

Índice

- [lista de comando](#)
- [Comandos git](#)
- [Retirando da área](#)
- [Criando Branch](#)
- [Git merge](#)
- [Criando um Repositório Remoto](#)
- [Git push sincronizar o repositório local com o remoto](#)
- [Git pull sincronizar o repositório remoto com o local](#)
- [Fazendo o Download de repositórios](#)
- [Pull Request GitHub](#)
- [Links / Referências](#)

lista de comando !!!

Um rápido tutorial de comando no terminal!!!

- **pwd** *Mostra o caminho do diretório atual*
- **mkdir <nome-do-diretório>** comando para criar pastas
- **touch . ponto**
 - Exemplo **touch olamundo.txt** ou **touch ImpostoRenda.doc** ou **touch Planilha-de-Contas.xls** assim por diante.

Um detalhe no momento de criar o arquivo de preferência em utilizar um destes caracteres (Underline "_" ou hífen "-" ou CamelCase "ContasApagar") para separa as palavras a fim de tornar mais fácil selecionar a palavra pelo TERMINAL.

- **ls** *Comando que lista os arquivos no diretório atual*
- **cd** comando para mudar de pasta no terminal
 - Exemplo **cd home/Area-de-trabalho/facul**
 - **cd ..** comando que retorna um nível acima
 - Exemplo: o diretório atual e o facul usando o **cd ..** você voltaria para a Area-de-trabalho
 - **cd ~** comando para voltar o diretório raiz.
 - Exemplo: como você esta no diretório facul com o **cd ~** você ira para o home

Comandos git

Configuração do usuário

```
git config --global user.name "Seu nome"
git config --global user.email "seu_email@.com"
```

Conferindo as informações

```
git config --list "comando completo"
git config -l "comando curto"
```

Pedindo ajuda

```
git help "mostra a pagina de ajuda"
git help <comando> "mostra ajuda para o comando digitado"
```

Iniciando um repositório

```
git init "comando que inicia o repositório git"
```

Verificando alteração

```
git status "comando para ver as alterações nos repositório" "diga-se de passagem um dos comando mais digitados"
```

Adicionar arquivos e diretórios ao repositório

```
git add <nome-do-arquivo> "nome do arquivo ou diretório"
git add . "adicionado todos os arquivos no repositório"
```

Salvando os arquivos

```
git commit "descrição detalhada do commit geralmente abre o editor de text padrão do terminal! para passar a mensagem"
```

descrição curta >

```
git commit -m "digite sua mensagem entre as aspas "
```

Verificando os Commits

```
git log "mostra o commit"
git log --pretty=oneline
git log --pretty=format:"%h -%an, %ar : %s"
git log --all --graph --decorate
git log --all --graph --decorate --oneline --simplify-by-decoration
```

Buscando os Commits

```
git show <numero da hash> "hash e um identificador único criado no momento do commit"
```

Um atalho para o add/commit

Uma vez que você já tenha add o arquivo no repositório pode-se usar um combinação de comando no caso o "git add" com o "git commit -m" ficaria assim

```
git commit -a -m "digite a mensagem"
git commit -am "digite a mensagem aqui"
git commit --amend "renomear o ultimo commit"
```

Retirando da área

Desfazer alterações não "commitadas"

Exemplo: temos dois arquivos um chamado **DocumentoA.docx** e outro **DocumentoX.docx** só que por um erro usamos o comando `git add .` mas o documento que é para ser add a *área de Stage* era o **DocumentoX.docx** então podemos usar o comando `git reset HEAD DocumentoA.docx` com isso o documentoA sai da área de Stage

```
git reset HEAD <nome-do-arquivo>
```

Muito cuidado com o checkout

Não use esse comando a menos que você tenha certeza absoluta que não quer o arquivo .

Exemplo arquivo.txt está commitado "salvo" com essa mensagem

```
este arquivo e salvo na extensão md de markdown muito utilizado para passar informações,
post de blog "em especial blog de devs" por se tratar de um arquivo muito fácil de escrever e
formatar
```

Então foi usado o comando

```
git commit -am "informação básica"
```

logo após foi add essa outra mensagem

```
este arquivo e salvo na extensão md de markdown muito utilizado para passar informações,
post de blog "em especial blog de devs" por se tratar de um arquivo muito fácil de escrever e
formatar
```

```
essa Linha se auto-destruirá com o comando "git checkout --arquivo.txt"
```

pois utilizando o comando a alteração feita "os dados" nesse arquivo será perdida não podendo ser retornada

```
git checkout -- <nome-do-arquivo>
```

