



SOFTEX
PERNAMBUCO

 **Softex**

GOVERNO
FEDERAL
MINISTÉRIO DA
EDUCAÇÃO
E INOVAÇÃO

BRASIL
união é o nosso futuro



Aula 04 | Linha de comando & Git



Interface de Linha de Comando (Command Line)

- O que é:
 - Uma interface de texto onde você digita comandos para interagir com o computador, sem precisar de janelas ou cliques.





Linha de Comando (Command Line)



- **Pra que serve:**
 - Navegar entre pastas e arquivos.
 - Criar, copiar, mover e apagar arquivos.
 - Instalar e gerenciar programas.
 - Automatizar tarefas usando scripts.
- **Vantagens:**
 - Mais rápido para tarefas repetitivas.
 - Funciona em servidores e ambientes sem interface gráfica.
 - Ferramenta essencial para programadores, DevOps e administradores de sistemas.



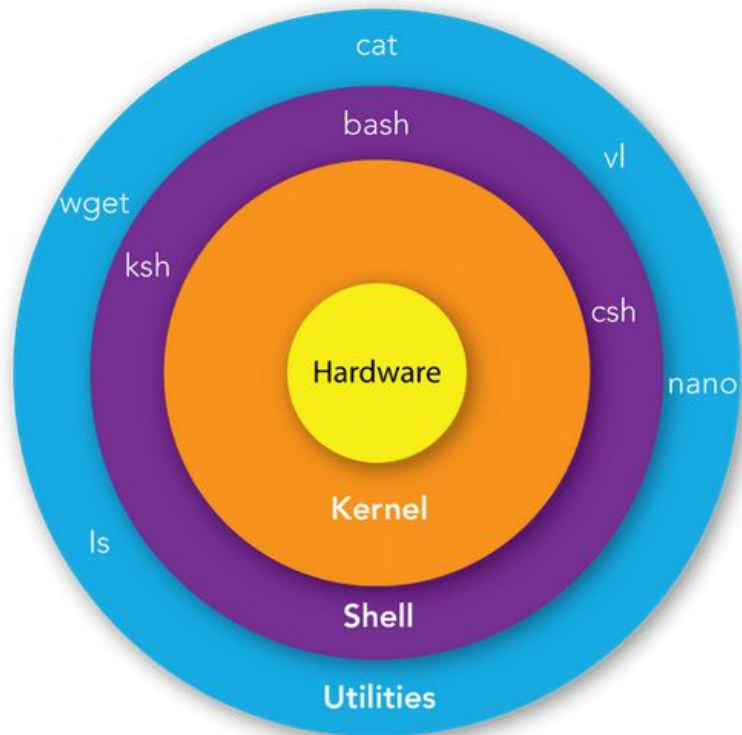
Shell



- **O que é:**
 - É o programa que interpreta e executa os comandos que você digita na linha de comando.
- **Linha de comando vs Shell:**
 - A linha de comando é a tela onde você digita. O shell é o cérebro por trás que entende os comandos e fala com o sistema operacional.
- **Exemplos:**
 - Bash(Bourne-Again Shell), Zsh (Z-Shell), CMD (Command Prompt)
- **Significado**
 - Chamamos de Shell (que em inglês significa "casca") pois ele "envolve" o kernel, funcionando como a camada que permite que o usuário interaja com o sistema



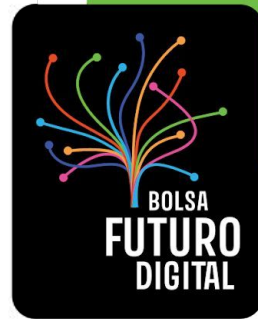
Shell





Bash

- **O que é:**
 - Uma das shells mais populares do Linux/UNIX, que interpreta comandos digitados no terminal.
- **Como usar:**
 - **Linux e MacOS:** O Bash já vem pré-instalado e é o shell padrão nesses sistemas. Não há necessidade de instalar nada
 - **Windows:** Uma das formas é instalando o **Git for Windows**, ele inclui uma ferramenta chamada **Git Bash**.





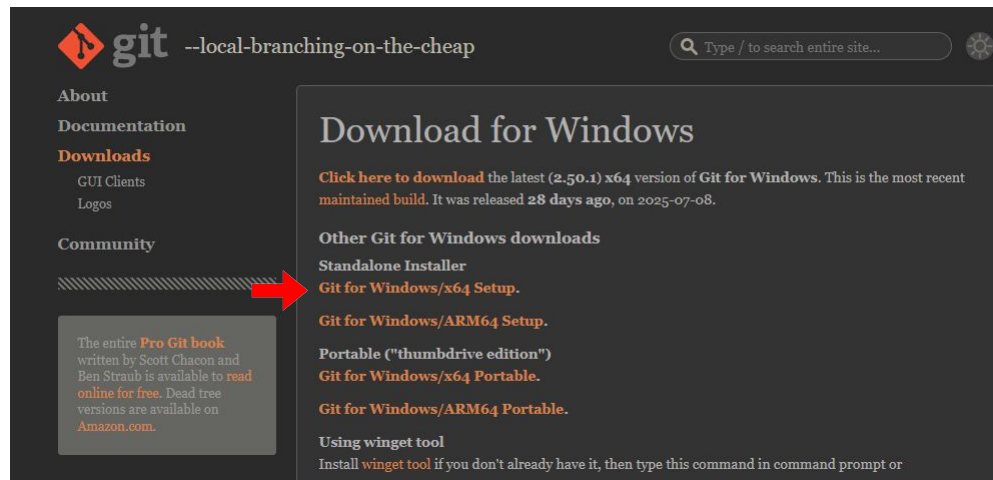
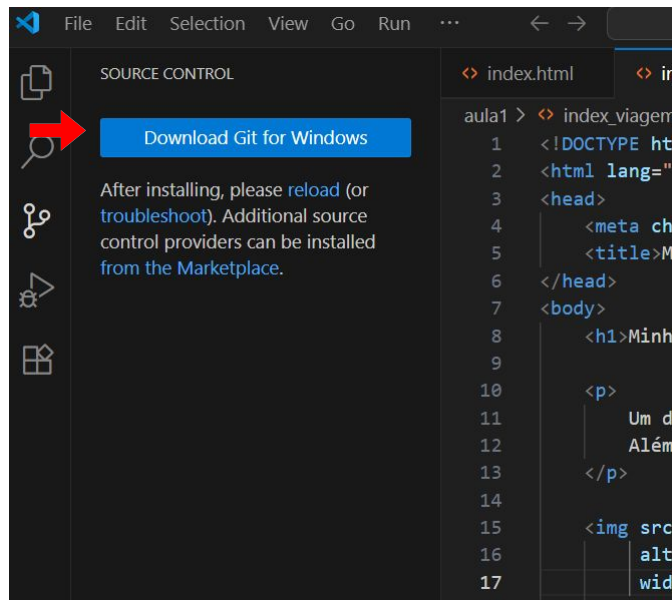
Git Bash



- O que é:
 - É um emulador de terminal que fornece um ambiente Bash para rodar comandos do Git, mas também é útil para executar a maioria dos comandos Bash



INSTALANDO O GIT e o GIT BASH





Git Bash

Usando a linha de comando do linux

ls

- Abreviação de “list”, esse comando mostra todos os arquivos e pastas do diretório em que você está no momento.





Git Bash

Usando a linha de comando do linux

cd <diretorio>

- Abreviação de “**C**hange **D**irectory”, esse comando muda o diretório atual para o <diretório> que você indicar
- O atalho para o diretório atual é .
- O atalho para o diretório pai (um nível acima) é ..
- Se bater a dúvida sobre em qual diretório você está (apesar de o próprio terminal geralmente mostrar isso no prompt), basta digitar pwd (present working directory).





Git Bash

Usando a linha de comando do linux

`mkdir <diretorio>`

- Abreviação de “make directory”, esse comando cria um novo subdiretório chamado `<diretório>` dentro do diretório atual.





