



INSTITUTO FEDERAL

PARÁ
Campus Paragominas

Controle de Versão com Git

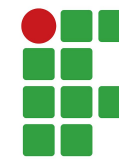
Me. Bruno Gutierrez Ratto Clemente



Git

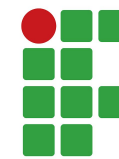
- ▷ Inicialmente projetado por Linus Torvalds para o desenvolvimento do Kernel Linux, para trabalhar em paralelo com voluntários sem a limitação geográfica.





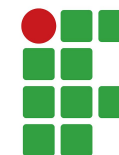
Git

- ▷ O Git é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, embora possa ser usado para qualquer tipo de arquivo.
 - Você pode escrever código de um programa de computador,
 - Você pode escrever um livro,
 - Você pode escrever artigos,
 - Entre outros...



Git

- ▷ É uma ferramenta de código aberto e software livre:
 - termo que se refere a um software cujo código esta disponível para download para qualquer pessoa e a filosofia de criação de aplicativos voltada para a colaboração de desenvolvedores.
 - Para ser Open Source, não basta apenas acesso ao código fonte. É importante que se siga alguns critérios que podem ser encontrados em opensource.org



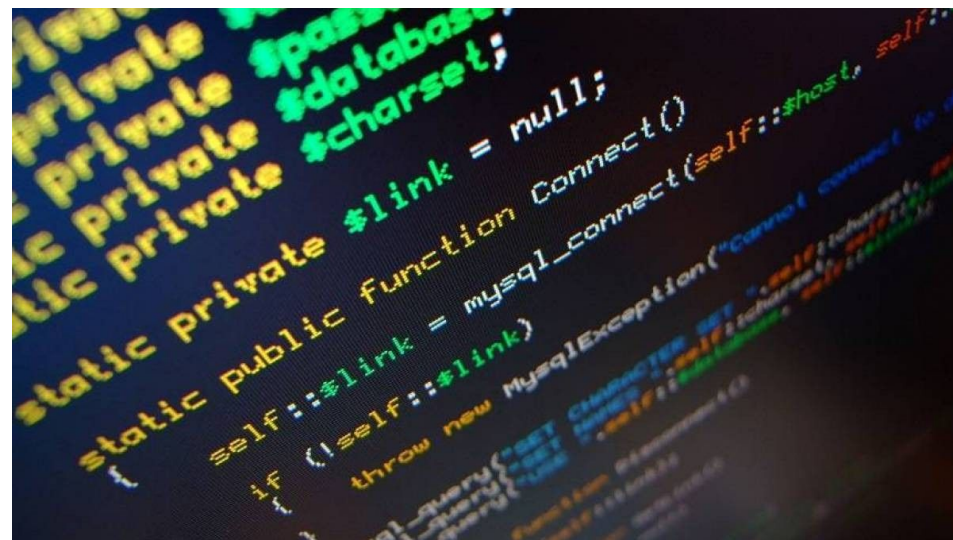
Git

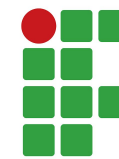
- ▷ Um programa Open Source deve garantir distribuição livre.
 - A licença não deve restringir qualquer parte de vender ou distribuir o software e a licença não pode exigir taxas de royalties ou quaisquer outras taxas para vendas.
- ▷ Sobre trabalhos derivados:
 - A licença deve permitir modificações e trabalhos derivados.
 - Além disto, estes novos trabalhos devem ser distribuídos sobre os mesmos termos da licença original.



Git

- ▷ Sobre o código fonte, um programa Open Source deve:
 - O programa deve incluir seu código fonte e deve permitir sua distribuição também na forma compilada.

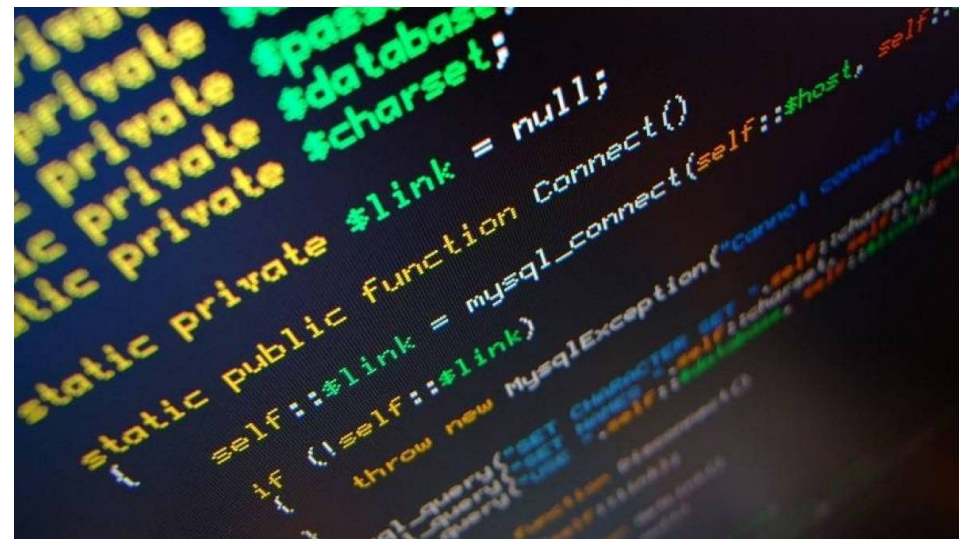




Git

- ▶ Além disto, Se o programa não for distribuído com seu código fonte, então
- ▶ deve haver algum meio de obtê-lo;
 - Inclusive por um repositório com Git !

▶ O código deve ser legível e inteligível por qualquer programador.

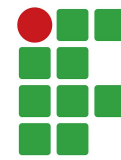




Antes de continuarmos...

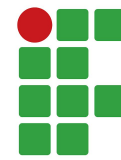
- ▷ Porque usar o Git?





Git

- ▷ Git é um software livre e alguns sites provêm hospedagem gratuita de código fonte para repositório Git:
 - GitHub
 - Gitorious
 - Sourceforge
 - GNU Savannah
 - Project Kenai
 - SourceRepo
 - Google Code
 - Bitbucket



GitHub

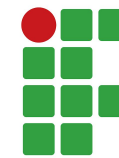
- ▷ GitHub é uma dessas várias plataformas que hospedam código para controle de versão e colaboração.
- ▷ O GitHub é ainda mais: é uma espécie de rede social (ou fábrica social de software) que promove milhares de códigos fonte para as mais diversas finalidades

Outra alternativa: BitBucket



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ O BitBucket funciona de forma similar ao GitHub, porém oferta a possibilidade de se utilizar repositórios privados para equipes de até 5 membros.
 - Dessa forma, costuma ser uma tendência para desenvolvedores que não pretendem expôr seu código de forma pública, mas pretendem fazer uso de seu controle de versão robusto e suas ferramentas de gerenciamento de configuração de software.



Conceitos importantes

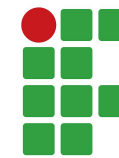
- ▷ **Iniciais:**
 - Init
 - Remote add
- ▷ **Status**
- ▷ **Push**
- ▷ **Pull**
- ▷ **Commit**
- ▷ **Checkout**
- ▷ **Branch**
- ▷ **Merge**
- ▷ **Revert**



Conceitos importantes

- ▷ **Git init**
 - Inicia uma pasta do seu computador como um repositório a ser gerenciado pelo git

- ▷ **Git remote add origin <caminho>**
 - Cria-se um vínculo entre a pasta do computador e um repositório remoto



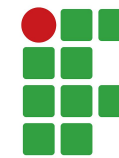
Conceitos importantes

▷ Pull

- Do inglês puxar
 - Atualiza o repositório local com base no seu repositório remoto
 - “Puxa” para o seu computador os dados do servidor

▷ Push

- Do inglês empurrar
 - Atualiza o repositório remoto com base no seu repositório local
 - “Empurra” os dados do seu computador para o servidor



