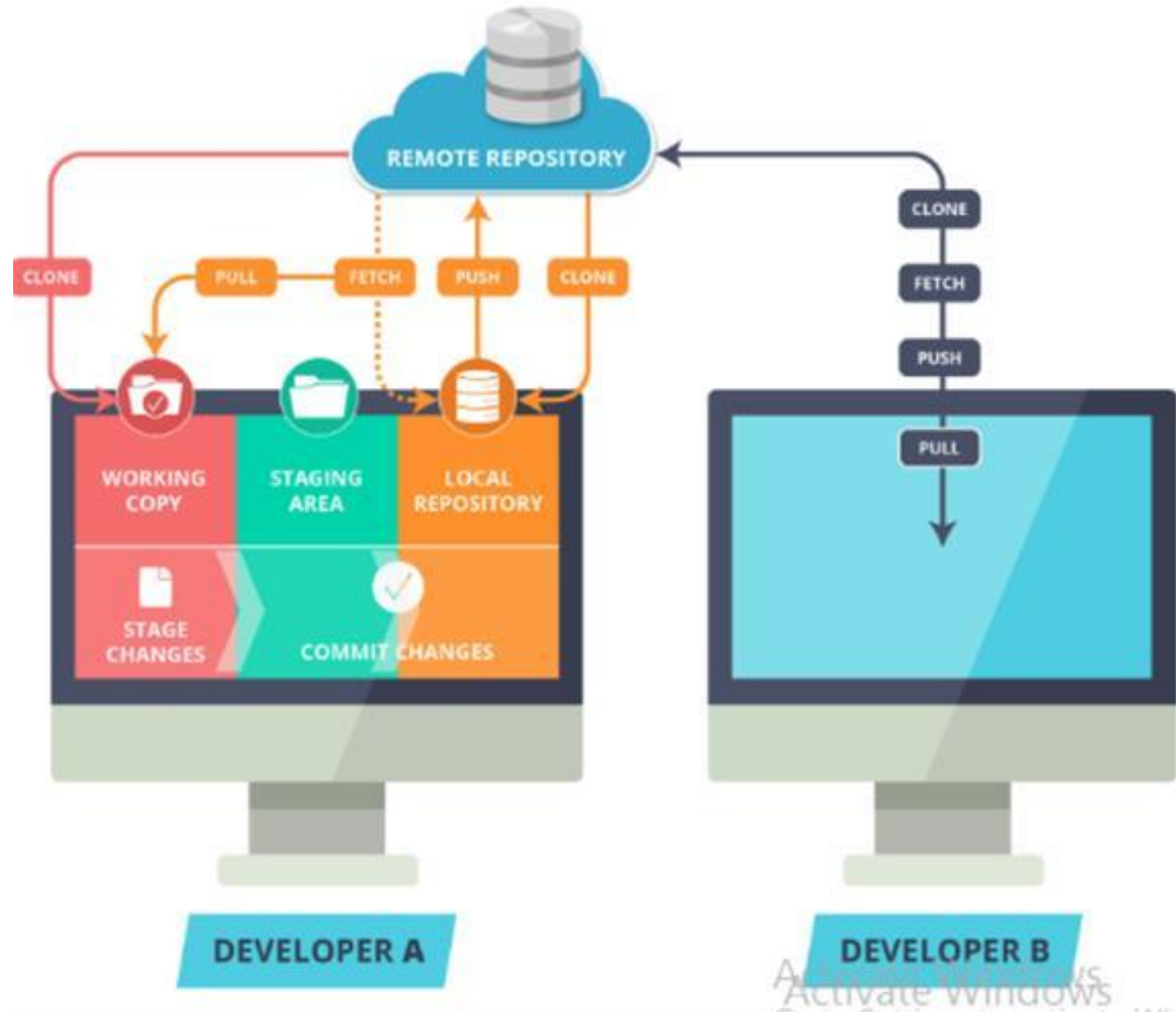
Funcionalidades do GitHub

- **Licença:** desenhada para facilitar aos desenvolvedores a escolha e aplicação de licenças de software aos seus projetos. Esta funcionalidade é crucial, pois a licença sob a qual um projeto é distribuído define como ele pode ser usado, modificado e compartilhado.