Criando Branch

O branch serve para desviar o fluxo de trabalho normal possibilitando desenvolvimento paralelo do mesmo arquivo

```
git branch "lista os branch e mostra o branch atual"

git branch <nome-do-branch> cria o branch "ramo"

git branch -v "lista o branch com o ultimo commit"

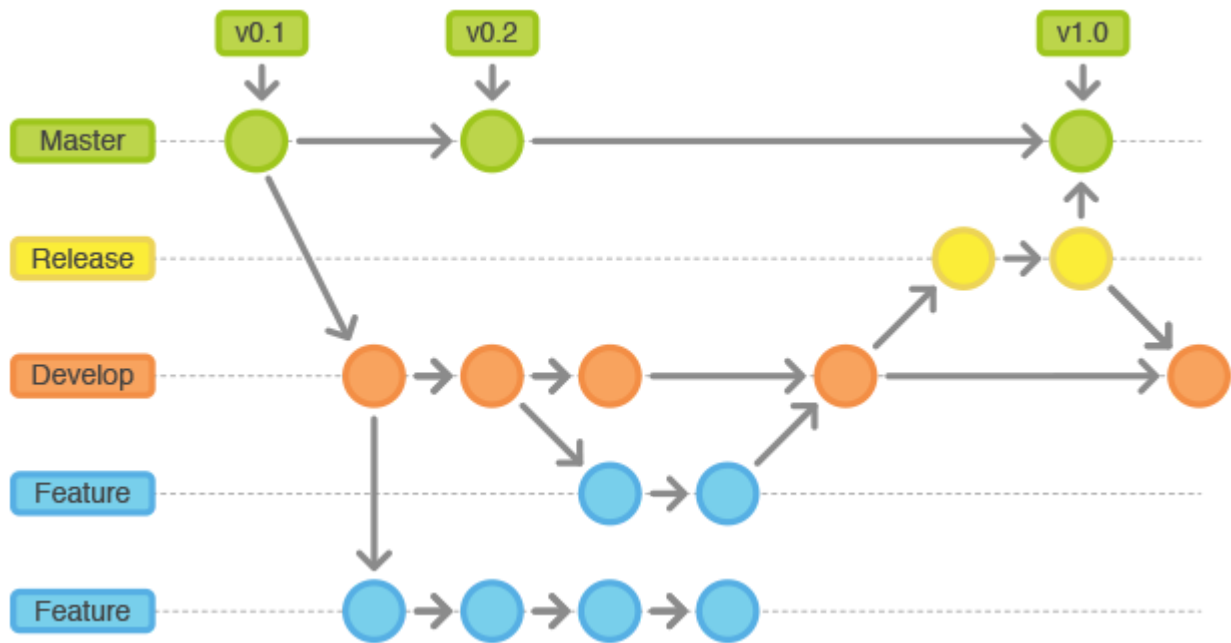
git checkout -b <nome-do-branch> "cria o branch e já utilizá-lo"

git branch -d <nome-do-branch> "apaga o branch"

git checkout <nome-do-branch> "muda para o branch"
```

UM PEQUENO EXEMPLO DO FLUXO DE BRANCH QUE PODE SER USADO POR UMA EQUIPE DE DEVS

O FLUXO ABAIXO PODERIA SER NO DESENVOLVIMENTO DE UM APP DE SMARTPHONE



No caso o produto que será enviado para o cliente esta no branch master

Lembrando que esse fluxo não e obrigatório, contudo facilitar o desenvolvimento de trabalho em equipes

Git merge

Usado para juntar fluxo de trabalho de outros branch como também de outros desenvolvedores.

Atenção geralmente nesse ponto que aparece os conflitos de códigos então tome cuidado na hora do merge!!!

```
git merge <nome-do-branch>
```

lembrando que o merge é sempre feito do branch atual

Exemplo: As alterações feitas no branch **feature** deve ser unida no branch **master** para isso devemos mudar para o branch **master**

```
git checkout master
```

 e logo em seguida usar

```
git marge feature
```

 com esse comando as novas mudanças feitas no branch **feature** iram para o **master**

Cuidado!!!

***se o passo anterior for executado ao contrário ***

```
git checkout feature
```

 mudando para o branch feature e logo em seguida fazendo o merge

`git merge master` as alterações do master ficaram no feature mas as novas modificações realizadas no feature não pertenceram ao master

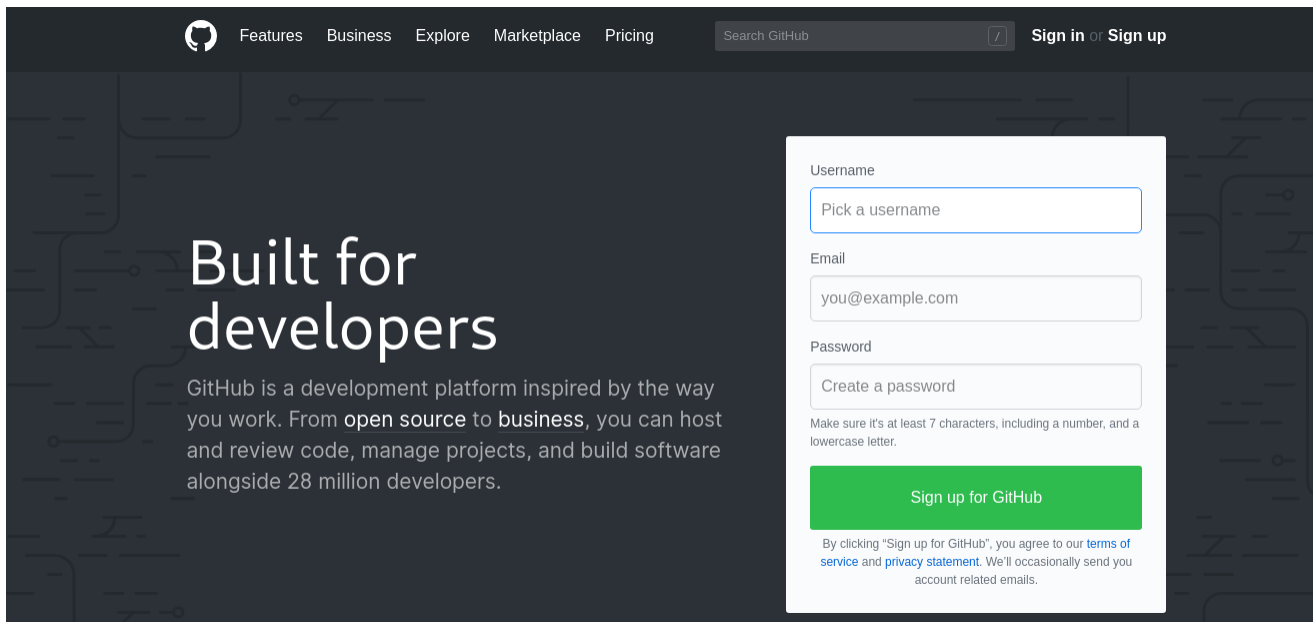
Essa abordagem deve ser evitada!