Git Bash

Usando a linha de comando do linux

`cp <origem> <destino>`

- Abreviação de “copy”, esse comando permite criar uma cópia do **arquivo** que você indicar como `<origem>`, salvando-a em `<destino>`.
- Se você quiser copiar **diretórios** inteiros, precisará ajustar o comando um pouco:

`cp -r <diretorio de origem> <diretorio de destino>`

O “-r” significa recursivo e indica ao cp que ele deve entrar no diretório e copiar tudo que estiver dentro dele (incluindo quaisquer subdiretórios que existam).





Git Bash

Usando a linha de comando do linux

`rm <arquivo>`

- Abreviação de “remove”, esse comando vai apagar o `<arquivo>` depois de pedir sua confirmação (y/n) para deletá-lo:
- Você pode pular a confirmação digitando: `rm -f <arquivo>`
- Para apagar **diretórios** inteiros, você precisa usar a flag -r, assim como acontece com o cp:
 - `rm -r <diretorio>`
- Você também pode combinar as flags -r e -f em -rf.





Git Bash

Usando a linha de comando do linux

`mv <origem> <destino>`

- Abreviação de “move”, esse comando permite, na prática, renomear um arquivo, movendo-o de `<origem>` para `<destino>`.





TRABALHANDO com GIT e GitHub

- Criando conta no GitHub

<https://github.com/signup?source=login>

Sign up to GitHub

Email*

dunfrey.aragao@servidor.uepb.edu.br

Password*

.....

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

Username*

Prof-Dunfrey

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

Your Country/Region*

Brazil

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

Create account >

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.



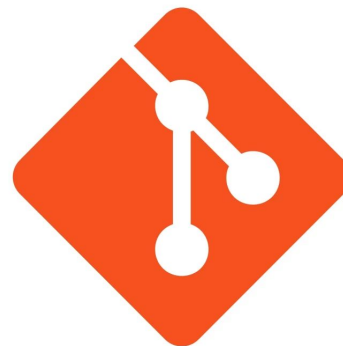


Git



O que é Git?

- **Sistema de Controle de Versão:** É o programa que você instala no seu computador.
- **Função:** Rastreia as alterações nos seus arquivos e projetos.
- **Local:** Funciona no seu computador, controlando o histórico de versões.
- **Comandos:** Você usa comandos como `git commit` e `git branch` para interagir com ele.
- **O "motor":** É a tecnologia que permite o versionamento.

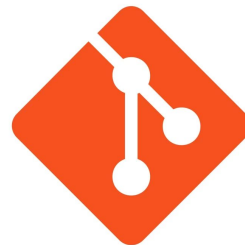




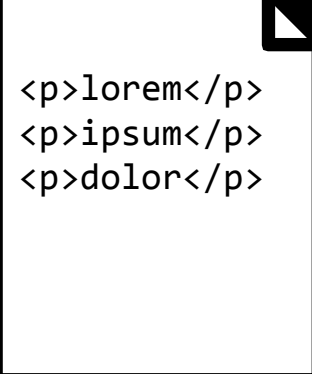
Git & Github

O que podemos fazer com o git e o github?

1. Acompanhar alterações no código
2. Sincronizar código entre diferente pessoas
3. Testar alterações no código sem perder versão origina
4. Voltar a versões anteriores



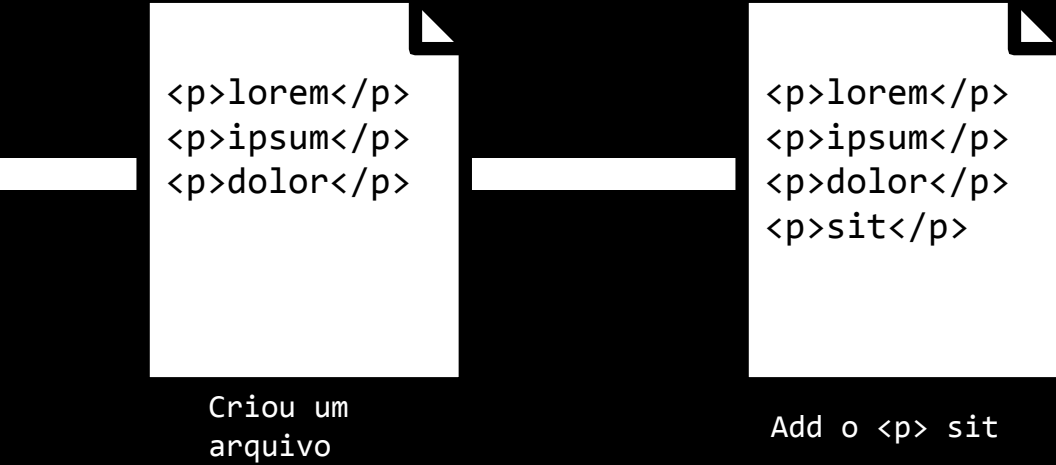
1 - Acompanhar as alterações no código



```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

Criou um
arquivo

1 - Acompanhar as alterações no código