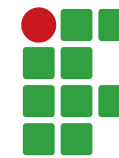
Conceitos importantes

▷ Commit

- Confirma uma determinada transação, de forma definitiva

▷ Checkout

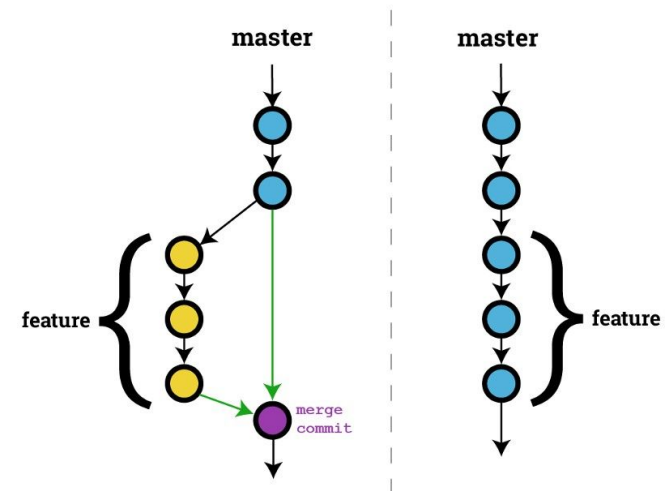
- Aponta seu repositório local para um determinado “momento” do repositório remoto

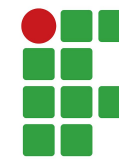


Conceitos importantes

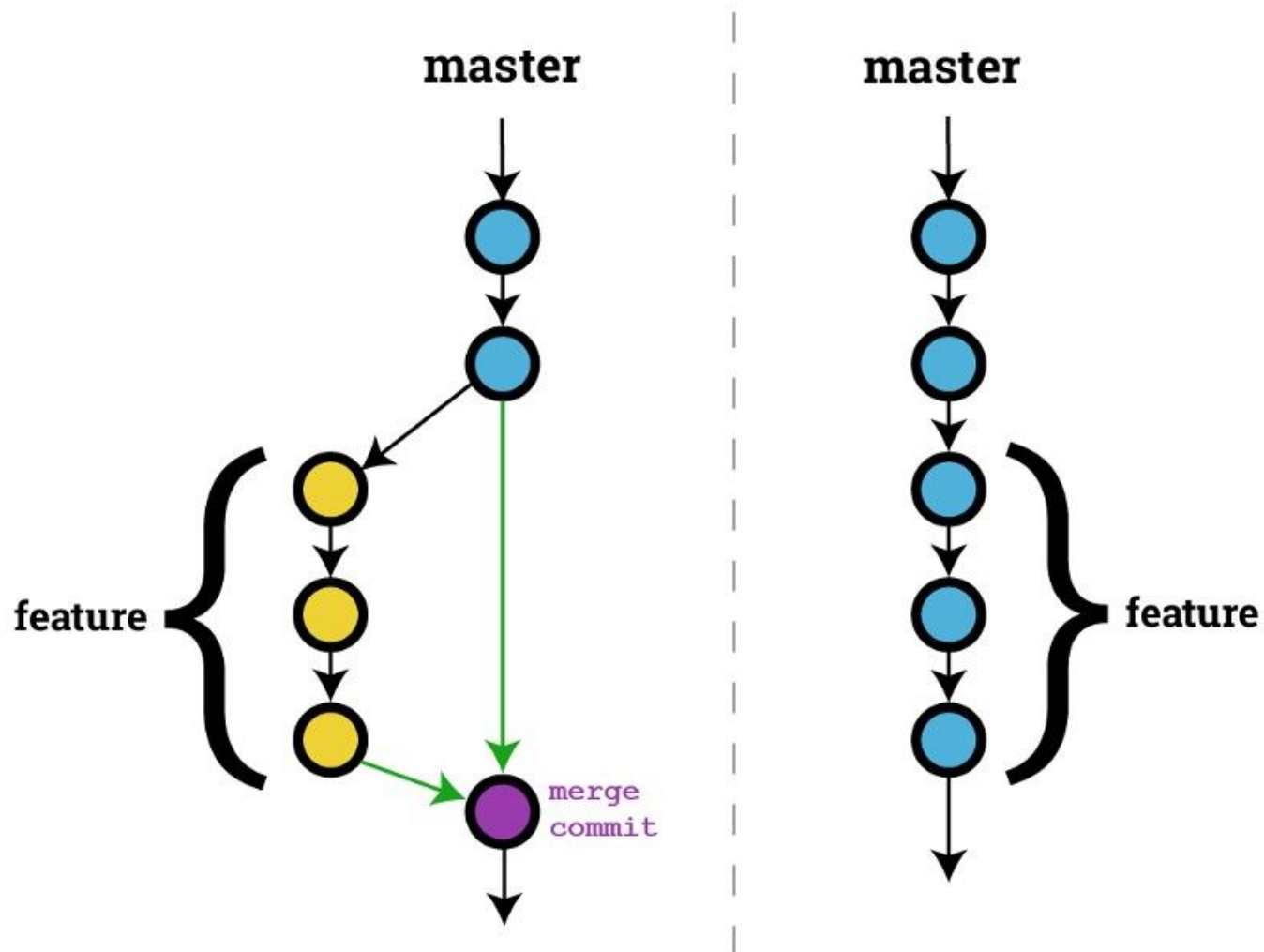
▷ Branch

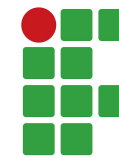
- Do inglês “galho”
 - Cria uma “ramificação” a partir da árvore principal, para se desenvolver sem afetar o código principal





Conceitos importantes

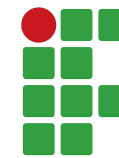




Conceitos importantes

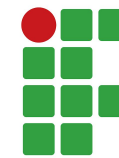
▷ Merge

- “Fusão” entre dois branches, ou seja, reunir 2 implementações diferentes, de forma a ter apenas uma, definitiva



Conceitos importantes





Conceitos importantes

▷ Revert

- Retorna o programa para um determinado momento (branch).
 - Gera um novo commit, apagando o que foi feito no commit revertido.
 - Lembrando: Commit é algo DEFINITIVO.

▷ Issues

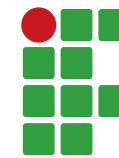
- Tarefas a serem feitas.



Antes de começarmos

- ▷ É essencial fazer o download do Git
 - <https://git-scm.com/downloads>
- ▷ Caso você utilize linux, você já consegue baixar direto pelo comando abaixo:
 - Sudo apt install git
- ▷ Após a instalação bem sucedida do Git, podemos começar a usar o GitHub

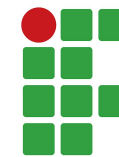
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Antes de tudo, é necessário criar uma conta na plataforma
- ▷ Logo em seguida, você tem acesso à principal interface do site, com um Overview de seus repositórios, equipes e projetos

Começando com o GitHub



INSTITUTO FEDERAL

PARÁ
Campus Paragominas

The screenshot shows the GitHub homepage. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the left, the 'Repositories' section lists several repositories by user 'bruno88'. The main content area features a large card titled 'Discover interesting projects and people to populate your personal news feed.' with an 'Explore GitHub' button. On the right, there is a 'Securing software together' notification and an 'Explore repositories' section highlighting the 'quasarframework/quasar' project.

Repositories [New](#)

Find a repository...

- [bruno88/IFPA_PADM_3B_2019](#)
- [bruno88/IFPA_PADM_3A_2019](#)
- [bruno88/IFES_P2_2018_02](#)
- [bruno88/IFES_LP1_2018_02](#)
- [bruno88/asdasdasd](#)
- [bruno88/IFES_PS_2018_02](#)

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you [watch](#) and people you [follow](#).

[Explore GitHub](#)

Securing software together ×

Introducing new ways to identify and prevent security vulnerabilities across your code base.

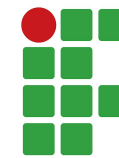
Explore repositories

quasarframework/quasar

Quasar Framework - Build high-performance VueJS user interfaces in record time

JavaScript ★ 12.4k

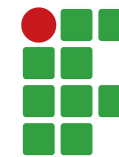
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▶ Agora é necessário criar um Repositório: um local no servidor do GitHub onde ficará nosso projeto

Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

Create a new repository

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

Owner



bruno88 ▾

Repository name *

/ livroGIT_palestra



Great repository names are short and memorable. Need inspiration? How about **redesigned-dollop**?