Deve ser definida pela equipe de qual branch terá todas as alterações e ser disponibilizado para o usuário final

No caso o branch MASTER TERÁ TODAS AS IMPLEMENTAÇÕES SENDO O BRANCH MAIS ESTÁVEL DO DESENVOLVIMENTO

Criando um Repositório Remoto

Acesse o site www.github.com



The screenshot shows the GitHub homepage with a dark background and a white sign-up form overlay on the right. The form includes fields for Username, Email, and Password, along with a 'Sign up for GitHub' button. The background text on the left reads 'Built for developers' and 'GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 28 million developers.'

Features Business Explore Marketplace Pricing Search GitHub Sign in or Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 28 million developers.

Username
Pick a username

Email
you@example.com

Password
Create a password

Make sure it's at least 7 characters, including a number, and a lowercase letter.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Faça o login



Sign in to GitHub

Username or email address

wagnerrodrigo.pan@gmail.com

Password

[Forgot password?](#)

.....

Sign in

New to GitHub? [Create an account.](#)

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

Clique no ícone do mais e logo após em New repository

The screenshot shows the GitHub homepage. At the top, there's a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the right side of the navigation bar, there's a user profile icon and a plus icon. A dropdown menu is open from the plus icon, showing options: New repository (highlighted), Import repository, New gist, and New organization. The main content area has a light blue background with the text 'Learn Git and GitHub without any code!' and a subtext 'Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.' Below this are two buttons: 'Read the guide' (green) and 'Start a project' (grey). At the bottom, there are several notifications: a 'GitHub Universe' event announcement, a 'Browse activity' section showing a star on 'PaulRosset/previs', and a 'Discover repositories' link. There are also banners for 'Our new Terms of Service and Privacy Statement are in effect.' and a 'Star' button for the repository shown.

Coloque um nome para o repositório

a descrição e opcional como o arquivo de Readme


O repositório o private é necessário add um cartão de credito

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner


Repository name

 **wagnerrodrigo** / **PalestraGit** ✓

Great repository names are short and memorable. Need inspiration? How about **scaling-spoon**.

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

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

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Agora basta sincronizar o repositório remoto com o repositório local \o/

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

or

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# PalestraGit" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/wagnerrodrigo/PalestraGit.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/wagnerrodrigo/PalestraGit.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Git push sincronizar o repositório local com o remoto

Até agora você fez tudo o trabalho localmente "no seu computador" chegou a hora de sincronizar os repositórios com o comando

```
git remote add origin <nome-do-repositório-REMOTO>
git push -u origin master
```

Atenção O comando acima é digitado apenas uma vez depois o comando utilizado será o

```
git push
```

Git pull sincronizar o repositório remoto com o local

Em um aula na faculdade você fez algumas alterações no exercício que enviou para seu repositório remoto

```
git push lab-Prog
```

E chegando em casa você fazendo o processo inverso do git push, você sincroniza com o seu repositório remoto com o seu repositório local com o comando

```
git pull "atualiza o repositório local com base no repositório remoto"
```