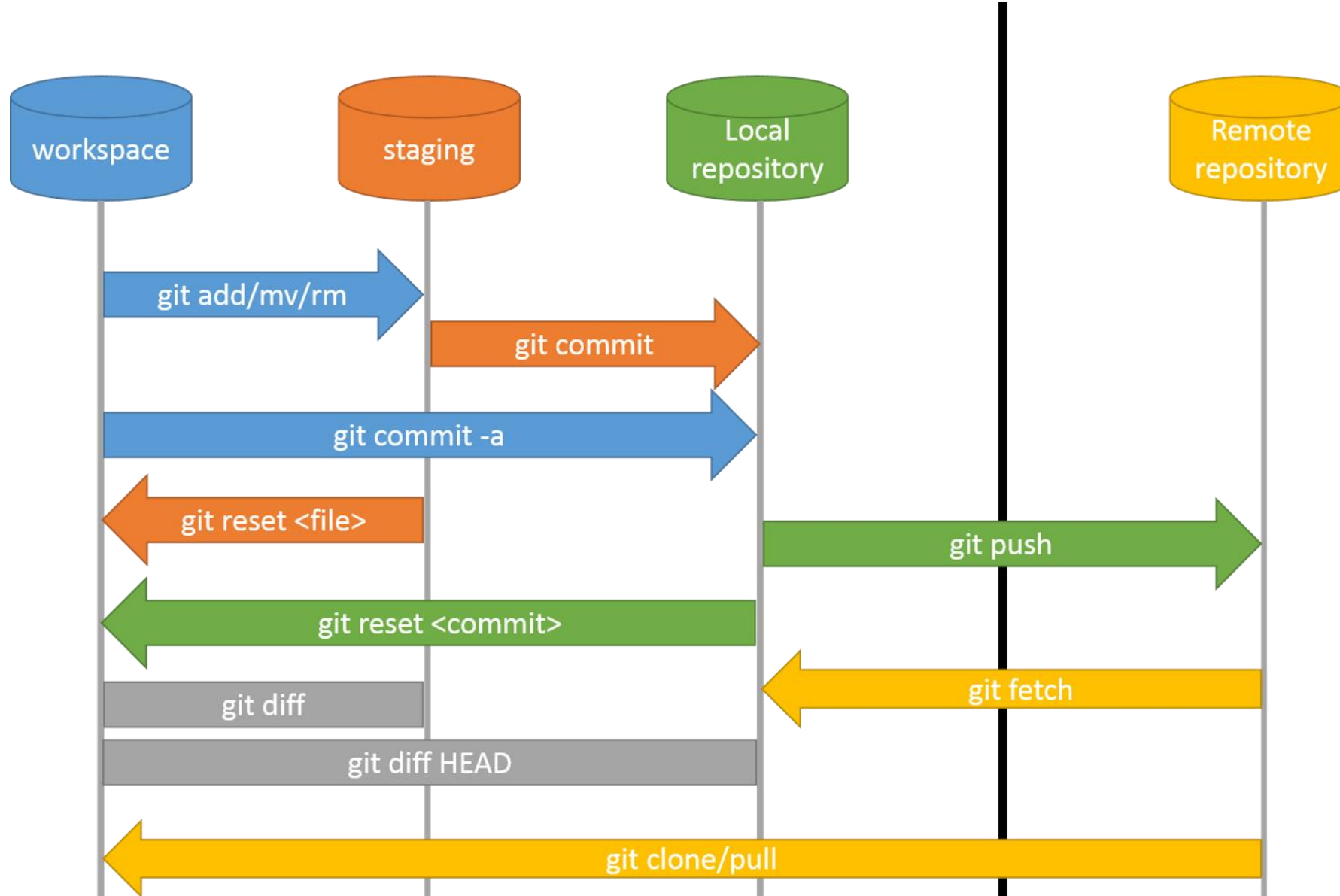
The screenshot displays the GitHub web interface for a repository named 'DhiegoFC / IoT_for_Embedded_Devices'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, the file path 'IoT_for_Embedded_Devices / LICENSE.md' is shown, with 'LICENSE.md' highlighted in a blue box. The 'main' branch is selected. In the file editor area, the 'Edit' button is active, and the 'Choose a license template' link is circled in red. A red arrow points from this link to a red text box on the right that reads: 'Você pode escolher licenças já existentes, como a GPL.' Below the editor, a text area is visible with the prompt 'Enter file contents here'.