```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

Criou um
arquivo

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<p>sit</p>
```

Add o <p> sit

1 - Acompanhar as alterações no código



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```

Criou um
arquivo

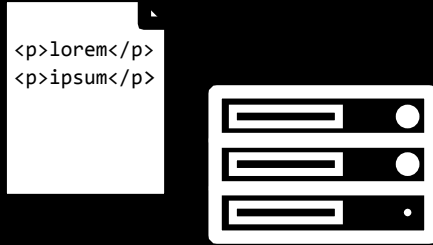
```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<p>sit</p>
```

Add o <p> sit

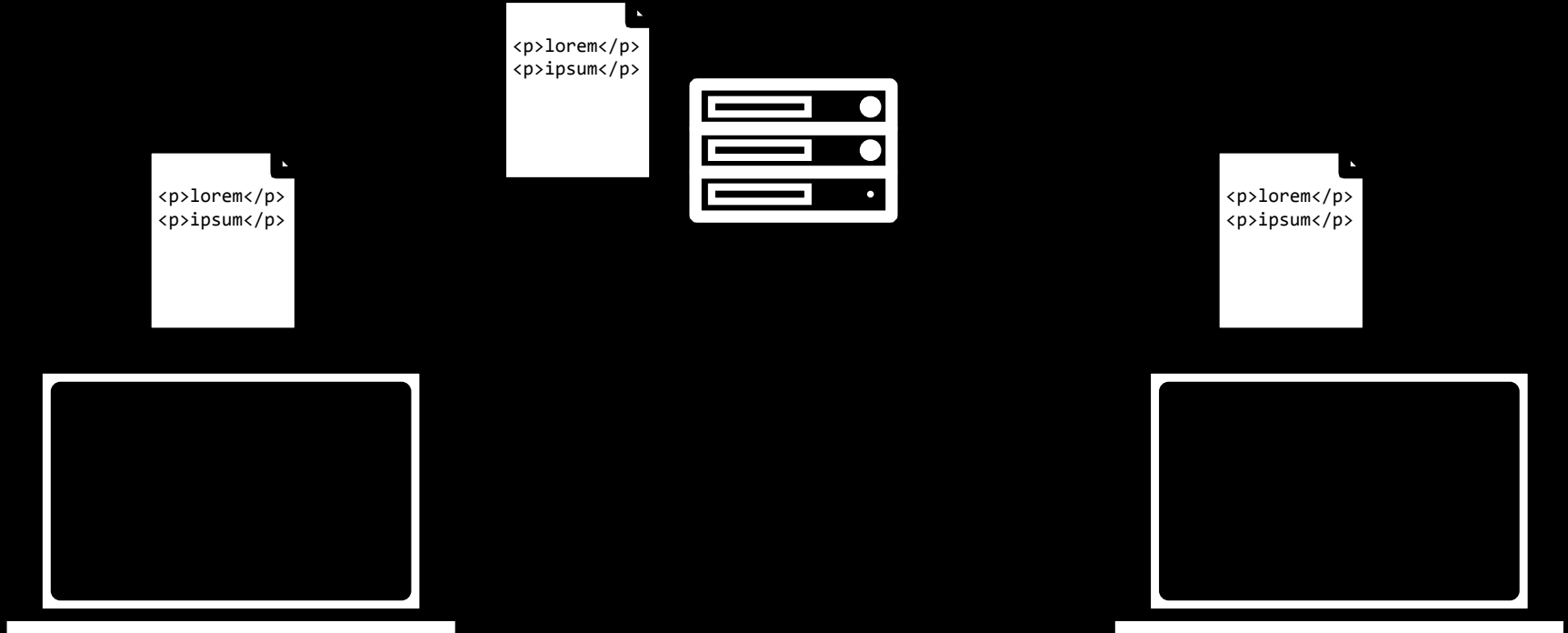
```
<p>lorem</p>
<p>dolor</p>
<p>sit</p>
```

Removeu o <p> ipsum