Description (optional)

Livro sobre Git que servirá como exemplo para a Palestra de uso do Git.



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

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

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

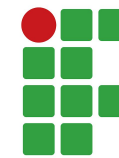
Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Agora é o projeto será exibido imediatamente na nossa tela

Começando com o GitHub



INSTITUTO FEDERAL

PARÁ
Campus Paragominas

bruno88 / livroGIT_palestra

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Livro sobre Git que servirá como exemplo para a Palestra de uso do Git.

Edit

[Manage topics](#)

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

bruno88 Initial commit

Latest commit f0ade70 now

README.md

Initial commit

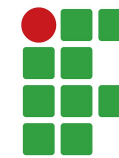
now

README.md

livroGIT_palestra

Livro sobre Git que servirá como exemplo para a Palestra de uso do Git.

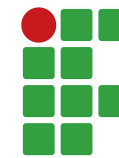
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Ao criar seu repositório, ele passa a existir “remotamente”, ou seja, apenas no servidor do GitHub, porém você pode fazer uma “ligação” entre o servidor remoto e uma pasta do seu computador.
 - Vamos criar uma pasta no computador e abrir o Git lá:

Começando com o GitHub

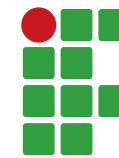


INSTITUTO FEDERAL

PARÁ
Campus Paragominas

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git init  
Initialized empty Git repository in /home/bruno/Documentos/Trabalho/repositorios/  
livroGIT_palestra/.git/  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git remote add origin https://github.com/bruno88/livroGIT_palestra.git  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```

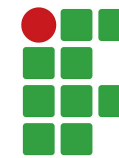
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Sempre que for começar a mexer em seu projeto, é importante usar o comando abaixo:
 - `Git pull origin master`
- ▷ O “pull”, puxa os dados do repositório remoto, atualizando o Local.
 - Isso é importante caso você tenha outras pessoas usando esse mesmo repositório, para evitar que seu repositório local não fique desatualizado.

Começando com o GitHub

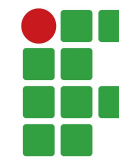


INSTITUTO FEDERAL

PARÁ
Campus Paragominas

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git pull origin master  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
From https://github.com/bruno88/livroGIT_palestra  
* branch          master      -> FETCH_HEAD  
* [new branch]     master      -> origin/master  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```

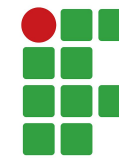
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas



Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Outro comando muito utilizado é o:
 - Git status
- ▷ Seu objetivo é identificar qual o status atual do repositório.
 - Aqui você verá:
 - se arquivos foram alterados
 - se há conflitos nos arquivos existentes
 - se foram adicionados ou removidos arquivos
 - se há algo pendente
 - se há arquivos não sendo acompanhados pelo Git
 - (...)

Começando com o GitHub

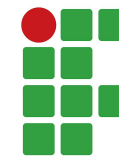


INSTITUTO FEDERAL

PARÁ
Campus Paragominas

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git status  
No ramo master  
nothing to commit, working tree clean  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```

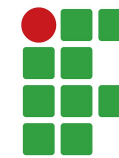
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Ao alterar o repositório, ele identificará que o repositório local está diferente do remoto, indicando conflito.
- ▷ Tente criar um novo arquivo e digitar “git status”, por exemplo:
 - teste.html

Começando com o GitHub

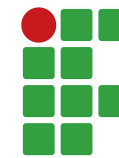


INSTITUTO FEDERAL

PARÁ
Campus Paragominas

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git status  
No ramo master  
Arquivos não monitorados:  
  (utilize "git add <arquivo>..." para incluir o que será submetido)  
  
  teste.html  
  
nada adicionado ao envio mas arquivos não registrados estão presentes (use "git  
add" to registrar)  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```

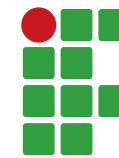
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Observe que o git indicará que há alterações, mas que elas não estão sendo observadas.
- ▷ O Git apenas acompanha arquivos que você indicou que devem ser acompanhados
- ▷ Para isso:
 - `Git add teste.html`
 - `Git add .`
 - O “ponto” indica que todos os arquivos devem ser observados.

Começando com o GitHub

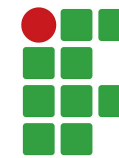


INSTITUTO FEDERAL

PARÁ
Campus Paragominas

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git add .  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git status  
No ramo master  
Mudanças a serem submetidas:  
  (use "git reset HEAD <file>..." to unstage)  
  
    new file:   teste.html  
  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```

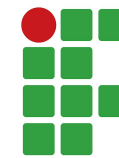
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Agora que você fez uma modificação e indicou que ela deve ser observada, o Git indicará (com o uso do `Git status`) que há uma diferença entre os repositórios remoto e local
 - O seu (local) está “mais avançado” que o remoto.

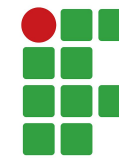
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Para enviar devemos usar o comando abaixo:
 - `Git commit -a -m "comentário"`
 - `-a`: All, ou seja, enviar todos os arquivos
 - `-m`: Message, ou seja, permite que você digite uma mensagem no commit
 - Use mensagens como:
 - Adicionado o arquivo teste.html
 - Adicionada botão de "mostrar senha"
 - Ajuste na máscara do campo CPF
 - Entre outros...
 - Isso garantirá Rastreabilidade !

Começando com o GitHub

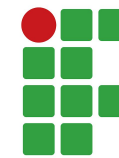


INSTITUTO FEDERAL

PARÁ
Campus Paragominas

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git commit -a -m "Enviando o arquivo inicial do projeto"  
[master 2dbf5df] Enviando o arquivo inicial do projeto  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 teste.html  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```

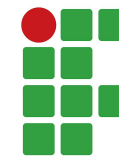
Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ Embora você tenha feito o commit, ainda se encontra apenas no repositório local, portanto é necessário enviá-lo para o repositório remoto, com o comando abaixo:
 - `git push origin master`
- ▷ Isso fará um envio para o repositório origin, branch master (principal por padrão)

Começando com o GitHub

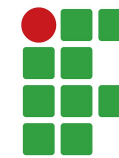


INSTITUTO FEDERAL

PARÁ
Campus Paragominas

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
git push origin master
Username for 'https://github.com': brunoclemente88@gmail.com
Password for 'https://brunoclemente88@gmail.com@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 300.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/bruno88/livroGIT_palestra.git
    f0ade70..2dbf5df  master -> master
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```

Começando com o GitHub



INSTITUTO FEDERAL
PARÁ
Campus Paragominas

- ▷ A partir desse momento, você pode acompanhar as alterações no site do GitHub, conferindo nas páginas Commit, Branches, Issues, etc...

Começando com o GitHub



INSTITUTO FEDERAL

PARÁ
Campus Paragominas

bruno88 / livroGIT_palestra

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Livro sobre Git que servirá como exemplo para a Palestra de uso do Git.

Edit

[Manage topics](#)

2 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

bruno88 Enviando o arquivo inicial do projeto

Latest commit 2dbf5df 2 minutes ago

README.md

Initial commit

12 minutes ago

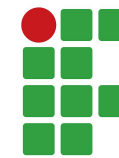
teste.html

Enviando o arquivo inicial do projeto

2 minutes ago

README.md





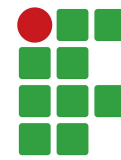
GitHub

- ▷ Tudo isso é um cenário perfeito:
 - Apenas 1 membro na equipe;
 - Apenas 1 branch;
 - Apenas 1 versão de código;
 - Apenas 1 Issue a ser resolvida;
 - Nenhum conflito entre commits;
- ▷ A realidade não é tão fácil assim, portanto temos os Branches



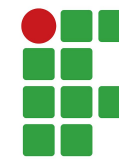
Branches

- ▷ Cada vez que você faz um commit, o Git armazena uma Hash do seu commit, criando uma Snapshot
 - ou uma Configuração de Software
 - Representa como o sistema está nesse momento