Comandos do Git

- **Diretório de Trabalho (*Workspace*):** refere-se aos arquivos e diretórios em seu ambiente de trabalho local no qual você faz alterações.
- **Área de Preparação (*Stage Area*):** é uma camada intermediária entre sua working copy e o repositório local.
- **Repositório Local (*Local Repository*):** é uma cópia do repositório do seu projeto armazenada no seu sistema local.
- **Repositório Remoto (*Remote Repository*):** é uma versão do seu repositório hospedada em um servidor remoto



Comandos do Git



Comandos do Git

Clonar Repositório

#O “git clone” faz uma cópia local de um repositório remoto

```
C:\Users\Dhiego>git clone https://github.com/DhiegoFC/loT_for_Embedded_Devices.git
```

```
Cloning into 'loT_for_Embedded_Devices'...
```

```
remote: Enumerating objects: 10, done.
```

```
remote: Counting objects: 100% (10/10), done.
```

```
remote: Compressing objects: 100% (8/8), done.
```

```
remote: Total 100 (delta 1), reused 10 (delta 1), pack-reused 90
```

```
Receiving objects: 100% (100/100), 12.34 KiB | 5.67 MiB/s, done.
```

```
Resolving deltas: 100% (1/1), done.
```

#Entra no Diretório do Repositório

```
C:\Users\Dhiego>cd loT_for_Embedded_Devices
```

```
C:\Users\Dhiego\loT_for_Embedded_Devices>
```

Comandos do Git

Configurar Credenciais

#É importante configurar seu email e nome de usuário para caso queira fazer atualizações

```
C:\Users\Dhiego\IoT_for_Embedded_Devices >git config --global user.email "seuemail@example.com"
```

```
C:\Users\Dhiego\IoT_for_Embedded_Devices >git config --global user.name "SeuNomeDeUsuarioGitHub"
```

#Caso queira subir as modificações feitas.

```
C:\Users\Dhiego>git push origin main
```

```
Username for 'https://github.com': SeuNomeDeUsuarioGitHub
```

```
Password for 'https://SeuNomeDeUsuarioGitHub@github.com': <AQUI VOCÊ DIGITA SEU TOKEN DE ACESSO PESSOAL>
```

Comandos do Git

Baixar as mudanças do repositório sem aplicá-las

#O “git fetch origin main” baixa as atualizações do repositório remoto para o repositório local, #verificando as suas mudanças.

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git fetch origin main
```

```
From https://github.com/DhiegoFC/IoT_for_Embedded_Devices
```

```
* branch      main    -> FETCH_HEAD
```

#O “git merge origin/main” mescla as atualizações no repositório local no workspace.

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git merge origin/main
```

```
Updating a1b2c3d..e4f5g6h
```

```
Fast-forward
```

```
README.md | 2 ++
```

```
1 file changed, 2 insertions(+)
```


Comandos do Git

Atualizar o repositório

#O “git pull” atualiza o diretório

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git pull origin main
```

```
remote: Enumerating objects: 27, done.
```

```
remote: Counting objects: 100% (27/27), done.
```

```
remote: Compressing objects: 100% (15/15), done.
```

```
remote: Total 22 (delta 7), reused 22 (delta 7), pack-reused 0
```

```
Unpacking objects: 100% (22/22), done.
```

```
From https://github.com/DhiegoFC/IoT_for_Embedded_Devices
```

```
* branch      main    -> FETCH_HEAD
```

```
abc123d..def456g main    -> origin/main
```

```
Updating abc123d..def456g
```

```
Fast-forward
```

```
yourfile.txt | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

Comandos do Git

Criando novas Branches

#O “git branch” adiciona uma nova branch

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git branch nova_branch
```

#O “git checkout” acessa a nova branch

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git checkout nova_branch
```

```
Switched to branch 'nova_branch'
```

Comandos do Git

Status do Repositório

#O “git status” mostra o status do repositório

C:\Users\Dhiego\IoT_for_Embedded_Devices>**git status**

On branch main

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

nothing to commit, working tree clean

Comandos do Git

Enviar mudanças para o Repositório Remoto

#O “git add” adiciona o arquivo modificado a cópia a ser enviada.