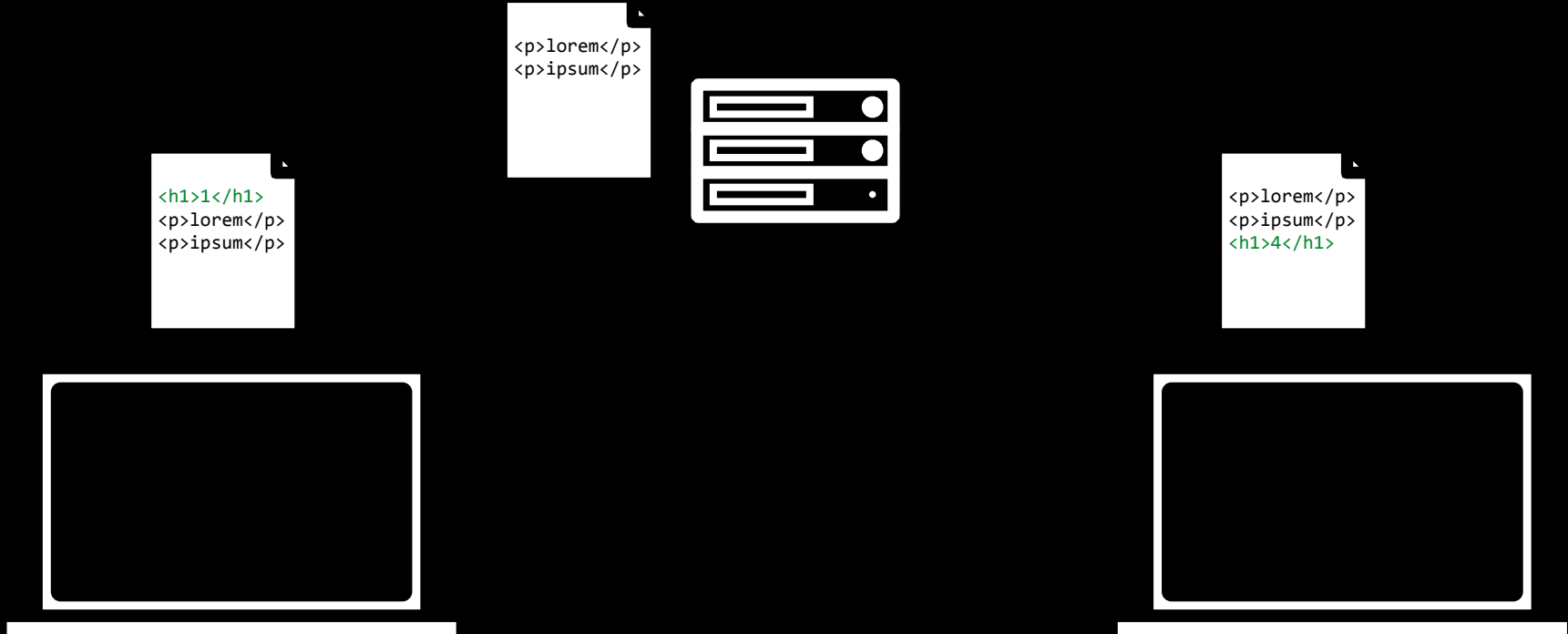
2 - Sincronizar o código entre diferentes pessoas



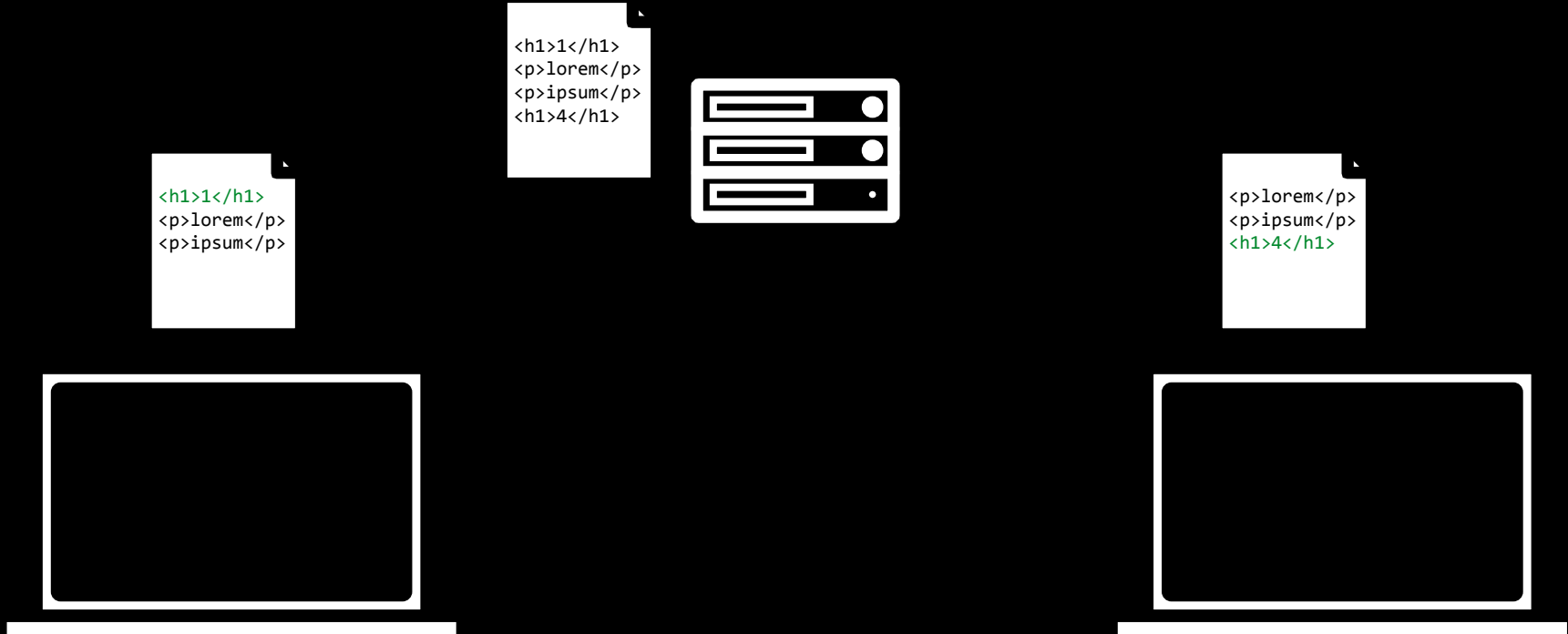
2 - Sincronizar o código entre diferentes pessoas



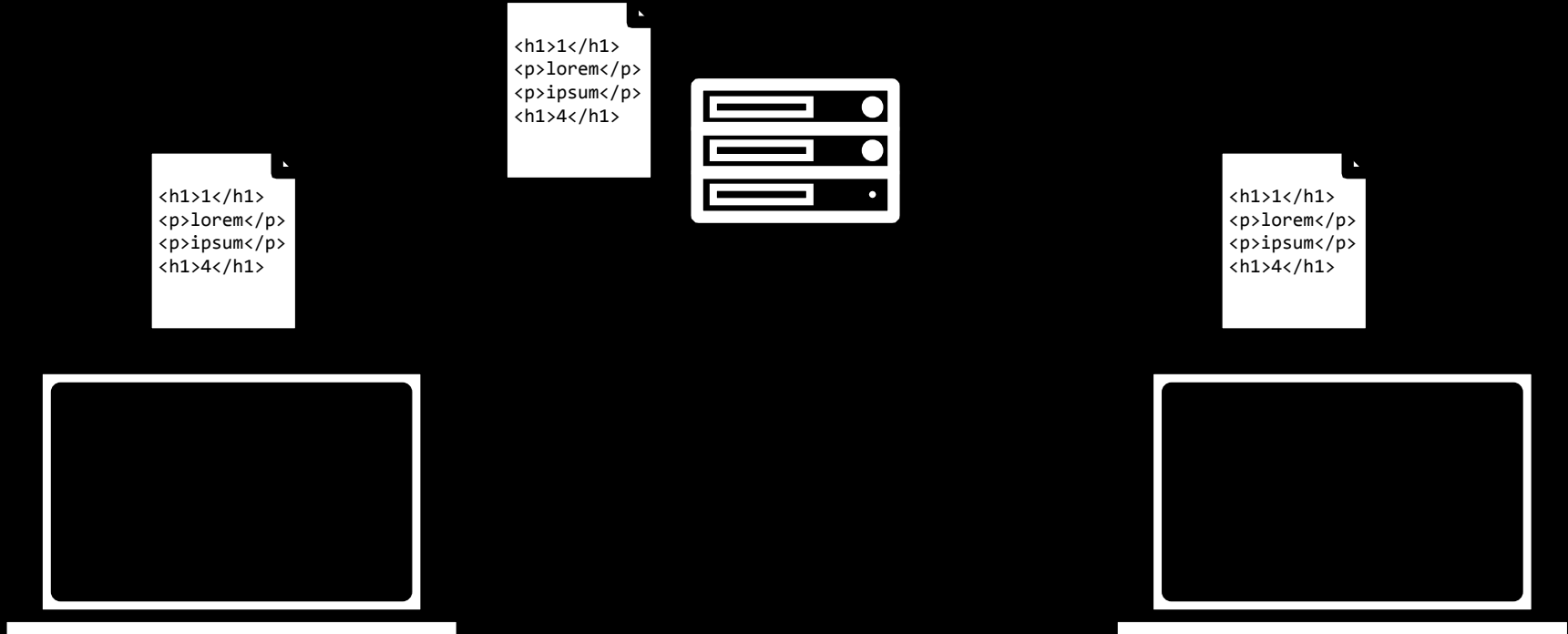
2 - Sincronizar o código entre diferentes pessoas



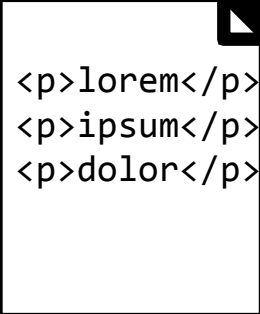
2 - Sincronizar o código entre diferentes pessoas



2 - Sincronizar o código entre diferentes pessoas

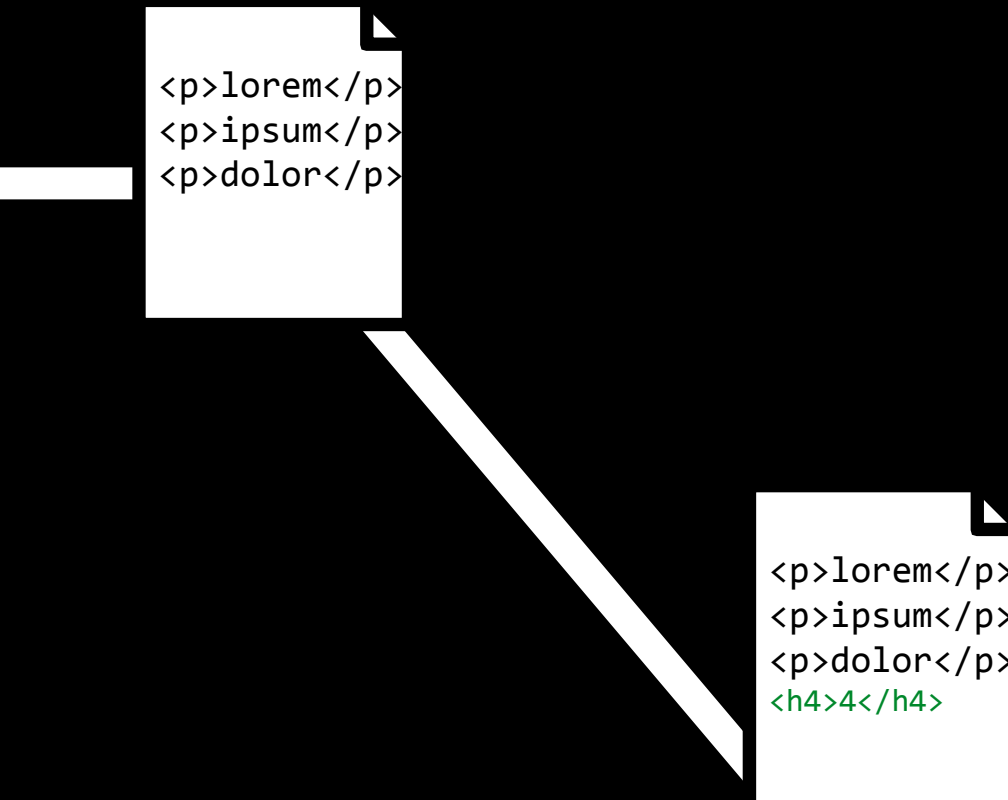


3 - Teste alterações no código sem perder a versão original