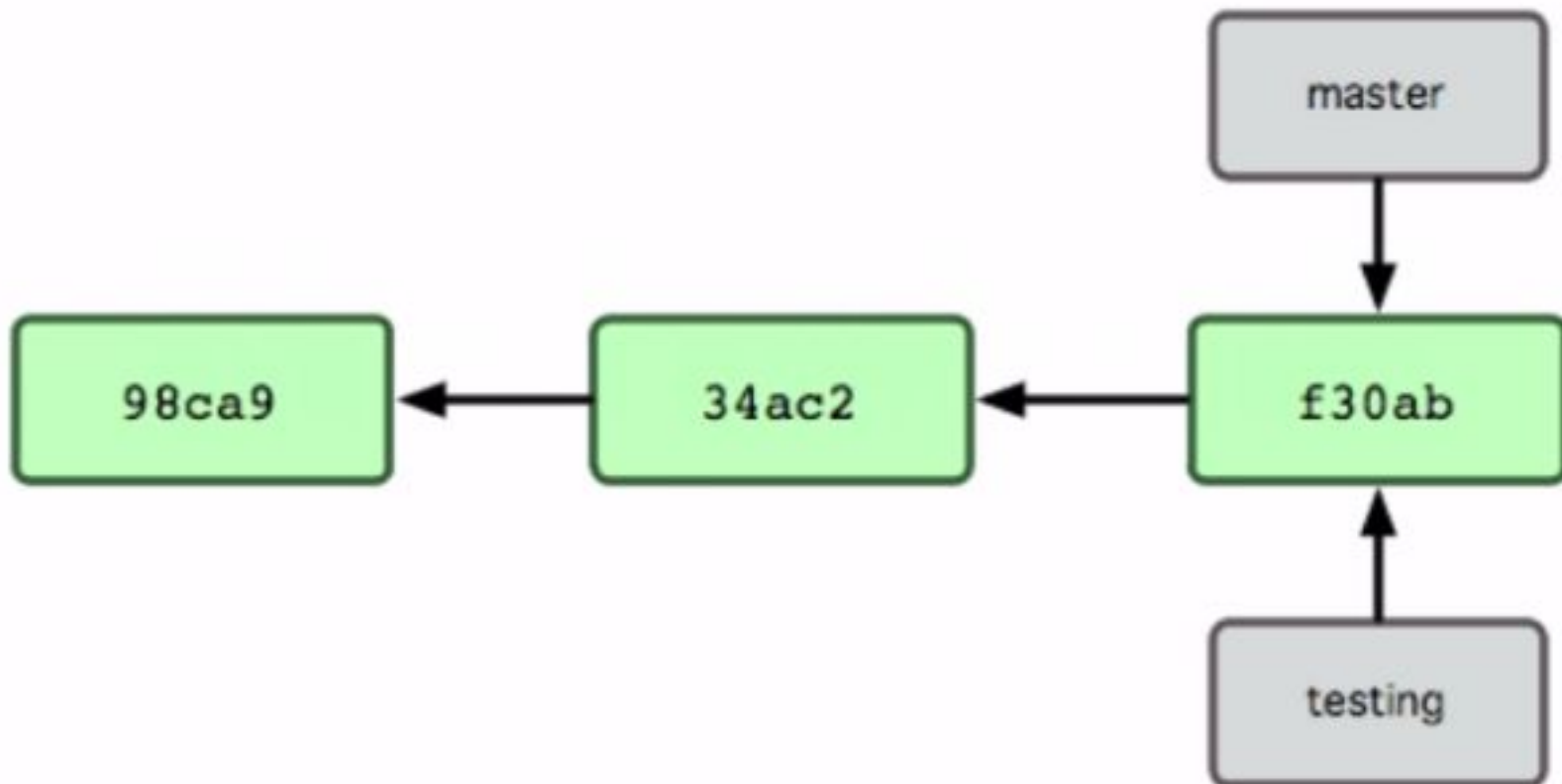


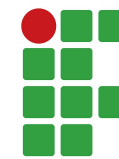
Branches

- ▷ Quando você cria um Branch, você está criando um novo ponteiro, para algum Hash de commit
 - Pode ser o próprio master ou qualquer outro.

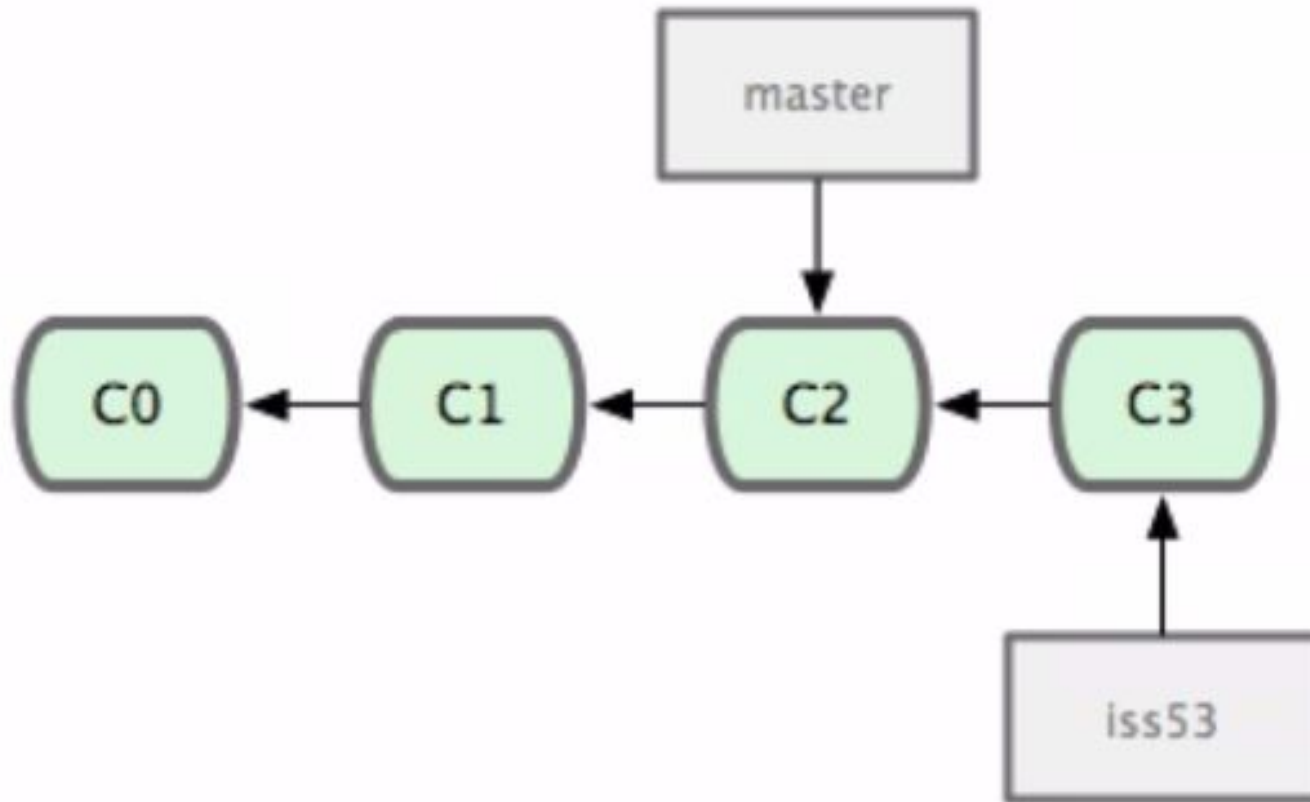


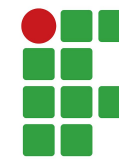
Branches





Branches

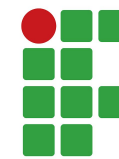




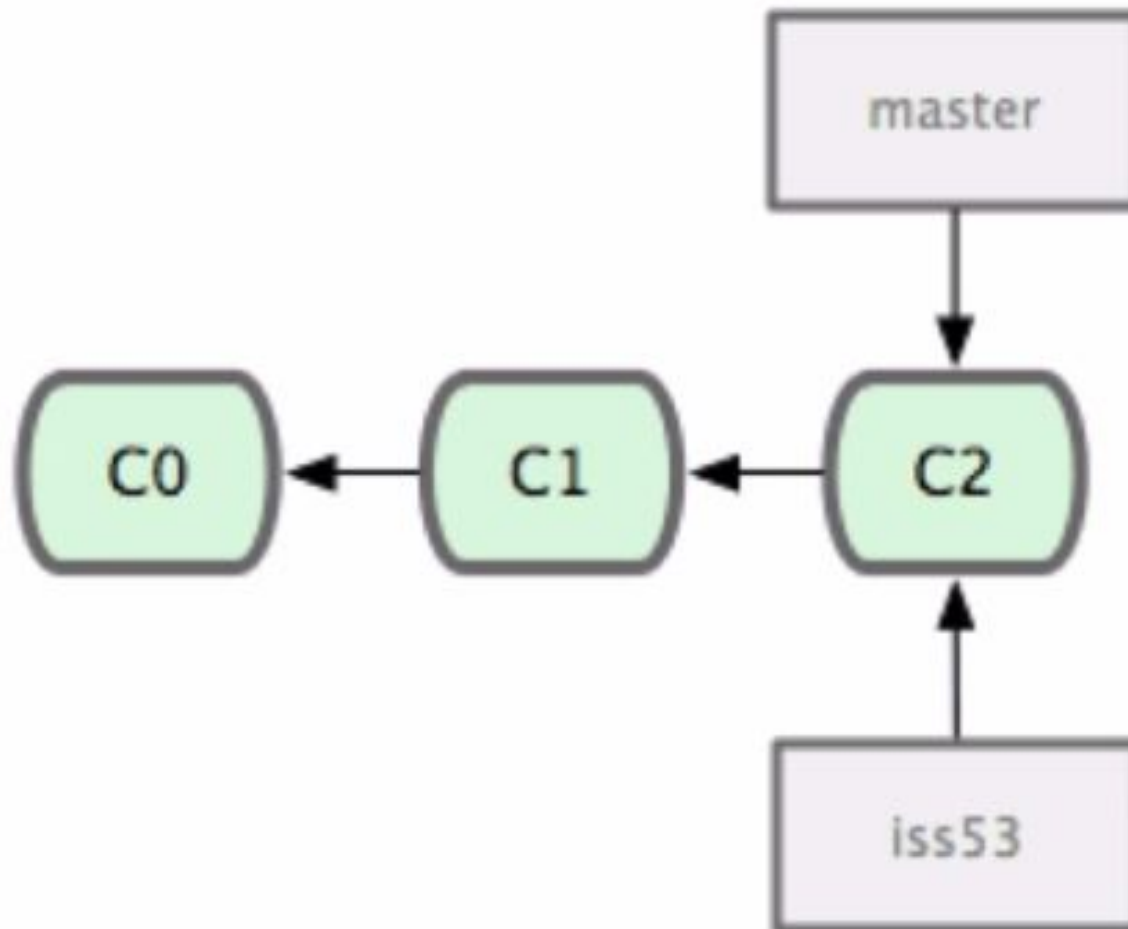
Branches

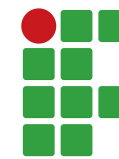
▷ Vantagens:

- Pode-se modificar o código sem alterar o branch master (o código principal)
- São facilmente “desligáveis”
 - Você pode criar e apagar os branches de forma simples e rápida
- Múltiplos desenvolvedores trabalhando em conjunto sem conflitos
 - O mesmo desenvolvedor pode estar trabalhando em mais de 1 branch

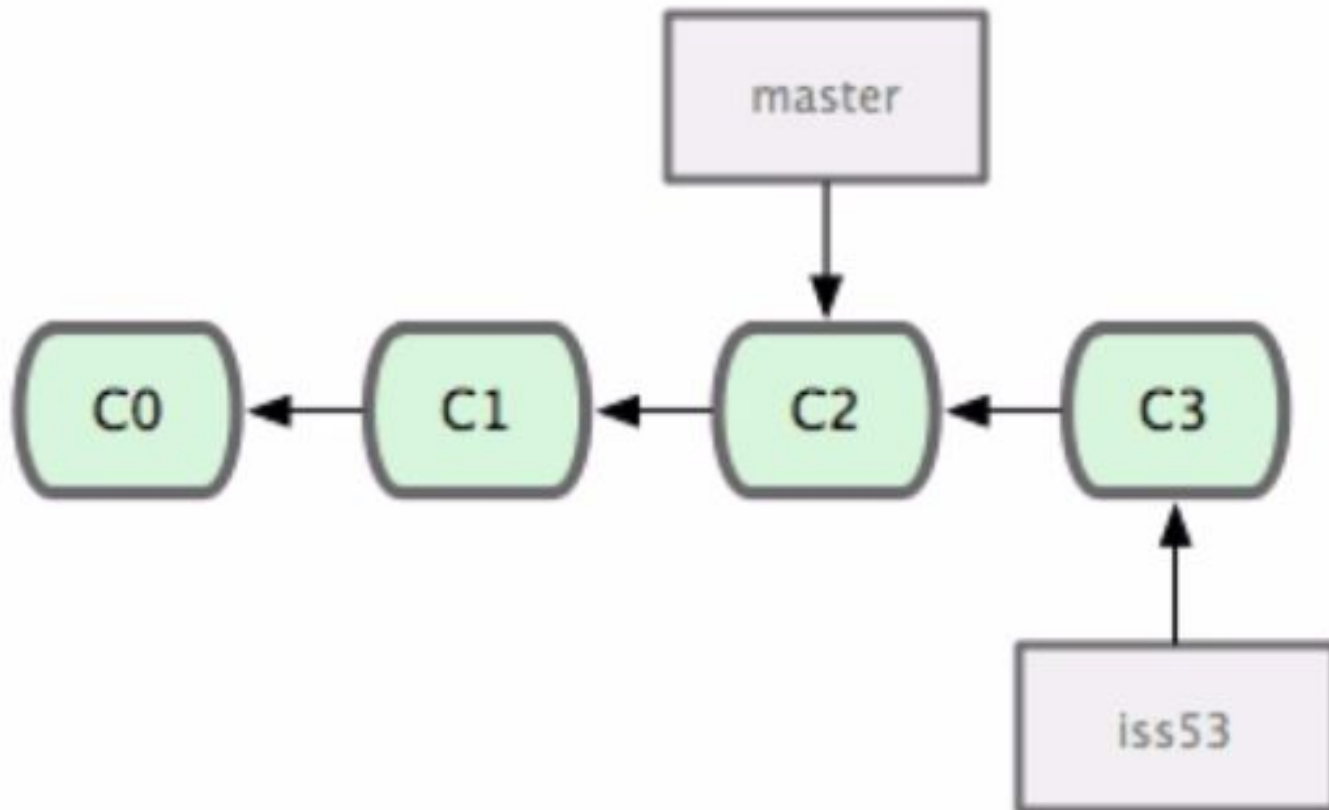


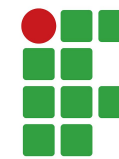
Entendendo o Merge



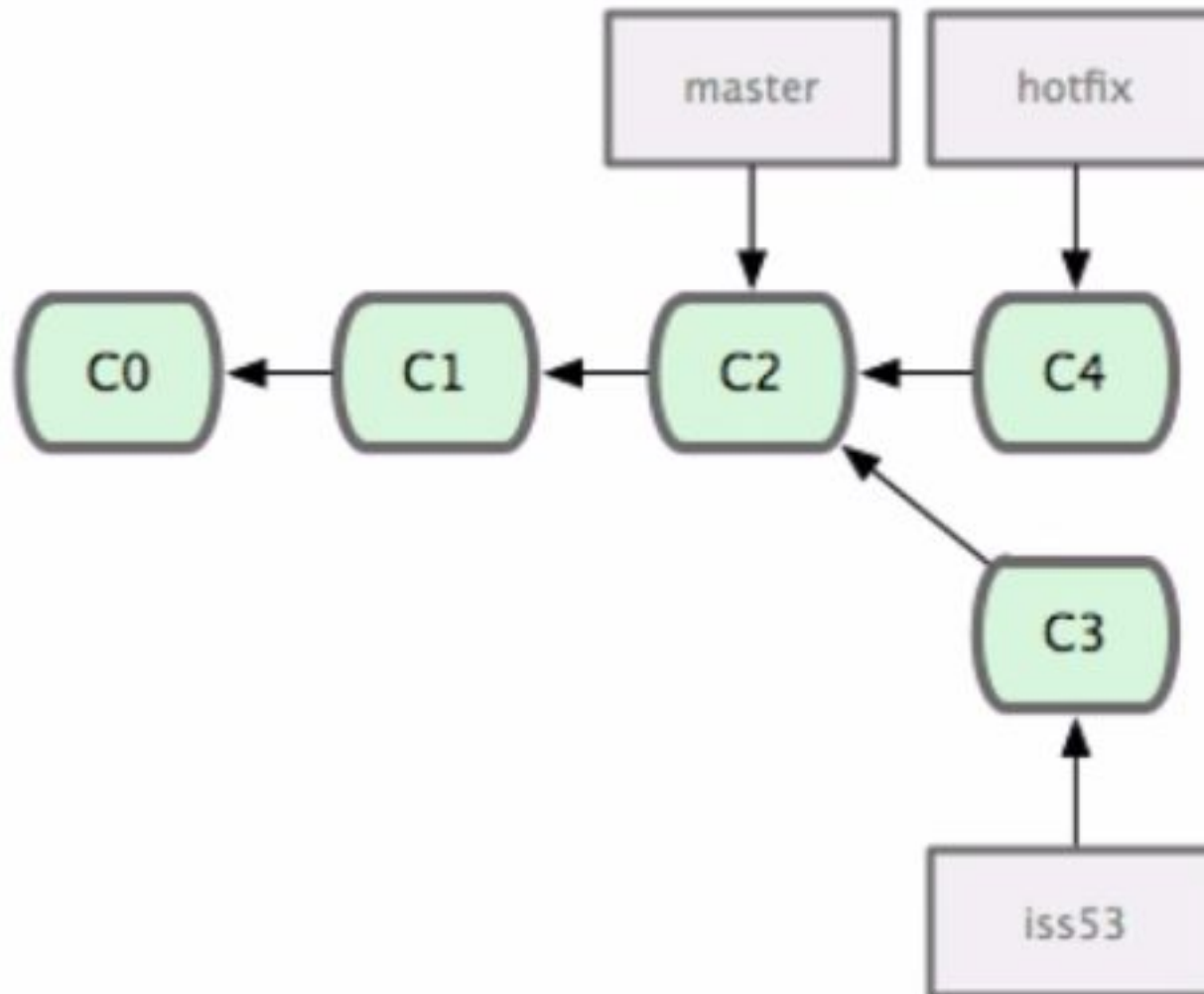


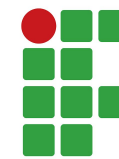
Entendendo o Merge



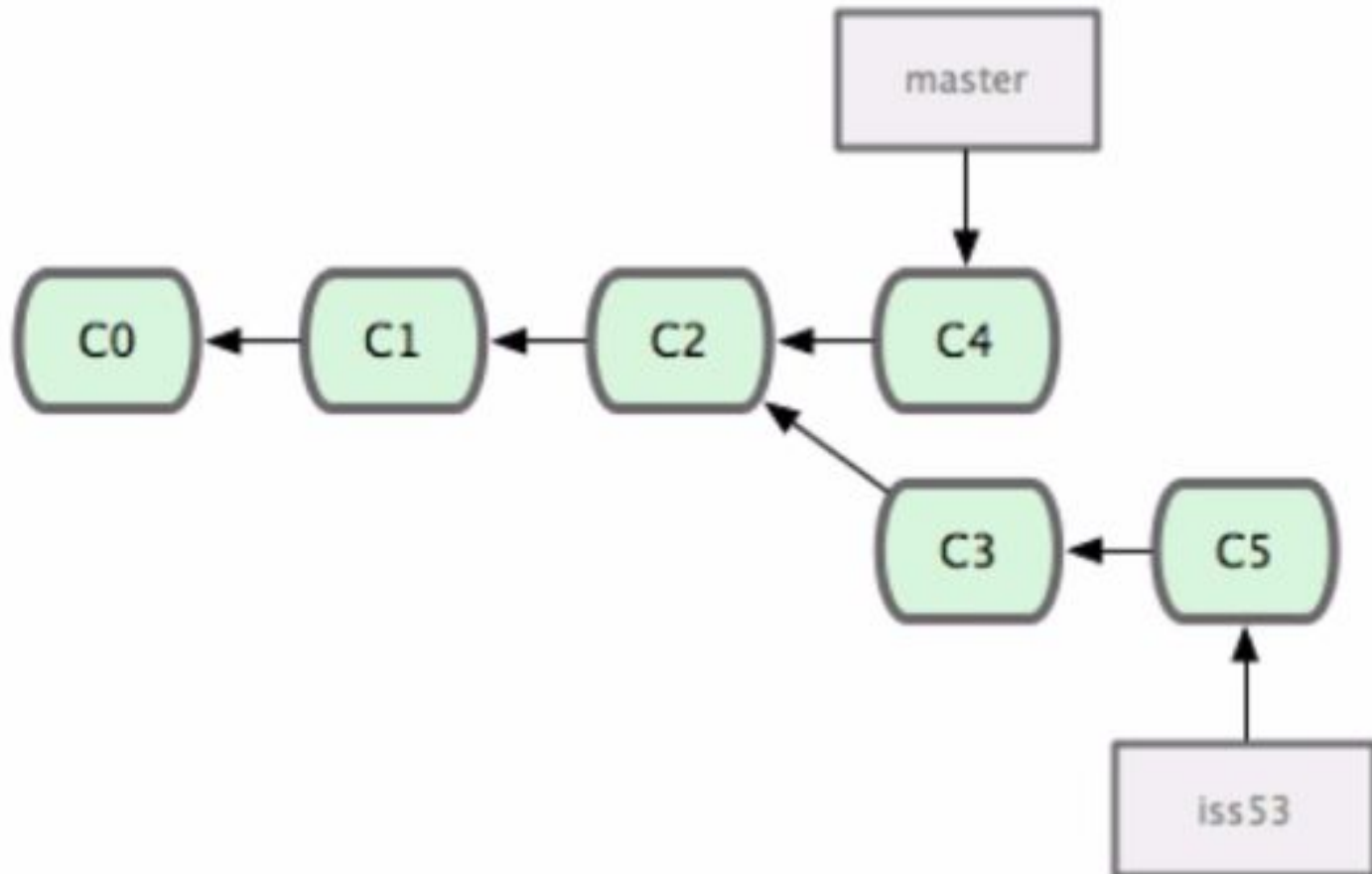


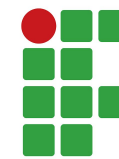
Entendendo o Merge



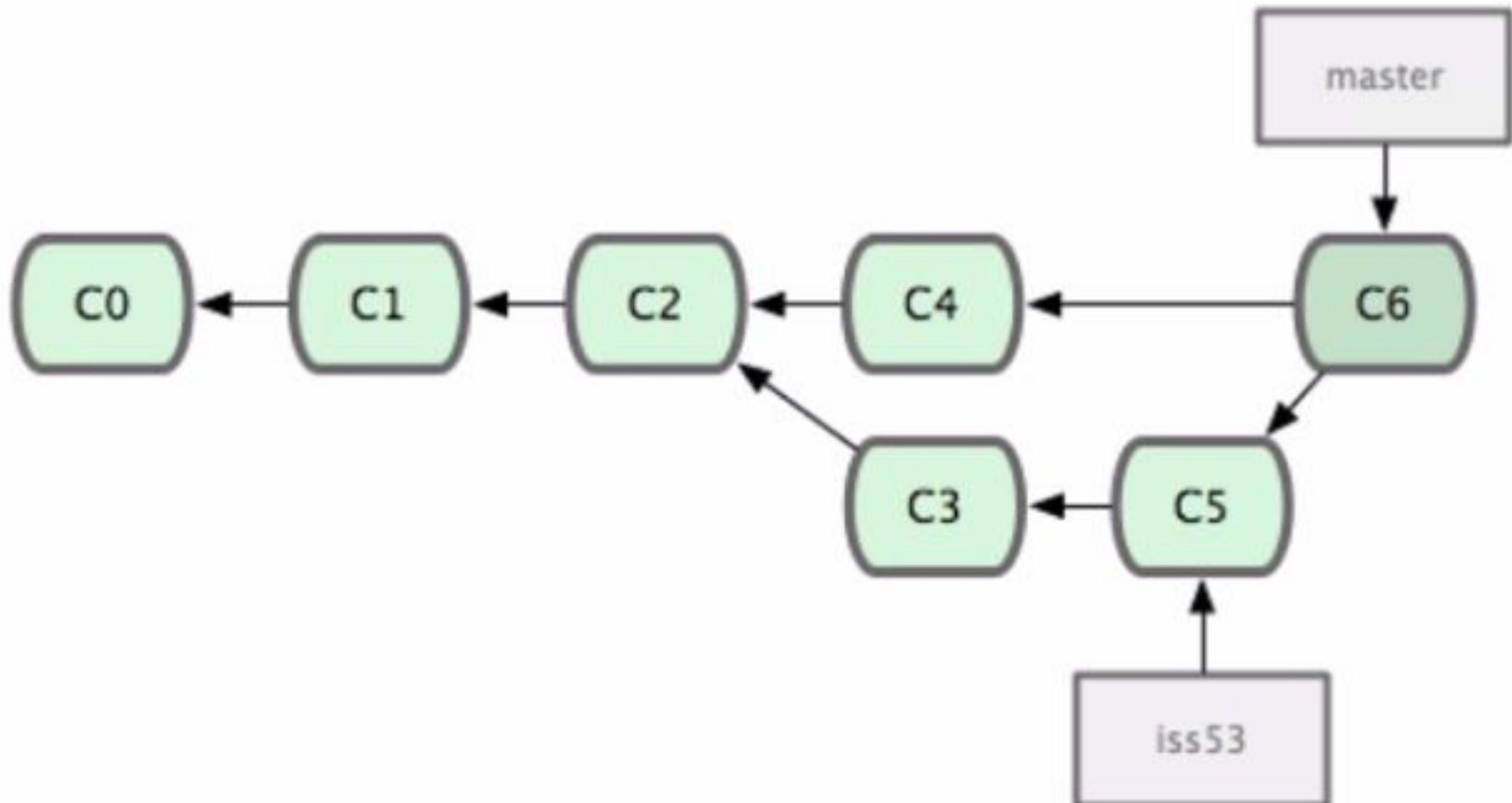


Entendendo o Merge





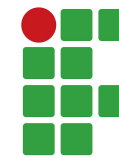
Entendendo o Merge





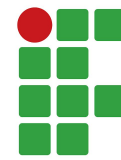
Branches

- ▷ Para fazer um branch:
 - `git branch sumario`
- ▷ Depois é necessário “ir” até o branch, ou seja, indicar para o git que é neste branch que você está trabalhando:
 - `git checkout sumario`
- ▷ Podemos abreviar em apenas 1 comando:
 - `git checkout -b sumario`



Branches

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git checkout -b sumario  
Switched to a new branch 'sumario'  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
□
```



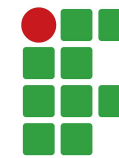
Branches

- ▷ Daqui pra frente, qualquer nova alteração será feita apenas no branch “sumario”, deixando a linha base (master) intacta.
- ▷ Aqui temos liberdade pra usar, testar, modificar e melhorar, apenas em nosso computador, com liberdade.
- ▷ Vamos adicionar um sumário simples



Branches

```
teste.html x
<html>
  <head>
    <title> Livro Git </title>
  </head>
  <body>
    <p> Sumário </p>
    <p> 1. Introdução </p>
    <p> 2. Prática </p>
    <p> 3. Conclusão </p>
  </body>
</html>
```

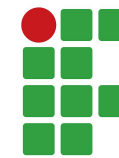


Branches

← → ↻ 🏠 ⓘ Arquivo | /home/bruno/Documentos/Trabalho/repositorios/livroGIT_palestra/teste.html

Sumário

1. Introdução
2. Prática
3. Conclusão



Branches

- ▷ Faremos novamente o processo de commit dentro do branch:

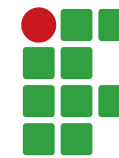


Branches

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
git add .
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
git status
No ramo sumario
Mudanças a serem submetidas:
  (use "git reset HEAD <file>..." to unstage)

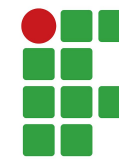
    modified:   teste.html

bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
git commit -a -m "Adicionando código HTML com o sumário"
[sumario 94b7fb5] Adicionando código HTML com o sumário
 1 file changed, 11 insertions(+)
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
git push origin sumario
Username for 'https://github.com': brunoclemente88@gmail.com
Password for 'https://brunoclemente88@gmail.com@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 431 bytes | 431.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'sumario' on GitHub by visiting:
remote:   https://github.com/bruno88/livroGIT_palestra/pull/new/sumario
remote:
To https://github.com/bruno88/livroGIT_palestra.git
 * [new branch]      sumario -> sumario
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```



Branches

- ▷ Agora, o que nos resta é fazer o merge entre dois branches
 - Branch master, onde está a linha base
 - Branch sumario, onde fizemos nossa melhoria



Branches

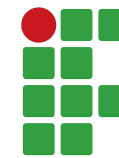
- ▷ Basta dar checkout no branch “original” e digitar:
 - Git merge sumario
 - Sendo sumario o nome do branch à ser “puxado”
- ▷ Se o git for capaz, ele fara o merge sozinho.
- ▷ Se houverem conflitos que o git não consegue fazer sozinho, ele irá pedir que você resolva o conflito manualmente, editando manualmente o arquivo



Branches

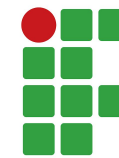
- ▷ Observe a palavra “fast-forward”:
 - O Git foi capaz de resolver o problema sozinho

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git checkout master  
Switched to branch 'master'  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git merge sumario  
Updating 2dbf5df..94b7fb5  
Fast-forward  
 teste.html | 11 ++++++++  
 1 file changed, 11 insertions(+)  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
█
```



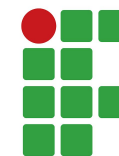
Branches

- ▷ Lembre-se que esta alteração está apenas em nosso computador.
 - Após resolver o conflito, basta enviar os dados ao GitHub com o comando push no branch “original”
 - Master, no nosso caso



Branches

```
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$  
git push origin master  
Username for 'https://github.com': brunoclemente88@gmail.com  
Password for 'https://brunoclemente88@gmail.com@github.com':  
Total 0 (delta 0), reused 0 (delta 0)  
To https://github.com/bruno88/livroGIT_palestra.git  
    2dbf5df..94b7fb5  master -> master  
bruno@bruno-Inspiron-5567:~/Documentos/Trabalho/repositorios/livroGIT_palestra$
```



Branches

- Podemos conferir na página do GitHub a alteração confirmada.

Livro sobre Git que servirá como exemplo para a Palestra de uso do Git.

Edit

[Manage topics](#)

3 commits

2 branches

0 releases

1 contributor

Branch: master ▼

New pull request

Create new file

Upload files

Find file

Clone or download ▼

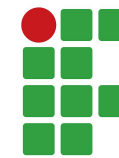
 **bruno88** Adicionando código HTML com o sumário Latest commit 94b7fb5 6 minutes ago

 [README.md](#) Initial commit 27 minutes ago

 [teste.html](#) Adicionando código HTML com o sumário 6 minutes ago

 [README.md](#)





Branches

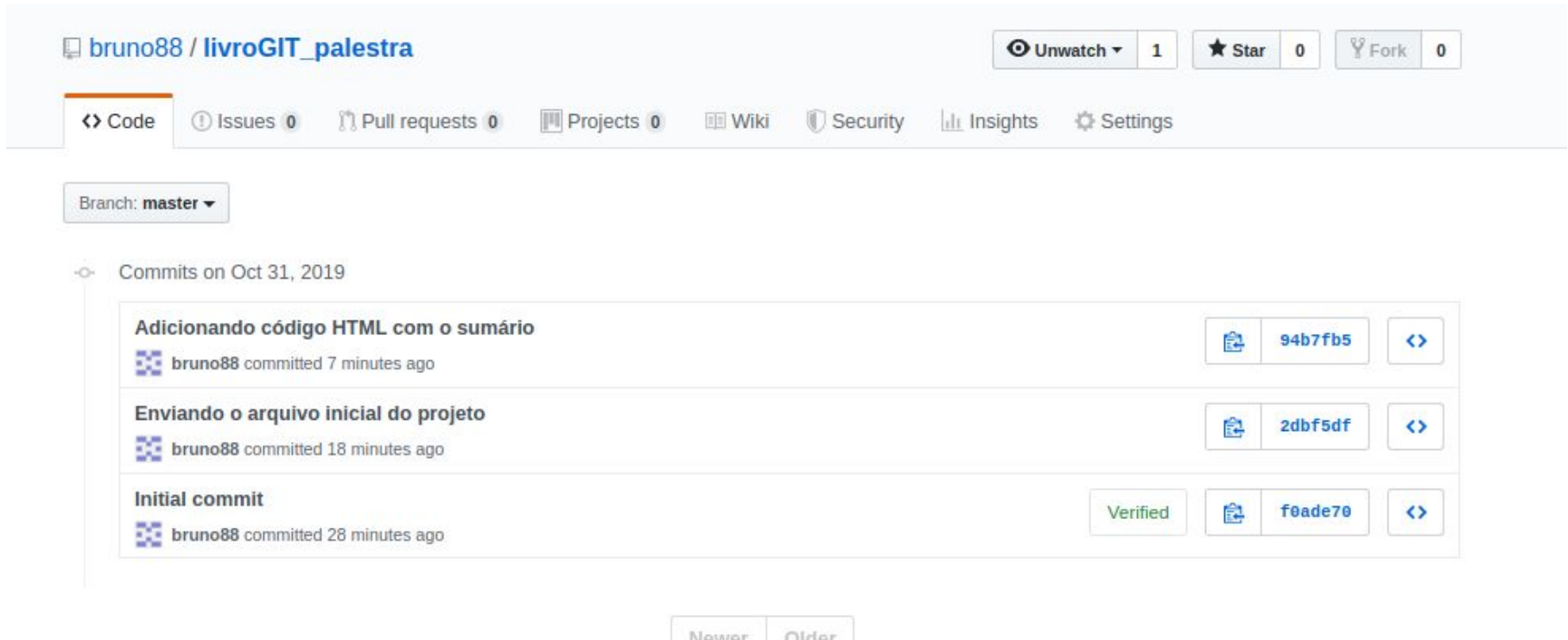
- ▷ EXTRA: E se tivesse dado errado?
 - O git indicará que houve um conflito e você deverá abrir o arquivo e alterá-lo.
 - O arquivo aparecerá como abaixo:

```
<<<<<< HEAD:file.txt
Hello world
=====
Goodbye
>>>>>> 77976da35a11db4580b80ae27e8d65caf5208086:file.txt
```

Ou seja, ambas as alterações estão presentes, basta você definir como ficará e salvar !

GitHub

- Na aba Commits, podemos ver todos eles:



bruno88 / livroGIT_palestra

Unwatch 1 Star 0 Fork 0

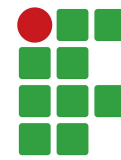
Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Branch: master ▼

Commits on Oct 31, 2019

Adicionando código HTML com o sumário bruno88 committed 7 minutes ago	94b7fb5	<>
Enviando o arquivo inicial do projeto bruno88 committed 18 minutes ago	2dbf5df	<>
Initial commit bruno88 committed 28 minutes ago	Verified f0ade70	<>

Newer Older



GitHub


- ▷ Observe que lá temos:
 - Nome do usuário que fez o envio;
 - Quando foi feito o commit (data e hora)
 - Qual a descrição do commit, o texto enviado com o comando.
 - Há também, para cada commit, um código hash, como “94b7fb5”, por exemplo.
 - E se clicarmos neste código...














INSTITUTO FEDERAL

PARÁ
Campus Paragominas


GitHub


 **bruno88** / **livroGIT_palestra**


 Unwatch ▾ 1  Star 0  Fork 0




 Code  Issues 0  Pull requests 0  Projects 0  Wiki  Security  Insights  Settings



Adicionando código HTML com o sumário



 master

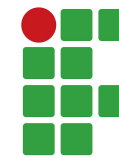
 **bruno88** committed 9 minutes ago 1 parent [2dbf5df](#) commit [94b7fb53d04c186abee6970739875ea99b360ac5](#)

 Showing **1 changed file** with **11 additions** and **0 deletions**.  

▼ 11  teste.html 

```
...    ...    @@ -0,0 +1,11 @@
1      + <html>
2      +   <head>
3      +     <title> Livro Git </title>
4      +   </head>
5      +   <body>
6      +     <p> Sumário </p>
7      +     <p> 1. Introdução </p>
8      +     <p> 2. Prática </p>
9      +     <p> 3. Conclusão </p>
10     +   </body>
11     + </html>
```

0 comments on commit [94b7fb5](#)



GitHub

- ▷ Nesta nova tela, além das informações do Commit já existentes, temos uma demonstração:
 - Arquivo à Arquivo
 - Linha à Linha
- ▷ De tudo que foi feito neste commit.
 - Linhas verdes, com o sinal de + foram adicionadas.
 - Linhas vermelhas, com o sinal de - foram removidas.

GitHub

- ▶ Nesta mesma tela também há um espaço para comentários. O GitHub funciona como uma rede social !

0 comments on commit 94b7fb5

 Lock conversation



Write

Preview

AA B *i* “ <> 🔗 ☰ ☷ ✓ @ 📌 ↶

Leave a comment

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



Comment on this commit

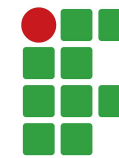


Unsubscribe

You're receiving notifications because you're watching this repository.

GitHub

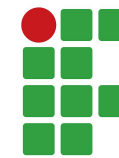
- ▷ O GitHub tem diversas funcionalidades sociais, como:
 - Clone: onde você clona o repositório de outra pessoa;
 - Pull request: onde você pode contribuir com o projeto de outras pessoas;
 - Issues: onde você pode indicar as tarefas pendentes
 - Todo um sistema social de curtidas (star), acompanhamento (watch) e comentários.



GitHub

- ▶ Mas com o que vimos hoje, já é possível controlar as versões de seu software ou qualquer outro tipo de arquivos.





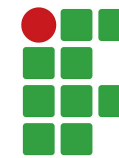
GitHub

- ▷ Última curiosidade....
 - Lembra que o commit é DEFINITIVO?

- ▷ E se eu errar? O que eu faço?

- ▷ `git revert <hash do commit>`
 - Lembra, aquele da aba commits?

GitHub



INSTITUTO FEDERAL

PARÁ
Campus Paragominas

bruno88 / livroGIT_palestra

Unwatch ▾

1

★ Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Branch: master ▾

Commits on Oct 31, 2019

Adicionando código HTML com o sumário

bruno88 committed 7 minutes ago



94b7fb5



Enviando o arquivo inicial do projeto

bruno88 committed 18 minutes ago



2dbf5df



Initial commit

bruno88 committed 28 minutes ago

Verified

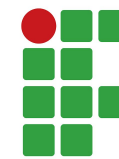


f0ade70



Newer

Older



GitHub

- ▷ Então, se eu quisesse reverter o commig que adicionou o sumário, poderíamos fazer:
 - `git revert 94b7fb5`
- ▷ Não é possível desfazer o commit, mas pode-se fazer um novo, que “desfaça” tudo que foi feito, ou seja:
 - Adicionar o que foi removido.
 - Remover o que foi adicionado.
- ▷ E é assim que o Git armazena seus dados !

GitHub

- ▷ Isso nos abre diversas possibilidades:
 - Várias pessoas trabalhando ao mesmo tempo em um único arquivo;
 - Controle de versão e alteração;
 - Histórico de atualizações;
 - Alta disponibilidade: Seu projeto pode ser acessado de qualquer lugar (mesmo um sem Git instalado).
 - Segurança e integridade dos seus dados.



INSTITUTO FEDERAL

PARÁ
Campus Paragominas

Obrigado pela atenção!

Dúvidas?

Contatos:

E-mail: bruno.clemente@ifes.edu.br

Celular/Whatsapp: +55 27 99978-3993