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git add README.md
```

#O “git commit” cria a cópia a ser enviado ao repositório remoto

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git commit -m "Update README.md with new information"
```

```
[main 1d3f5ab] Update README.md with new information
```

```
1 file changed, 2 insertions(+)
```

#Envia a cópia para o repositório remoto

```
C:\Users\Dhiego\IoT_for_Embedded_Devices>git push origin main
```

```
Username for 'https://github.com': SeuNomeDeUsuarioGitHub
```

```
Counting objects: 3, done.
```

```
Delta compression using up to 8 threads.
```

```
Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (3/3), 330 bytes | 330.00 KiB/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0)
```

```
To https://github.com/DhiegoFC/IoT_for_Embedded_Devices.git
```

```
e3f1c2a..1d3f5ab main -> main
```

Comandos do Git

Merge de uma nova branch

#Entra na branch principal

C:\Users\Dhiego\IoT_for_Embedded_Devices>**git checkout main**

Switched to branch 'main'

#Baixa as atualizações da branch principal

C:\Users\Dhiego\IoT_for_Embedded_Devices>**git pull origin main**

From https://github.com/DhiegoFC/IoT_for_Embedded_Devices

* branch main -> FETCH_HEAD

Already up to date.

#Merge da nova_branch com a main


C:\Users\Dhiego\IoT_for_Embedded_Devices>**git merge nova_branch**

Updating a1b2c3d..e4f5g6h

Fast-forward

README.md | 2 ++

1 file changed, 2 insertions(+)



É recomendável
fazer o “Pull
Request” antes
do Merge

Comandos do Git

Verificar as diferenças do workspace da stage area

#Adiciona modificações no README.md

```
C:\Users\Dhiego\MeuProjeto>echo "Nova alteração no README" >> README.md
```

#O “git diff” verifica as diferenças

```
C:\Users\Dhiego\MeuProjeto>git diff
```

```
diff --git a/README.md b/README.md
```

```
index 4c2a9d..e5f6g7h 100644
```

```
--- a/README.md
```

```
+++ b/README.md
```

```
@@ -1 +1,2 @@
```

```
Texto original no README.
```

```
+Nova alteração no README
```

Comandos do Git

Deletar alterações que foram commitadas, sem alterar a Área de Trabalho (workspace)

#Adiciona modificações no README.md, adiciona na stage area (add) e em seguida no repositório local (commit).

```
C:\Users\Dhiego\MeuProjeto>echo "Alteração" >> README.md
```

```
C:\Users\Dhiego\MeuProjeto>git add README.md
```

```
C:\Users\Dhiego\MeuProjeto>git commit -m "Atualiza README.md"
```

```
[main abcdef1] Atualiza README.md
```

```
1 file changed, 1 insertion(+)
```

#Remove as modificações que foram commitadas, sem modificá-la na área de trabalho (workspace)

```
C:\Users\Dhiego\MeuProjeto>git reset HEAD~1
```

```
Unstaged changes after reset:
```

```
M    README.md
```

#Mostra o status

```
C:\Users\Dhiego\MeuProjeto>git status
```

```
On branch main
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
modified: README.md
```

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

Comandos do Git

Deletar alterações que foram commitadas, sem alterar a Área de Trabalho (workspace)

#Adiciona modificações no README.md, adiciona na stage area (add) e em seguida no repositório local (commit).

```
C:\Users\Dhiego\MeuProjeto>echo "Nova Alteração" >> README.md
```

```
C:\Users\Dhiego\MeuProjeto>git add README.md
```

```
C:\Users\Dhiego\MeuProjeto>git commit -m "Atualização que será desfeita"
```

```
[main abcdef2] Atualização que será desfeita
```

```
1 file changed, 1 insertion(+)
```

#Remove as modificações que foram commitadas e também na área de trabalho (workspace).

```
C:\Users\Dhiego\MeuProjeto>git reset --hard HEAD~1
```

```
HEAD is now at abcdef1 Atualiza README.md
```

#Mostra o status

```
C:\Users\Dhiego\MeuProjeto>git status
```

```
On branch main
```

```
nothing to commit, working tree clean
```


Licenciamento

- O licenciamento no GitHub é uma parte fundamental do compartilhamento e da colaboração em projetos de software, pois define legalmente como outros podem usar, modificar e distribuir o seu trabalho.

License Types

Copyleft

*GPL

viral

attribution

protection
against
liability

permissive

MIT, BSD, ...

attribution

protection
against
liability

patents,
trademark
rules in details

Creative Commons

building blocks:

→ attribution

→ virality

→ no commercial
use ⚡

→ no derivatives ⚡








Public Domain

“Don’t care”

liability? ⚡

Para que serve o licenciamento?

- Define Permissões e Restrições.
- Protege os Direitos do Criador.
- Facilita a Colaboração Aberta.

						
Type	Permissive	Permissive	Permissive	Copyleft	Copyleft	Copyleft
Provides copyright protection	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE
Can be used in commercial applications	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE
Provides an explicit patent license	✓ TRUE	✗ FALSE	✗ FALSE	✗ FALSE	✗ FALSE	✗ FALSE
Can be used in proprietary (closed source) projects	✓ TRUE	✓ TRUE	✓ TRUE	✗ FALSE	✗ FALSE partially	✗ FALSE for web
Popular open-source and free projects	Kubernetes Swift Firebase	Django React Flutter	Angular.js jQuery, .NET Core Laravel	Joomla Notepad++ MySQL	Qt SharpDevelop	SugarCRM Launchpad

Licenciamento

- Caso queira saber mais sobre qual licença usar no seu projeto no Github, consulte o choosealicense.com

Choose an open source license

An open source license protects contributors and users. Businesses and savvy developers won't touch a project without this protection.

Which of the following best describes your situation?



I need to work in a community.

Use the **license preferred by the community** you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to **add a license**.



I want it simple and permissive.

The **MIT License** is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

Babel, **.NET**, and **Rails** use the MIT License.



I care about sharing improvements.

The **GNU GPLv3** also lets people do almost anything they want with your project, *except* distributing closed source versions.

Ansible, **Bash**, and **GIMP** use the GNU GPLv3.

What if none of these work for me?

My project isn't software.

There are licenses for that.

I want more choices.

More licenses are available.

I don't want to choose a license.

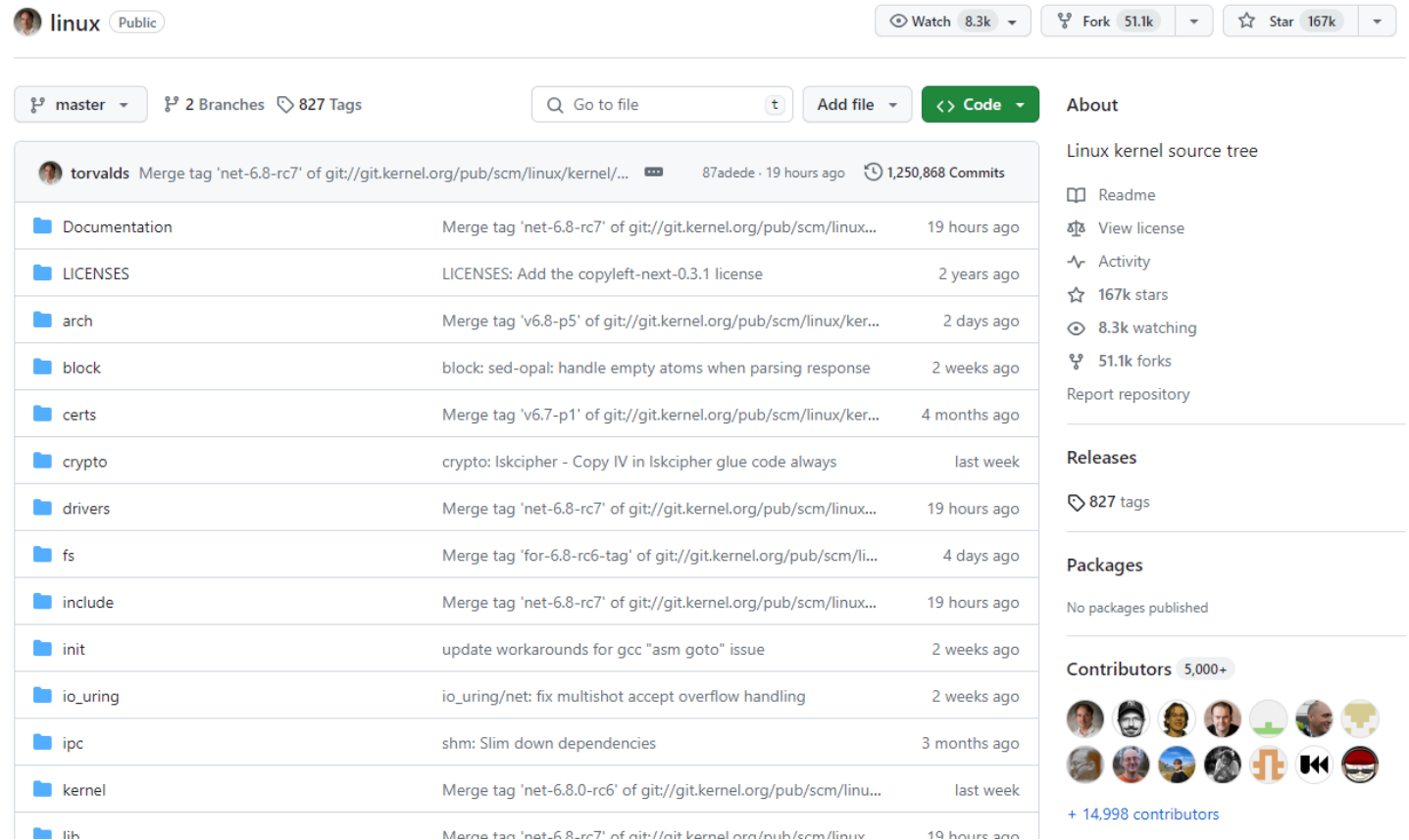
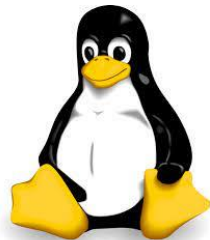
Here's what happens if you don't.

Projetos famosos no GitHub

- **Kernel do Linux:**

- Mantido por Linus Torvalds.
- Está sob a licença GPLv2.

- Página:
github.com/torvalds/linux

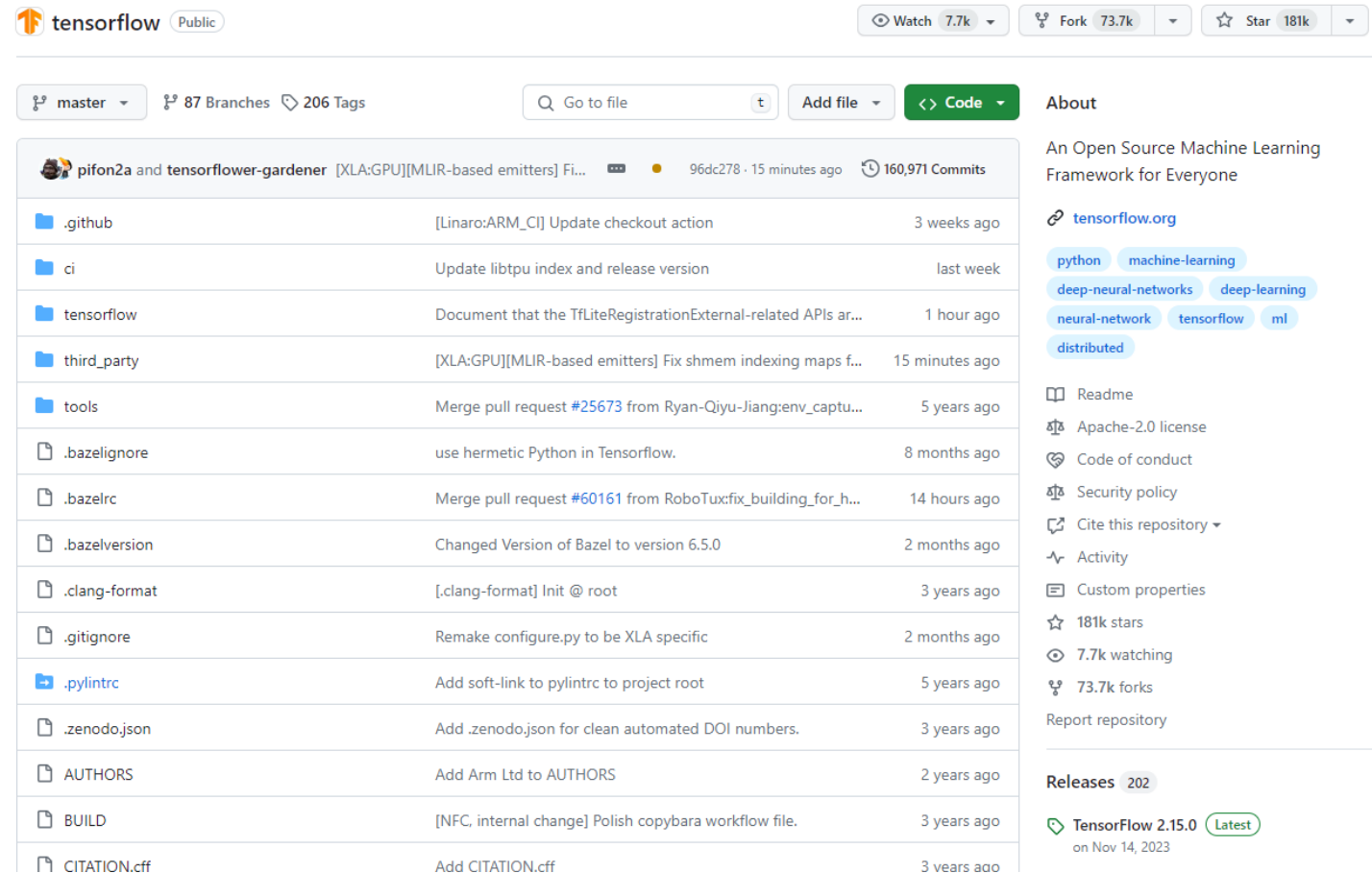


The screenshot shows the GitHub repository for the Linux kernel source tree, maintained by Linus Torvalds. The repository is public and has 167k stars, 51.1k forks, and 8.3k watchers. It contains 2 branches and 827 tags. The main branch is master. The repository is described as the "Linux kernel source tree". The file list shows various directories and files, including Documentation, LICENSES, arch, block, certs, crypto, drivers, fs, include, init, io_uring, ipc, kernel, and lib. The most recent commit is a merge tag 'net-6.8-rc7' of git://git.kernel.org/pub/scm/linux/kernel/... by 87ade, 19 hours ago, with 1,250,868 commits in total. The right sidebar shows the repository's statistics, including 167k stars, 51.1k forks, and 8.3k watchers. It also lists the repository's license, activity, and releases.

Projetos Famosos no Github

- **TensorFlow.**

- Biblioteca de código aberto para aprendizado de máquina e inteligência artificial.
- Licença: Apache Licence 2.0
- URL: github.com/tensorflow/tensorflow

A screenshot of the TensorFlow GitHub repository page. The repository is 'tensorflow/tensorflow', marked as 'Public'. It shows 87 branches, 206 tags, and 160,971 commits. The file list includes folders like .github, ci, tensorflow, third_party, and tools, as well as files like .bazelignore, .bazelrc, .bazelversion, .clang-format, .gitignore, .pylintrc, .zenodo.json, AUTHORS, BUILD, and CITATION.cff. The right sidebar shows the 'About' section with the description 'An Open Source Machine Learning Framework for Everyone', links to tensorflow.org, and various tags like python, machine-learning, deep-neural-networks, deep-learning, neural-network, tensorflow, ml, and distributed. It also lists the license (Apache-2.0), code of conduct, security policy, and activity. The 'Releases' section shows the latest release, TensorFlow 2.15.0, dated Nov 14, 2023.

Projetos Famosos no Github

- **Kubernetes**

- Sistema de orquestração de contêineres.
- Licença: Apache Licence 2.0
- URL:
github.com/kubernetes/kubernetes



kubernetes Public

Watch 3.2k Fork 38.1k Star 105k

master 53 Branches 1,078 Tags

Go to file Add file Code

About
Production-Grade Container Scheduling and Management
kubernetes.io
go kubernetes containers cncf
Readme
Apache-2.0 license
Code of conduct
Security policy
Activity
Custom properties
105k stars
3.2k watching
38.1k forks
Report repository

Releases 656
Kubernetes v1.29.2 Latest
2 weeks ago
+ 655 releases

Packages

k8s-ci-robot	Merge pull request #123554 from jsafrane/selinux-rwx-tests	055b517 · 1 hour ago	121,355 Commits
.github	Add new contribex leads to sig-contribex-approvers	last year	
CHANGELOG	CHANGELOG: Update directory for v1.30.0-alpha.3 release	2 days ago	
LICENSES	agnhost: merge registry.k8s.io/stress:v1 (github.com/vishh/st...	2 weeks ago	
api	Merge pull request #123560 from ivelichkovich/master	9 hours ago	
build	Merge pull request #123393 from cblecker/fix-make	4 days ago	
cluster	Revert "Update kubedns and nodelocaldns to v1.23.0"	yesterday	
cmd	Merge pull request #122293 from mengjiao-liu/controller-re...	20 hours ago	
docs	Make root approval non-recursive	2 years ago	
hack	Configure cpumanager policy options in local-up-cluster	6 hours ago	
logo	logo: better alignment of layers	2 years ago	
pkg	Merge pull request #123399 from pohly/test-context-use-ca...	7 hours ago	
plugin	system:kube-scheduler: extend the RBAC with pods/finalizers	4 days ago	
staging	Merge pull request #123613 from aojea/revert_ccm	3 hours ago	
test	Merge pull request #123554 from jsafrane/selinux-rwx-tests	1 hour ago	

Projetos Famosos no Github

- **Visual Studio Code**

- Editor de código-fonte desenvolvido pela Microsoft.
- Licença: MIT Licence
- URL:
<https://github.com/microsoft/vscode>



A screenshot of the Visual Studio Code GitHub repository page. The page shows the repository name 'vscode' with a 'Public' badge. It displays statistics: 3.3k watches, 27.3k forks, and 157k stars. The main content area shows a list of recent commits by user 'bhavyaus', including updates to configuration files, Dockerfiles, and issue milestones. The right sidebar contains links to the project website, documentation, and a list of releases, with the latest release being 'February 2024'.

Conclusões

- Nesta aula foi visto a história do GitHub, sendo o software Git desenvolvido pelo Linus Torvalds.
- Foi descrito as funcionalidades do GitHub.
- Foi ensinado os principais comandos do Git para clonagem, alteração e upload de um diretório remoto.
- Foi explicado a importância do Licenciamento nos Projetos desenvolvidos no GitHub.
- Foi mostrado os projetos de Software mais famosos no Github.

DÚVIDAS?

Exercícios

- Crie uma conta no GitHub, caso você não a tenha.
- Crie um repositório para algum projeto de alguma disciplina anterior que você já tenha desenvolvido em outra disciplina.
- Faça uma breve descrição do projeto no README.md.
- Clone o projeto para a sua máquina pessoal. Utilize os comandos do Git aprendidos nessa aula.