```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

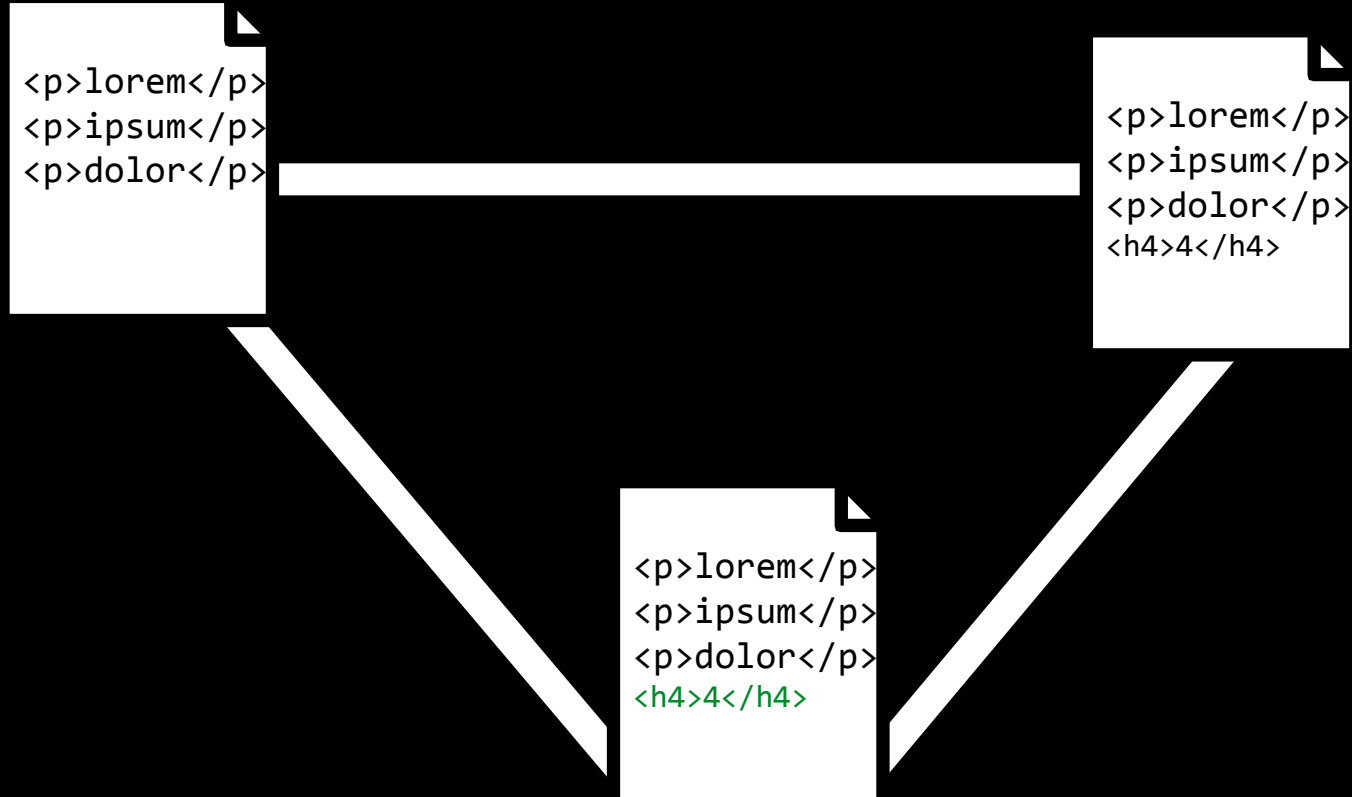
3 - Teste alterações no código sem perder a versão original