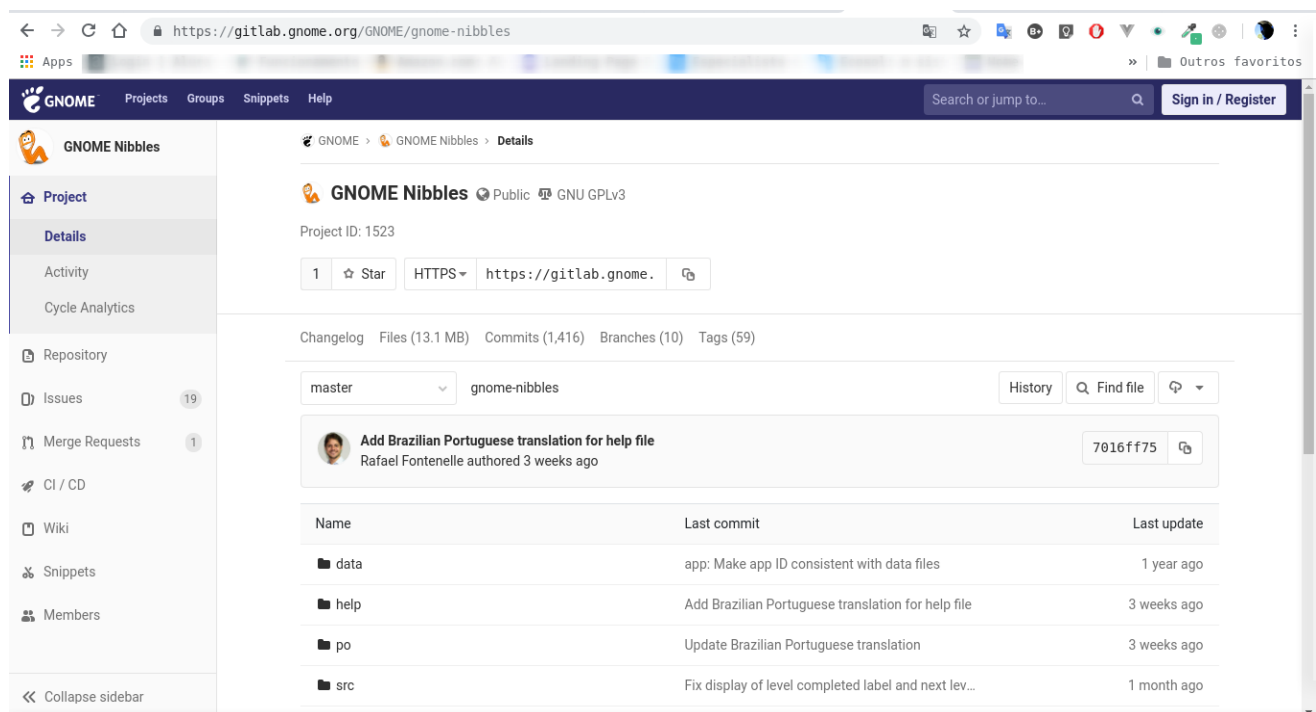
Fazendo o Download de repositórios

Você pode baixar o código de qualquer pessoa que esteja no github ou em outro serviço de armazenagem tendo o link do seu repositório com o comando

```
git clone <link-do-repositório>
```

Um exemplo

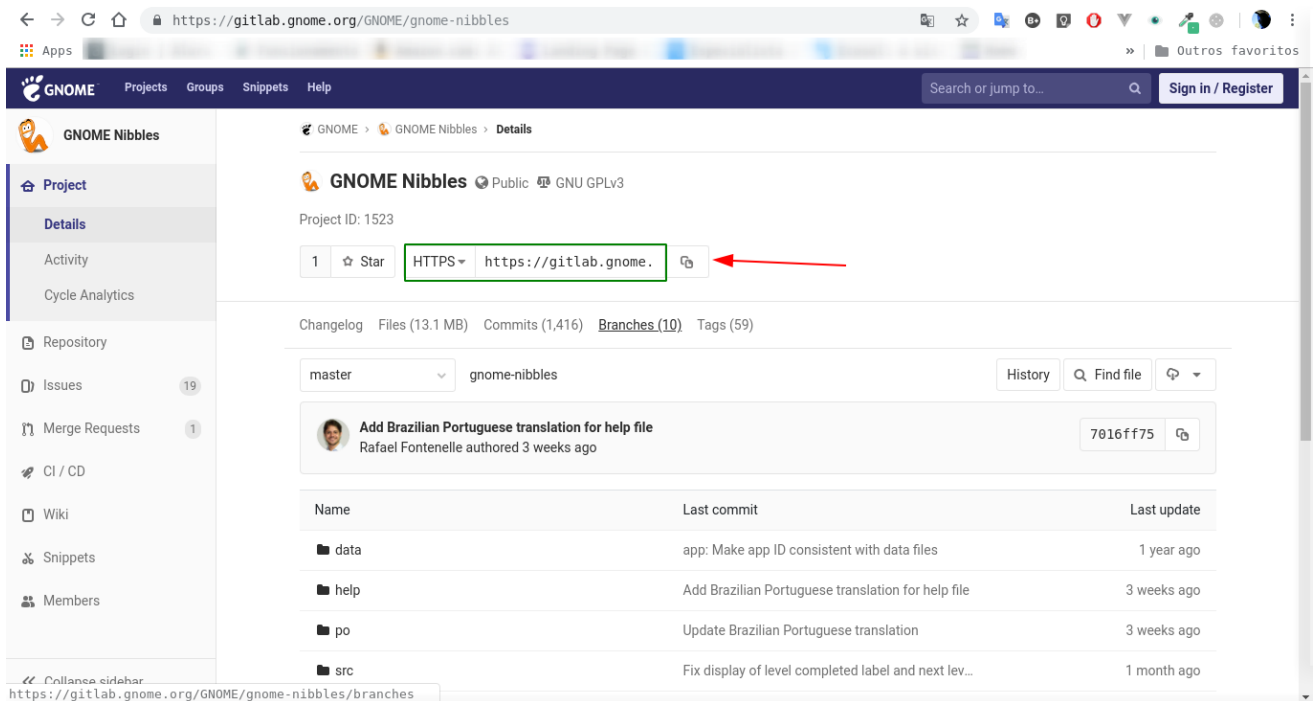
Digamos que você queira baixar um game da minhoca Gnome Nibbles acesse o site <https://gitlab.gnome.org/GNOME/gnome-nibbles>



The screenshot shows the GitLab interface for the GNOME Nibbles repository. The left sidebar contains navigation links: Project (Details, Activity, Cycle Analytics), Repository (Issues: 19, Merge Requests: 1, CI / CD, Wiki, Snippets, Members), and a Collapse sidebar button. The main content area displays the repository details for GNOME Nibbles, including the Project ID (1523), a star button, and the HTTPS URL (https://gitlab.gnome.org). Below this, there are links for Changelog, Files (13.1 MB), Commits (1,416), Branches (10), and Tags (59). A commit history table is shown with columns for Name, Last commit, and Last update. The table lists several commits, including one for adding a Brazilian Portuguese translation for the help file.

Name	Last commit	Last update
data	app: Make app ID consistent with data files	1 year ago
help	Add Brazilian Portuguese translation for help file	3 weeks ago
po	Update Brazilian Portuguese translation	3 weeks ago
src	Fix display of level completed label and next lev...	1 month ago

Clique no botão ou copie o link



escolha o diretório no seu computador

```
git clone https://gitlab.gnome.org/GNOME/gnome-nibbles.git
```

Com esse comando o git ira fazer download do repositório com os arquivo e o histórico do código.

Pull Request GitHub

Você tem uma conta no Github, acompanha projetos open source, gostaria de colaborar com algum deles, mas não sabe como fazer?

a resposta é simples faça um pull request!

O **pull request** é usado para sugerir melhorias, correções, traduções , etc em software ou projetos open source por exemplo o pedido de um aluno para adicionar o email da instituição em uma lista de licença de gratuita para estudantes de um empresa Software <https://github.com/JetBrains/swot/pull/5279>

e a correção do dominio <https://github.com/JetBrains/swot/pull/5364>

JetBrains / **swot**
forked from leereilly/swot

Watch 38 Star 503 Fork 6,021

Code Pull requests 28 Projects 0 Insights

change of domain #5364

Merged philipto merged 1 commit into JetBrains:master from wagnerrodrigo:master 17 days ago

Conversation 1 Commits 1 Checks 0 Files changed 1

wagnerrodrigo commented 22 days ago

Change in the institution's domain for `cesjf.txt` : from BR/EDU to BR.
Reference: <https://www.cesjf.br/>

change of domain b4db016

philipto merged commit bdee941 into JetBrains:master 17 days ago

philipto commented 17 days ago

@wagnerrodrigo Pull request merged. Thank you!

Reviewers
No reviews

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

para fazer o pull request precisamos fazer alguns passos

1. Acesse a pagina do github escolha o projeto clique no botão de Fork.
2. Com o Clone dos código em sua maquina modifique os arquivo.
3. Envie novamente para o seu *Repositório Remoto*.
4. Finalmente o Pull Request
5. Agora é só esperar

O Fork

GitHub, Inc. [US] | <https://github.com/JetBrains/swot>

Apps Pull requests Issues Marketplace Explore

JetBrains / **swot**
forked from leereilly/swot

Watch 38 Star 503 Fork 6,021

Code Pull requests 28 Projects 0 Insights

Identify email addresses or domains names that belong to colleges or universities. Help automate the process of approving or rejecting academic discounts.

10,154 commits 9 branches 22 releases 3,305 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

This branch is 9631 commits ahead, 1096 commits behind leereilly:master. #5441 Compare

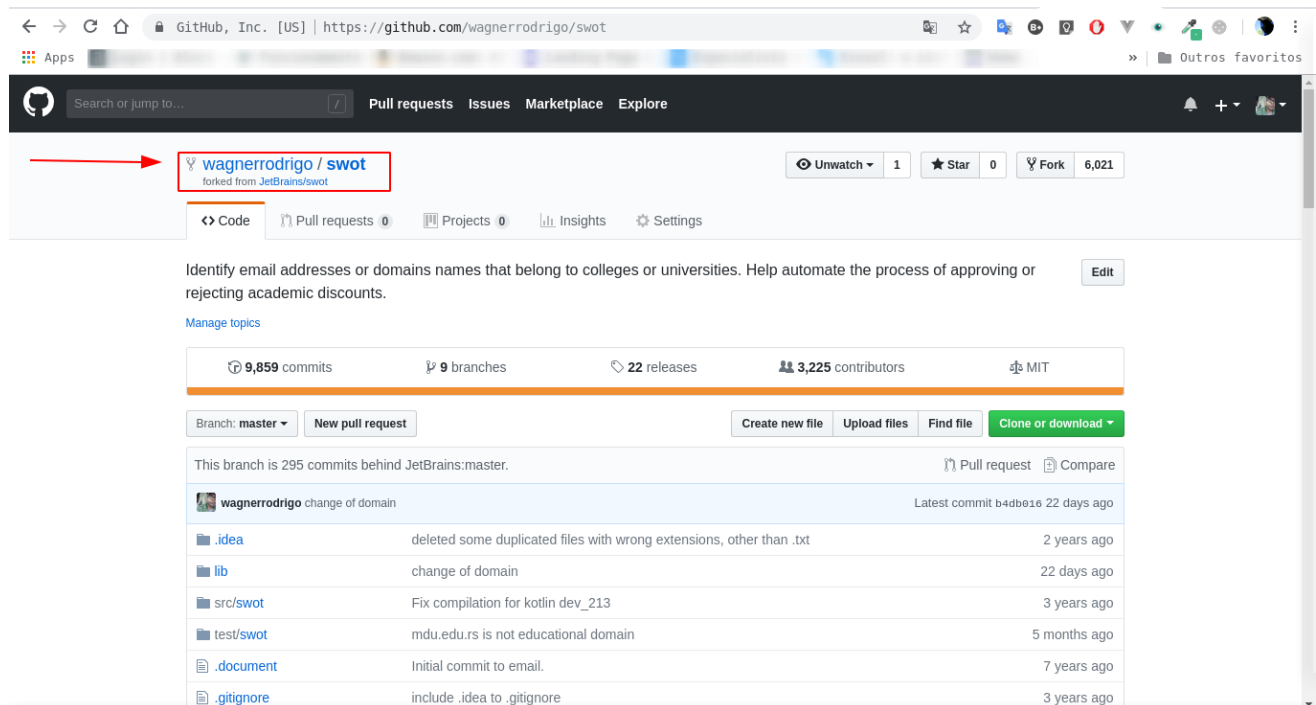
philipto Merge pull request #4943 from jeremy886/patch-2 Latest commit fe326c7 20 hours ago

.idea	deleted some duplicated files with wrong extensions, other than .txt	2 years ago
lib	Merge pull request #4943 from jeremy886/patch-2	20 hours ago
src/swot	Fix compilation for kotlin dev_213	3 years ago
test/swot	mdu.edu.rs is not educational domain	5 months ago
.document	Initial commit to email.	7 years ago
.gitignore	include .idea to .gitignore	3 years ago

<https://github.com/JetBrains/swot/pulse> ite CONTRIBUTING.md a month ago

<https://github.com/letBrains/swot>

Com isso será feita uma copia do repositório da empresa na sua conta do github

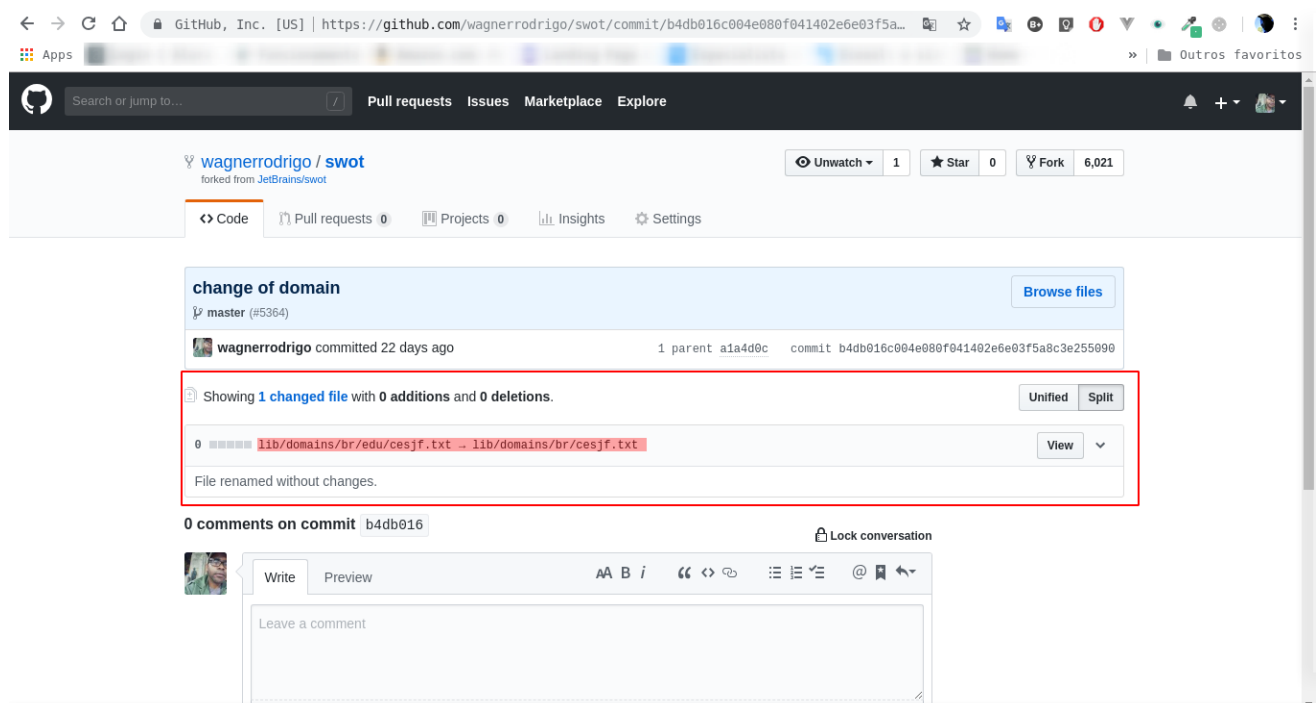


The screenshot shows the GitHub repository page for 'wagnerrodrigo / swot', which is forked from 'JetBrains/swot'. A red arrow points to the repository name. The page displays statistics: 9,859 commits, 9 branches, 22 releases, and 3,225 contributors. Below the statistics, there is a table of recent commits. The commit history table is as follows:

Commit	Message	Time
wagnerrodrigo	change of domain	Latest commit b4db016 22 days ago
	.idea deleted some duplicated files with wrong extensions, other than .txt	2 years ago
	lib change of domain	22 days ago
	src/swot Fix compilation for kotlin dev_213	3 years ago
	test/swot mdu.edu.rs is not educational domain	5 months ago
	.document Initial commit to email.	7 years ago
	.gitignore include .idea to .gitignore	3 years ago

<https://github.com/wagnerrodrigo/swot>

Com o codigo na usa maquina faça a modificação e envie novamente para o github

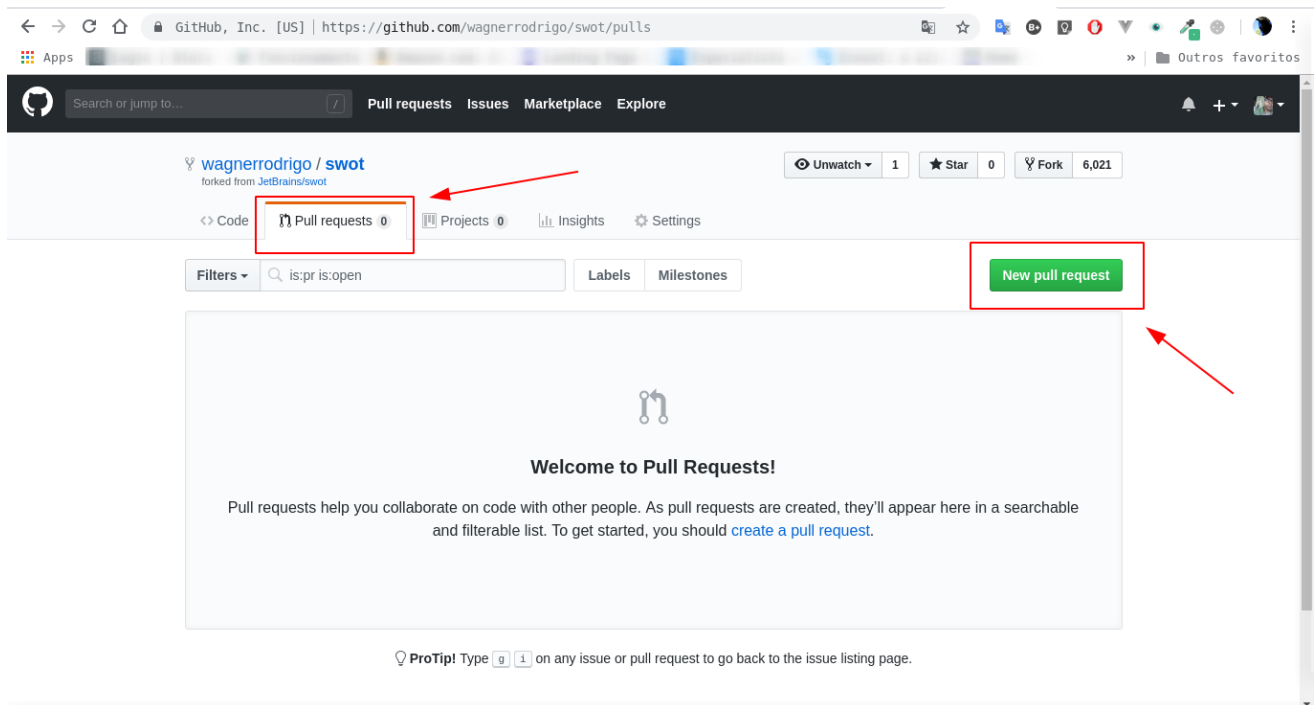


The screenshot shows the GitHub commit page for the 'change of domain' commit. A red box highlights the commit details, including the commit message 'change of domain', the commit hash 'b4db016', and the commit message '1 changed file with 0 additions and 0 deletions'. Below the commit details, there is a table of changed files. The table is as follows:

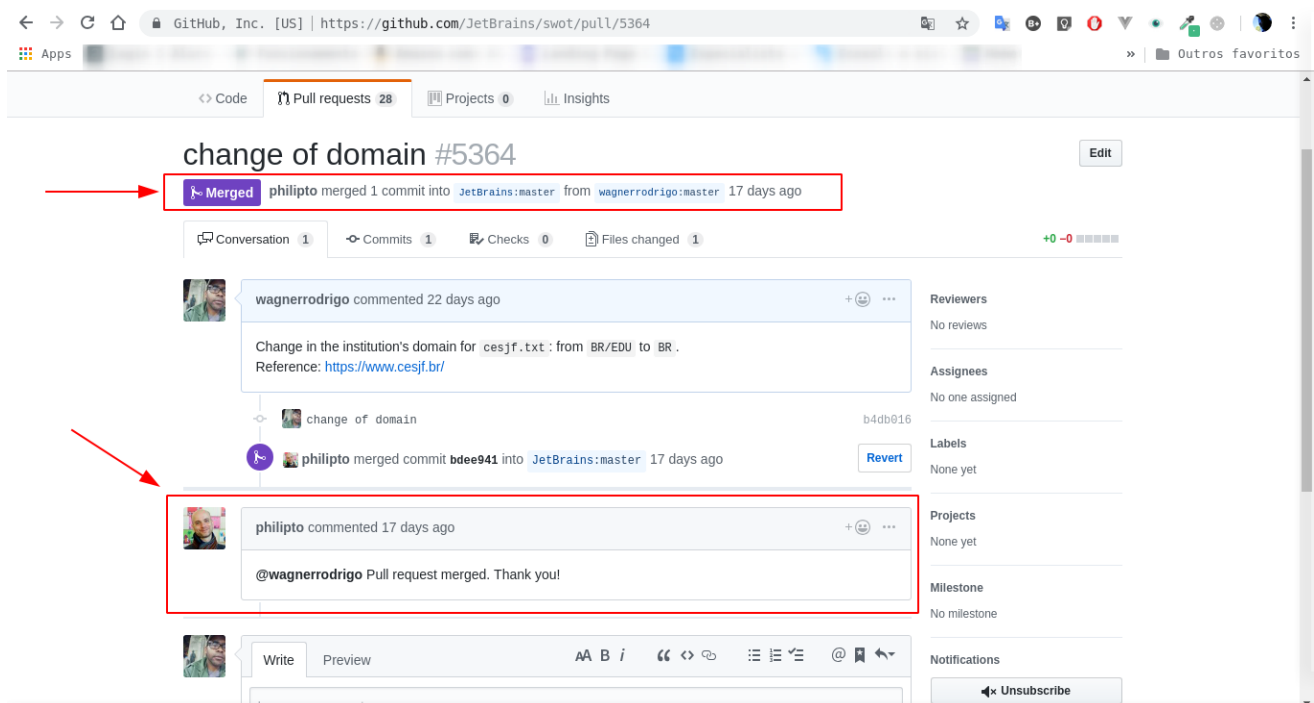
File	Changes
lib/domains/br/edu/cesjf.txt	11b/domains/br/edu/cesjf.txt

File renamed without changes.

Acesse a aba de pull request logo depois clique no botão NEW PULL REQUEST



Agora é só esperar a resposta do seu pedido



Links / Referências

Treino dos comandos

- <https://try.github.io/> iniciantes na linha de comando do git legal \O/
- <https://explainshell.com/> site que explica a linha de comando muito bom \O/

Git do kernel linux

- <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git> O que sera que Linus torvalds está fazendo hoje?
- <https://kernelnewbies.org/> comunidade de aspirantes do kernel linux
- <http://eudypula-challenge.org/> O Desafio de programação do kernel Linux

Serviços de Armazenamento de código

- <https://github.com/> GitHub
- <https://bitbucket.org/> BitBucket
- <https://about.gitlab.com/> GitLAB

Documentação

- <https://git-scm.com/> Documentação oficial
- <https://git-scm.com/book/pt-br/v2> Documentação em Português
- <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud> Bit bucket
 - <https://www.atlassian.com/git/tutorials/comparing-workflows>
- <https://guides.github.com/introduction/git-handbook/> Git handbook

Material

- <https://github.com/sindresorhus/awesome> Um conceito bem legal do github são as AWESOME LIST
- <https://github.com/ellisonleao/magictools#assetsplaceholders> AWESOME GAME!
- <http://cerebrobr.github.io/cerebro/> O Cerebro
- <https://github.com/JanVanRyswyck/awesome-talks#readme> Awesome talks
- <https://github.com/aleksandar-todorovic/awesome-linux> Awesome Linux
- <https://github.com/luongvo209/Awesome-Linux-Software> Awesome Linux Software

Workflow fluxo de trabalho

- https://danielkummer.github.io/git-flow-cheatsheet/index.pt_BR.html esse é o mais conhecido criado pelo daniel kummer
- <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow> tutorial da Atlassian

Ferramentas

- <https://gitforwindows.org/> **GitBash**
- <https://www.gitkraken.com/> **GitKraken**
- <https://www.sourcetreeapp.com/> **Source tree**
- <https://www.git-tower.com/mac> **Tower**
- <https://git-scm.com/download/gui/linux> **lista mais detalhada de ferramentas grafica do git**
 - <https://git-scm.com/download/gui/windows>
 - <https://git-scm.com/download/gui/mac>

Controle de versão

- <http://sccs.sourceforge.net/> ***Source Code Control System (SCCS)**
- https://www.gnu.org/software/rcs/?utm_source=blog&utm_campaign=rc_blogpost **Revisin Control System**
- https://subversion.apache.org/?utm_source=blog&utm_campaign=rc_blogpost **Apache Subversion**
- <https://www.perforce.com/solutions/game-development> **Perforce GAME!!!**
- <https://www.mercurial-scm.org/> **Mercurial**
- <https://www.bitkeeper.org/> **!!!! BitKeeper !!!!!**

links

- <https://smartmockups.com/> **Smart mockup**
- <https://fontawesome.com/> **Fontes**
- <https://code.jquery.com/jquery/> **Jquery**
- <https://www.pexels.com/> **download de imagens**

Cursos

- <https://www.udemy.com/git-e-github-ninja/> **Curso na udmy**
- <https://www.youtube.com/watch?v=p1VAghNq-qg> **Youtube gitFlow**

Videos

- <https://www.youtube.com/watch?v=w3jLIU7DT5E&t=6s> **Youtube What is GitHub?**
- <https://www.youtube.com/watch?v=afvT1c1ii0c&t=50s> **Youtube How we build software.**