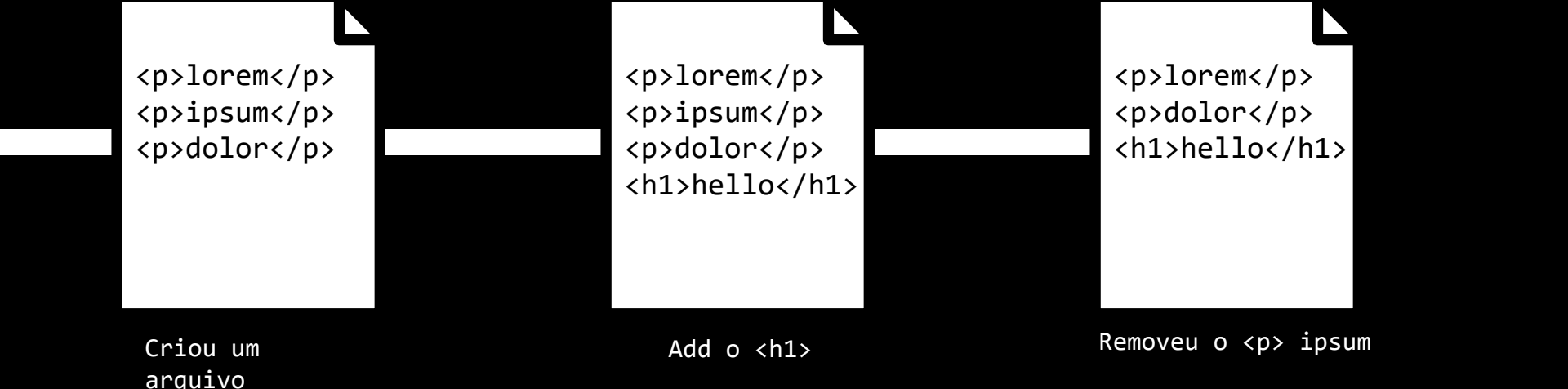
```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h4>4</h4>
```

3 - Teste alterações no código sem perder a versão original



4 - Voltar a versões anteriores do código



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```

Criou um
arquivo

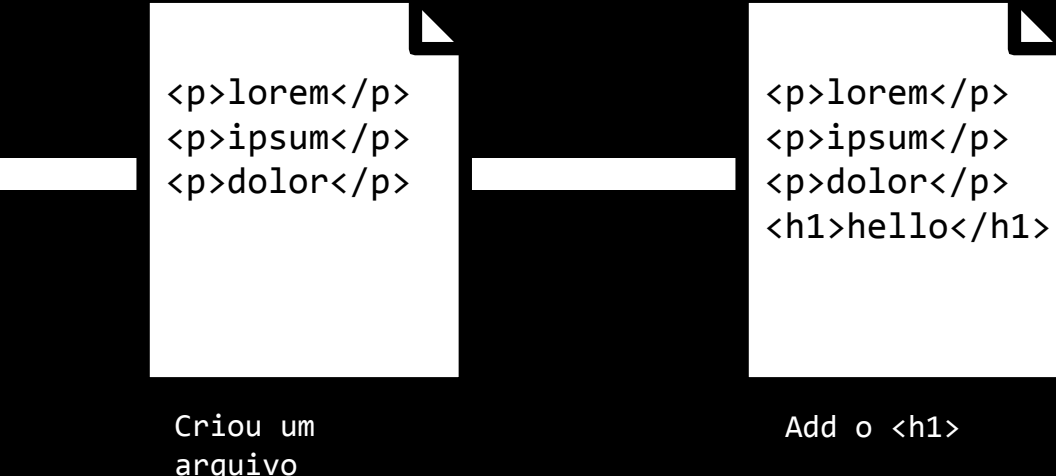
```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>hello</h1>
```

Add o <h1>

```
<p>lorem</p>
<p>dolor</p>
<h1>hello</h1>
```

Removeu o <p> ipsum

4 - Voltar a versões anteriores do código



```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

Criou um
arquivo

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>hello</h1>
```

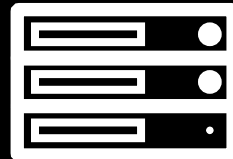
Add o <h1>

GitHub

git clone

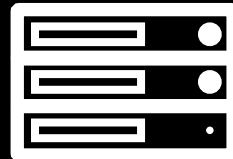
```
git clone <url>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



```
git clone <url>
```

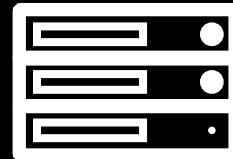
```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



```
git clone  
<url>
```

```
git clone <url>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



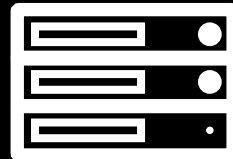
```
git clone  
<url>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

```
git add
```

```
git add <filename>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

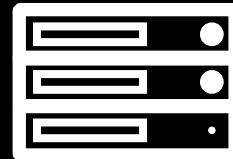


```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



```
git add <filename>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>4</h1>
```

```
git add <filename>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



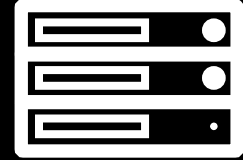
```
git add  
index.html
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>4</h1>
```



```
git add <filename>
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



```
git add  
index.html
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>4</h1>
```

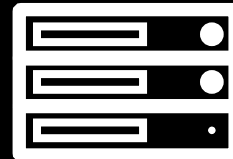
Alterações a serem
commitadas:

Modificado: index.html

git commit

```
git commit -m "message"
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

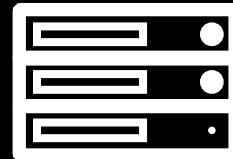


```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>4</h1>
```



```
git commit -m "message"
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```

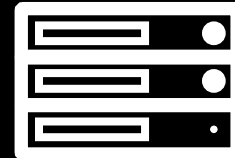


```
git commit  
-m  
    "Add  
    h1"
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>4</h1>
```

```
git commit -m "message"
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



```
git commit  
-m  
    "Add  
    h1"
```

```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



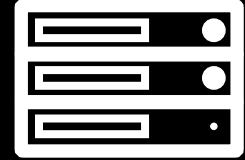
```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>4</h1>
```

Add h1

git status

git status

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```

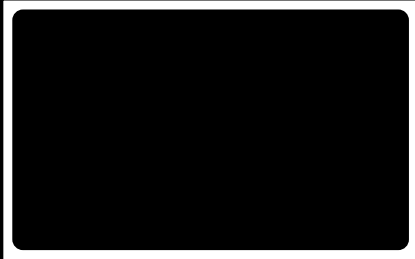


```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



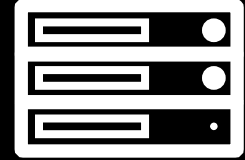
```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Add h1



git status

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

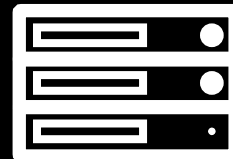
Add h1



```
git
status
```


git status

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Add h1

```
git
status
```

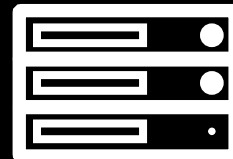
On branch master

Your branch is ahead of 'origin/master'
by 1 commit.

(use "git push" to publish your local
commits)

git status

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Add h1

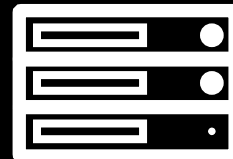
```
git
status
```

Na branch master
Sua branch está 1 commit à frente de
origin/master.
(use git push para publicar seus
commits locais)

git push

git push

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



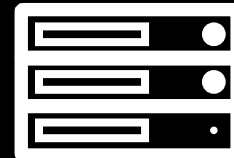
```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Add h1



git push

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```



```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```

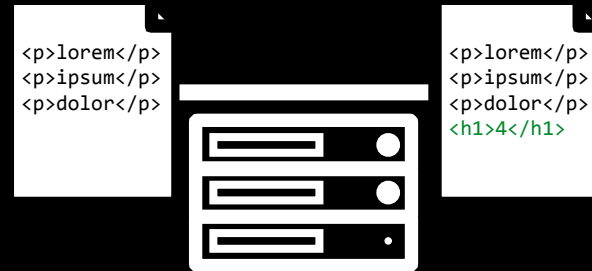


```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Add h1

git push

git push

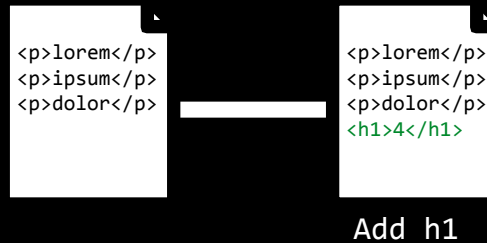
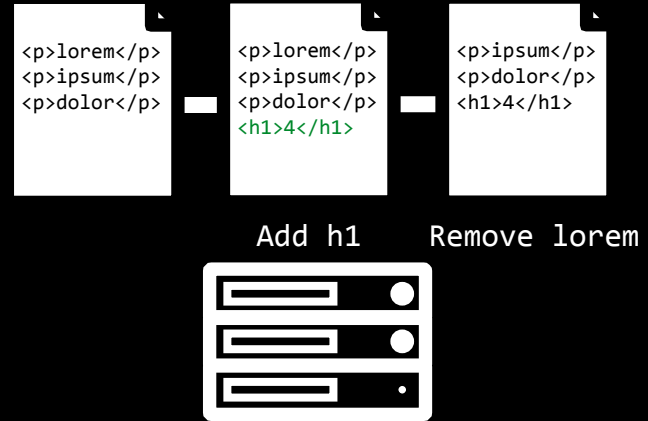


Add h1

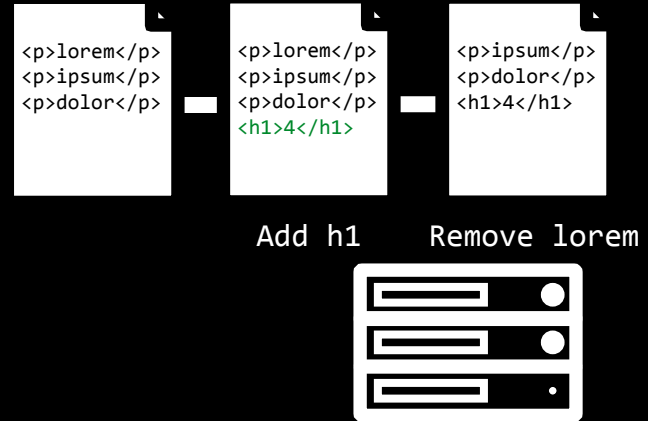
git push

```
git pull
```

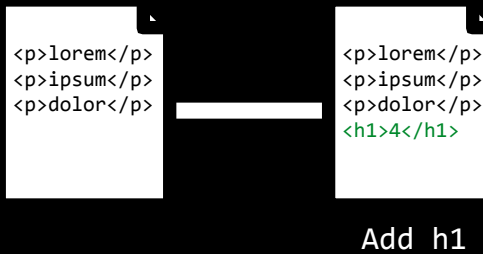
git pull



git pull



git pull



git pull

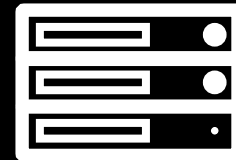
```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

```
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Add h1

Remove lorem



git pull

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
```

```
<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Add h1

```
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>
```

Remove lorem

git log

```
git log
```



```
git log
```



```
git  
log
```

git log



```
commit 436f6d6d6974204d73672048657265
Author: Pablo Monteiro<paablomt@gmail.com>
Date:   Tue Jan 14 14:06:28 2025 -0400
```

Remove lorem

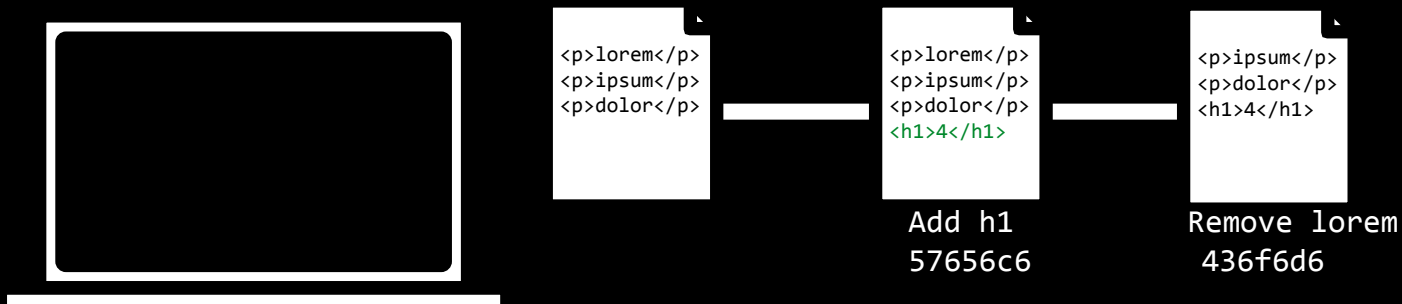
```
commit 57656c636f6d6520746f20576562
Author: Pablo Monteiro<paablomt@gmail.com>
Date:   Tue Jan 14 14:05:28 2025 -0400
```

Add h1

```
git reset
```

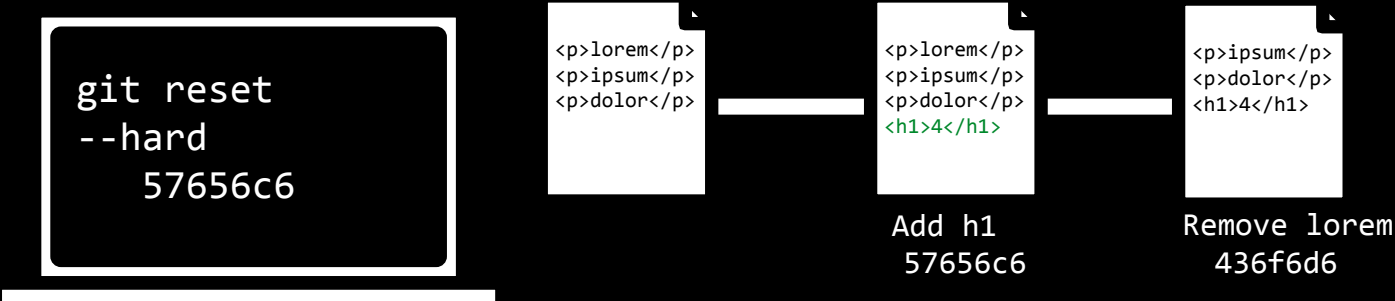
git reset

- git reset --hard
<commit>
- git reset --hard
origin/master



git reset

- git reset --hard
 <commit>
- git reset --hard
 origin/master



The diagram illustrates a git reset operation. On the left, a terminal window shows the command 'git reset --hard 57656c6'. To the right, a sequence of three commit boxes is shown, connected by horizontal lines. The first box contains the text '<p>lorem</p><p>ipsum</p><p>dolor</p>'. The second box, labeled 'Add h1' with hash '57656c6', contains the text '<p>lorem</p><p>ipsum</p><p>dolor</p><h1>4</h1>'. The third box, labeled 'Remove lorem' with hash '436f6d6', contains the text '<p>ipsum</p><p>dolor</p><h1>4</h1>'. The reset operation is shown as moving the head from the second commit back to the first commit.

```
git reset  
--hard  
57656c6
```

<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>

<p>lorem</p>
<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>


Add h1
57656c6

<p>ipsum</p>
<p>dolor</p>
<h1>4</h1>


Remove lorem
436f6d6

git reset

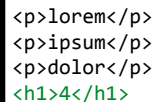
- git reset --hard
 <commit>
- git reset --hard
 origin/master



```
git reset  
--hard  
57656c6
```



```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>
```



```
<p>lorem</p>  
<p>ipsum</p>  
<p>dolor</p>  
<h1>4</h1>
```

Add h1
57656c6

Fazendo alterações



**first
commit**



**first
commit**

alterações



**first
commit**

alterações

**mais
alterações**

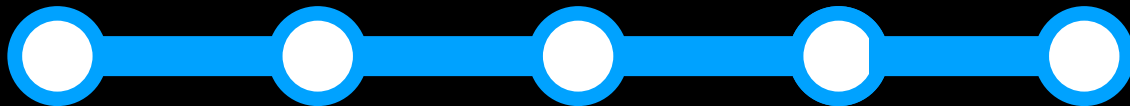


**first
commit**

alterações

**mais
alterações**

**iniciou nova
feature**



**first
commit**

alterações

**mais
alterações**

**iniciou nova
feature**

seguindo com a nova feature

Branching



**first
commit**



**first
commit**

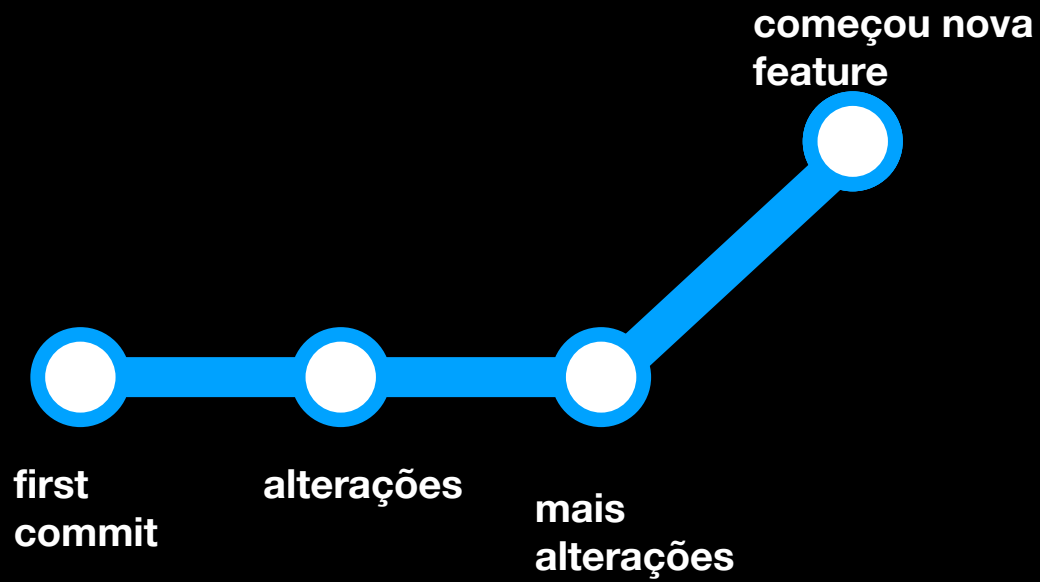
alterações

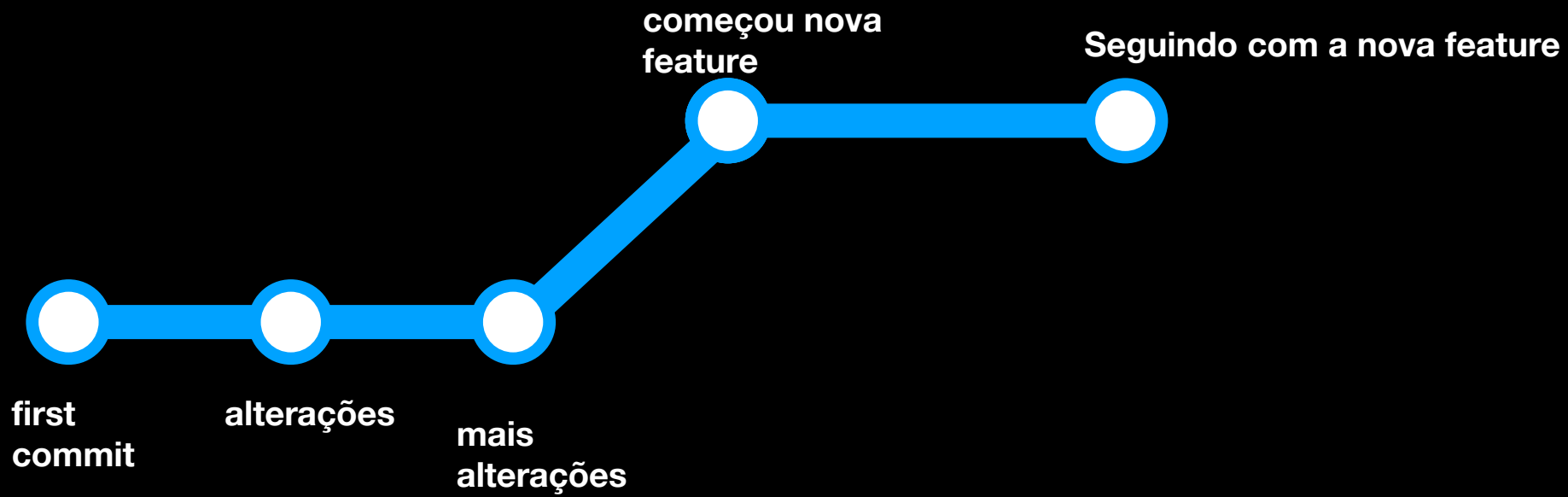


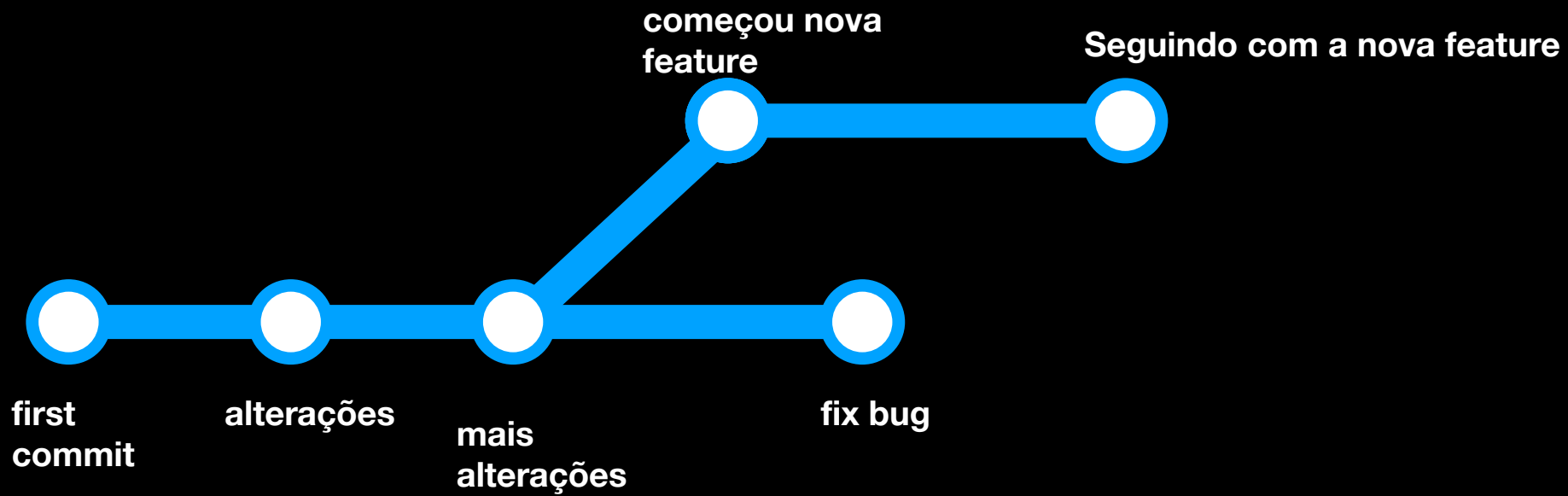
**first
commit**

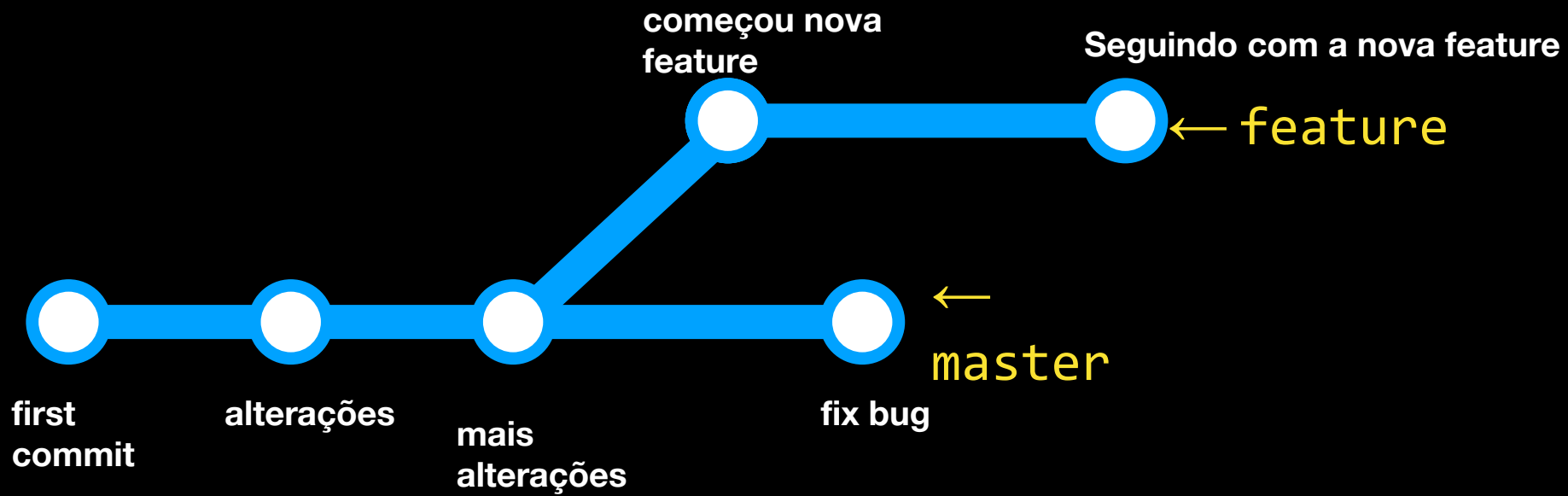
alterações

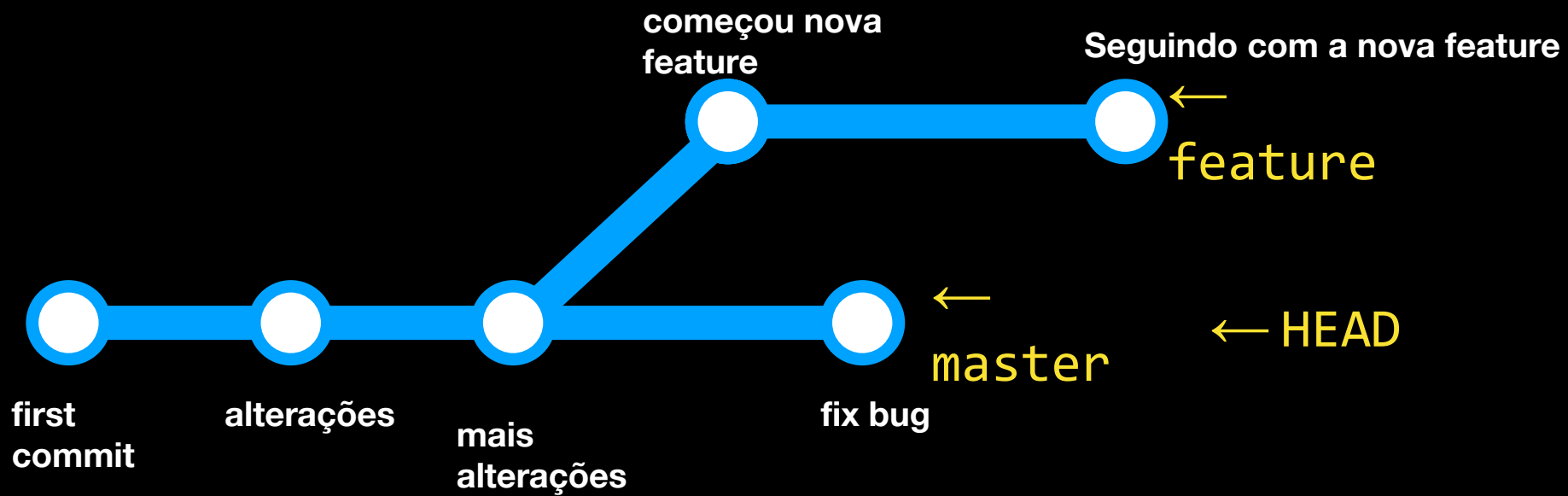
**mais
alterações**

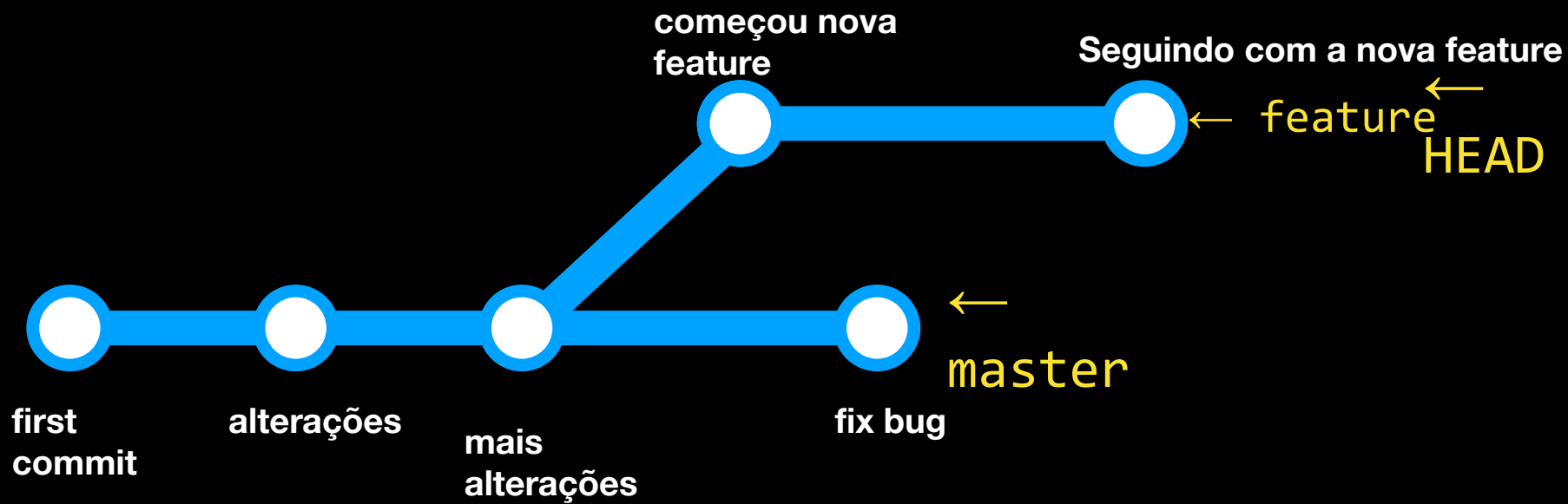












Obrigado !



SOFTEX
PERNAMBUCO

 **Softex**

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO