



**UNIFOR**  
ENSINANDO E APRENDENDO

# Disciplina de Sistemas Operacionais (Aula 5)



## Gerência de Memória

# Apresentação



## **Francisco Nauber Bernardo Gois**

Analista aprendizado de máquina  
no Serviço Federal de Processamento  
de Dados

Doutorando em Informática Aplicada  
Mestre em Informática Aplicada  
Especialista em desenvolvimento WEB

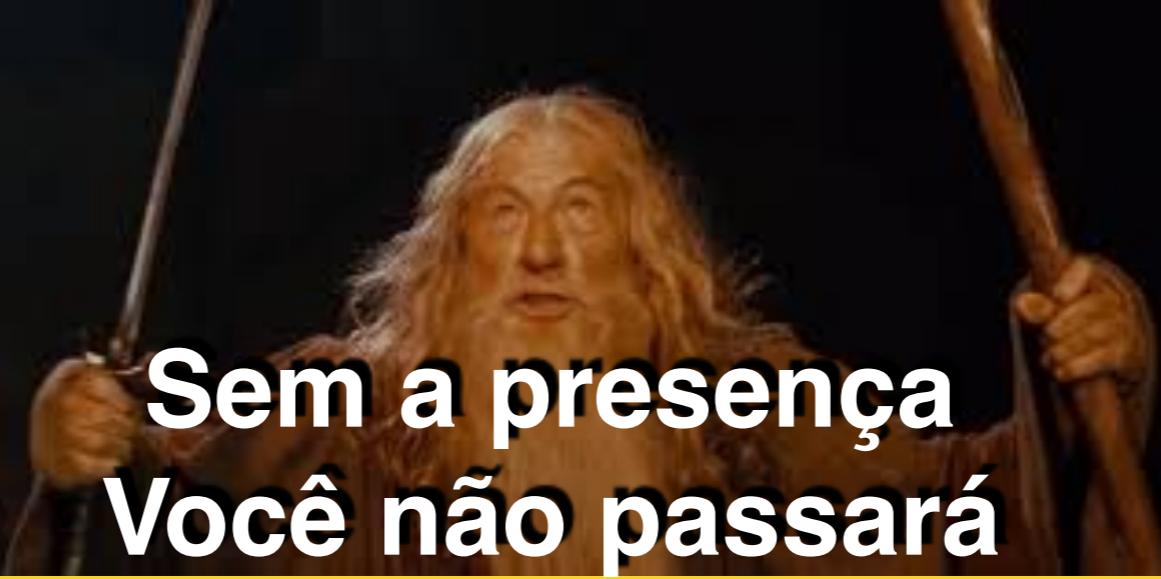
Dúvidas: [naubergois@gmail.com](mailto:naubergois@gmail.com)

**Jovem Padawan  
procure na aula**

**ao telefone  
não falar**



**Procure  
não  
conversar  
durante a  
aula**



**Sem a presença  
Você não passará**

**Para melhor  
desempenho na  
aula**

**Buscar  
aprendizado  
ao invés de  
pontos**

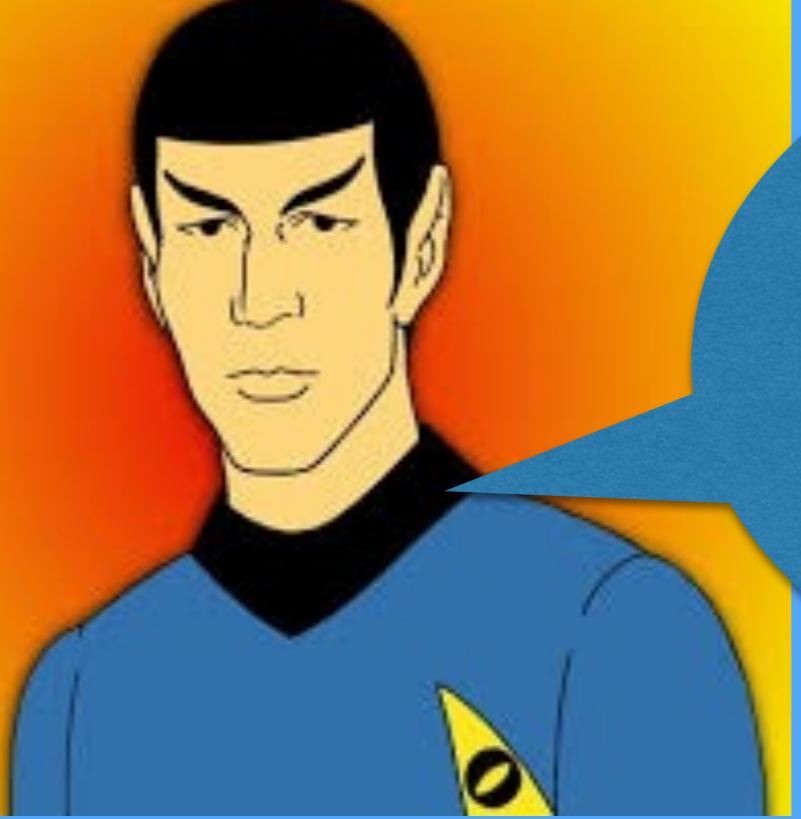


**Não  
teremos  
pontuação fora dos  
trabalhos e provas  
da disciplina**



**Cuidado  
com o  
Horário**





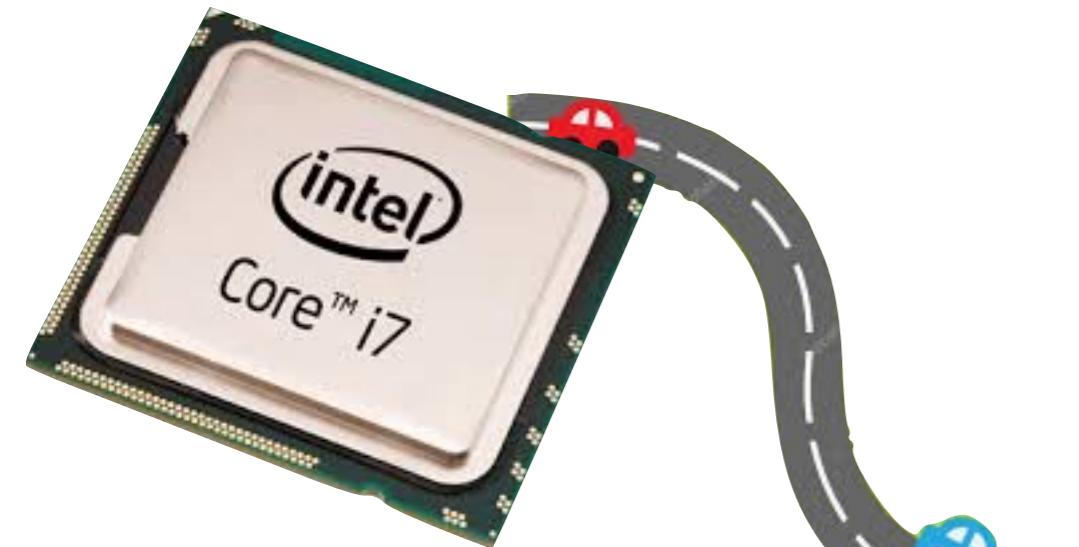
OS trabalhos  
deveram ser  
entregues uma  
semana antes  
da prova

Um cadeira longa  
e prospera

Não teremos  
pontos após a prova  
não adianta pedir



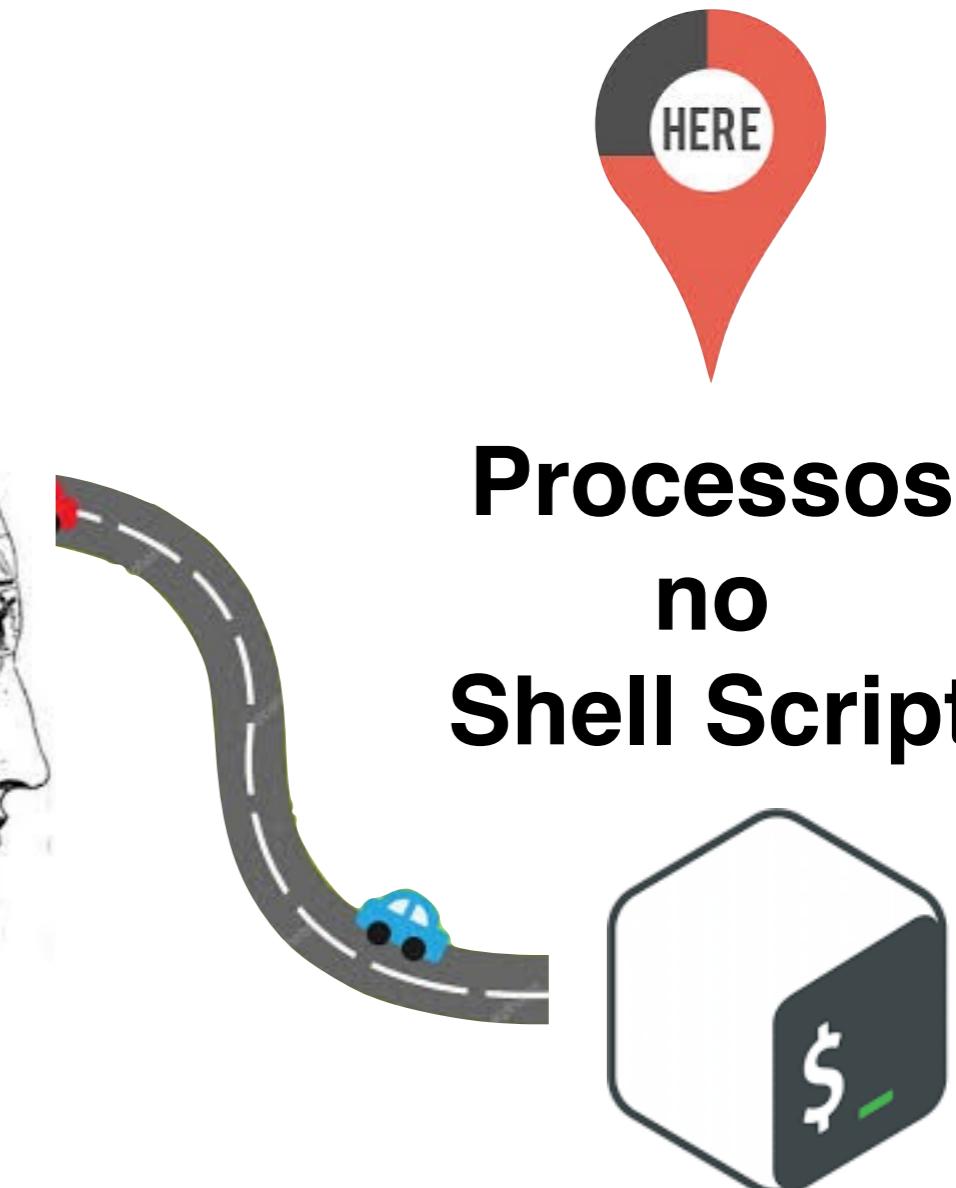
## Cronograma da Disciplina



**Gerência  
de Processador**

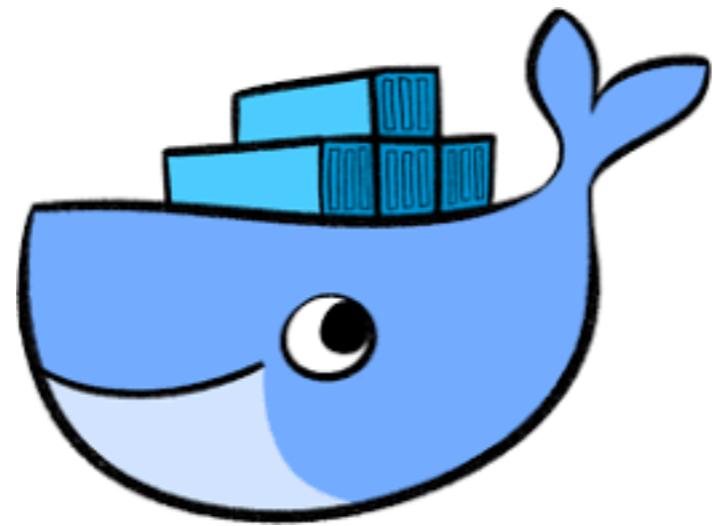


**Gerência de  
Memória**



## Docker com Apache

```
docker run -p 80:80 -dit --name my-app -v "$PWD":/usr/local/apache2/htdocs/ httpd:2.4
```



- `/dev`—Contains device drivers
- `/bin` and `/usr/bin`—Contains standard Linux commands
- `/lib` and `/usr/lib`—Contains standard Linux libraries
- `/var`—Contains configuration and log files
- `/etc`—Contains default configuration files
- `/usr/local/bin`—Contains commands not a part of the distribution, added by your administrator
- `/opt`—Contains commercial software
- `/tmp`—Stores temporary files
- `/sbin` and `/usr/sbin`—Contains system administration commands (`/sbin` stands for “safe” bin)

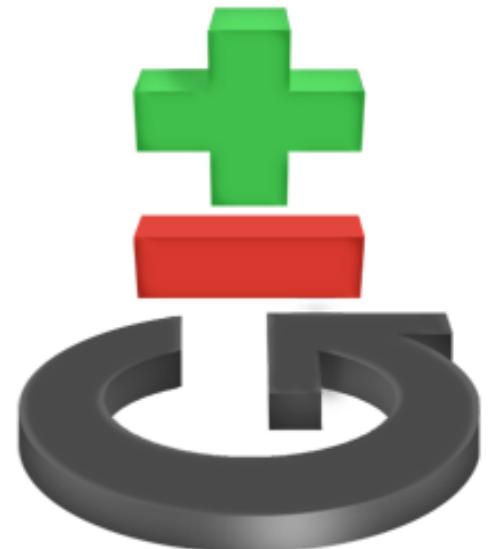
## GITLAB e Devops

"Git é um sistema de controle de versão distribuída, rápido e escalável"

*Tradução do Manual*



Basicamente é um versionador de arquivos, é utilizado principalmente para gerenciar versões de softwares desenvolvidos por um ou mais desenvolvedores, com ele podemos implementar novas funcionalidades e tudo é registrado em um histórico, o qual podemos retroceder sempre que necessário, os integrantes de um projeto podem enviar correções, atualizações, etc. As alterações enviadas para o projeto principal não comprometem o mesmo pois cabe ao dono do projeto a inclusão ou não das alterações efetuadas.



## GITLAB e Devops

O git foi desenvolvido inicialmente por Linus Torvalds (Criador do Linux), pela necessidade de ter um software capaz de controlar a versão do kernel do linux.

Mais informações no link do projeto Git abaixo:

<http://git.or.cz/>



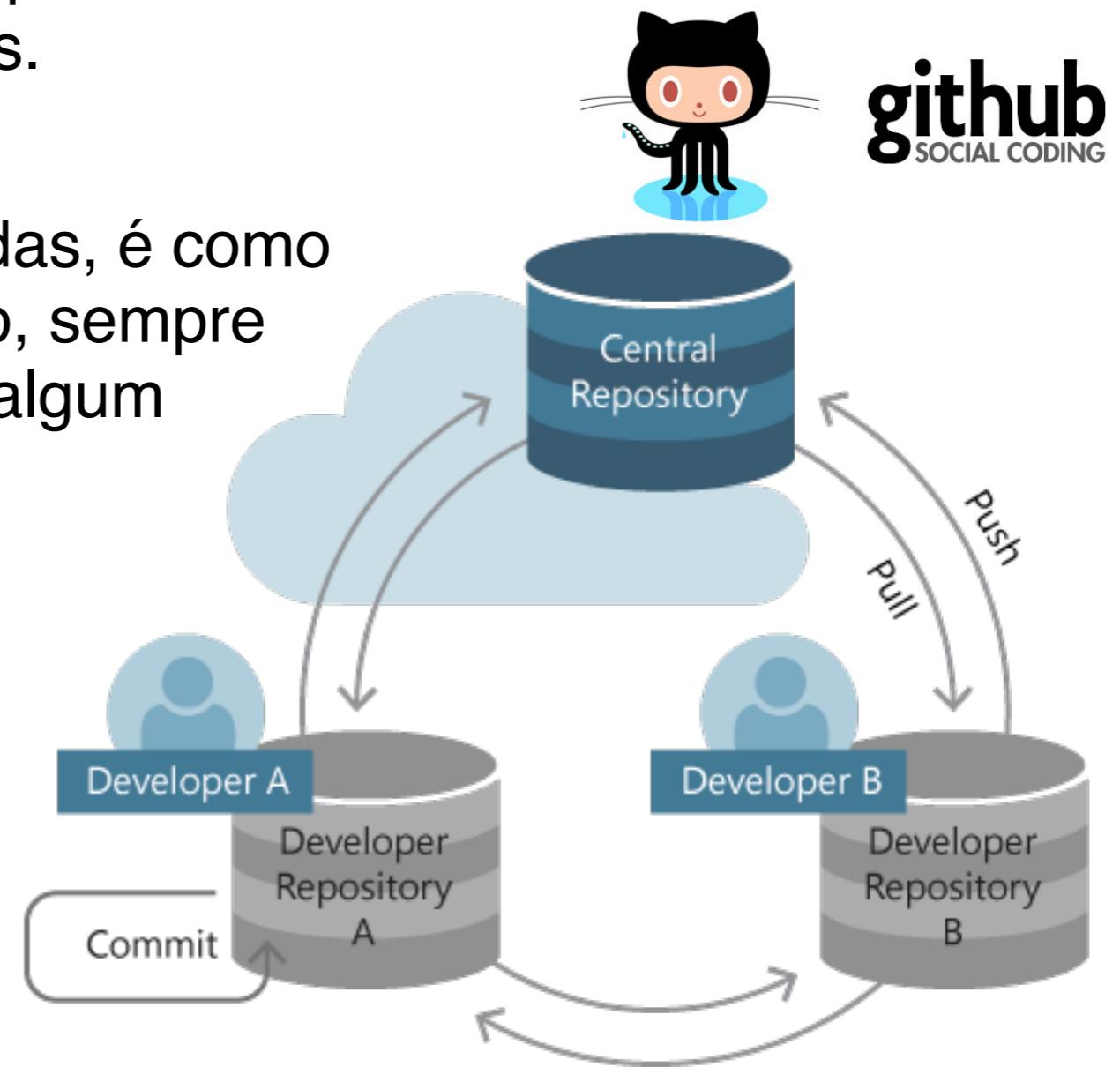
**github**  
SOCIAL CODING



## GITLAB e Devops

Repository: Local onde fica todos os arquivos do projeto, inclusive os históricos e versões.

Commit: Colecão de alterações realizadas, é como se fosse um “checkpoint” do seu projeto, sempre que necessário vc pode retroceder ate algum commit.



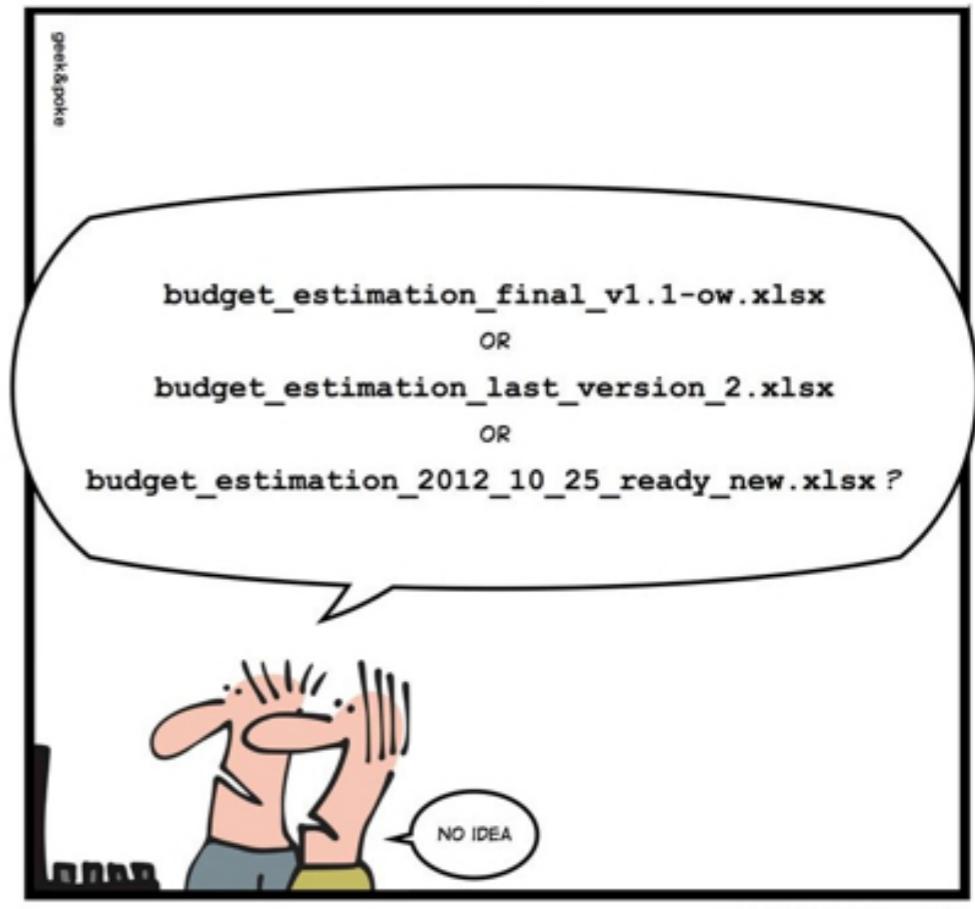
## GITLAB e Devops

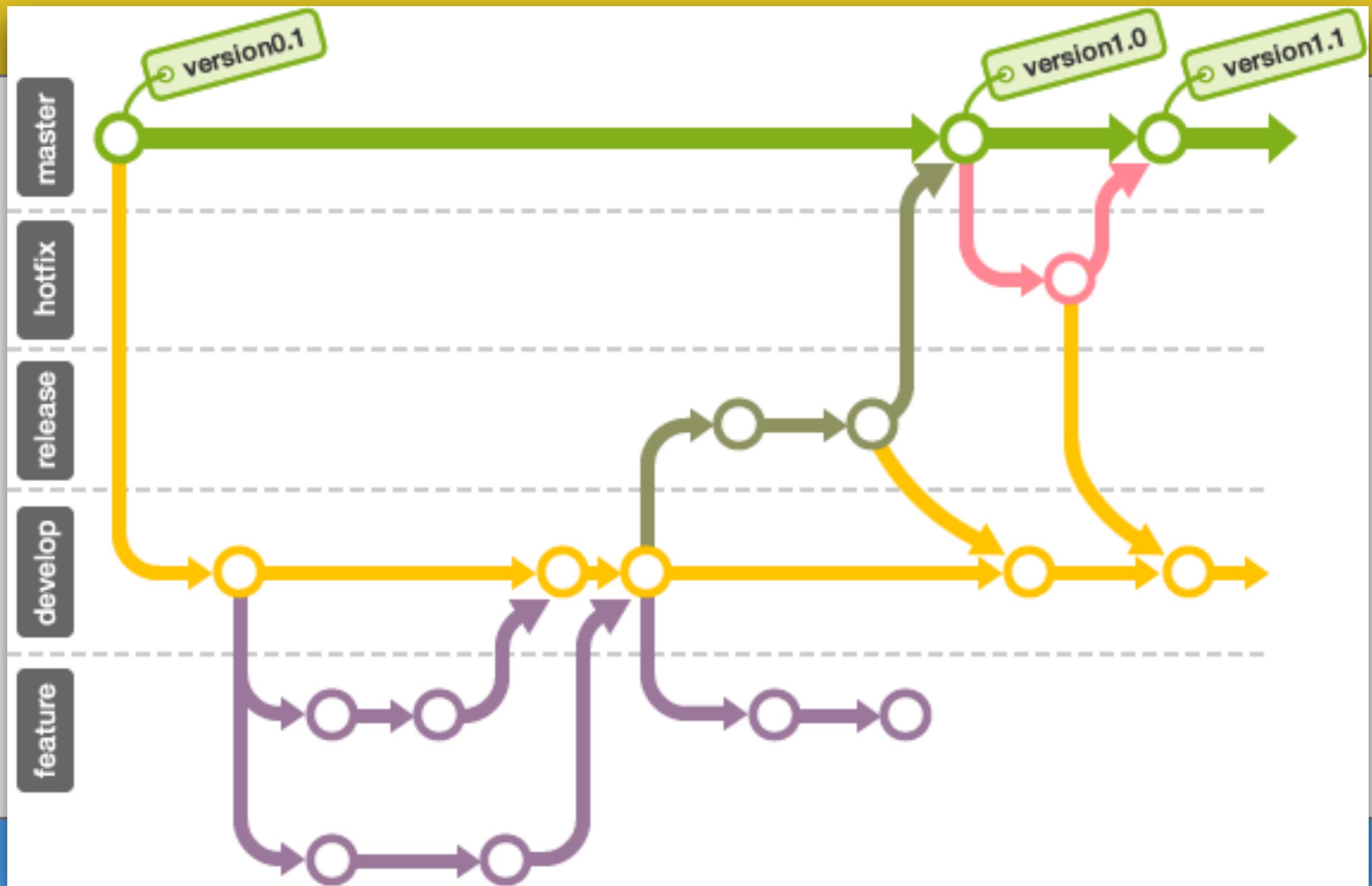
Branch: É uma ramificação do seu projeto, cada branch representa uma versão do seu projeto, e podemos seguir uma linha de desenvolvimento a partir de cada branch.

Fork: Basicamente é uma bifurcação, uma cópia de um projeto existente para seguir em uma nova direção.

Merge: É a capacidade de incorporar alterações do git, onde acontece uma junção dos branchs.

SIMPLY EXPLAINED





## GITLAB e Devops

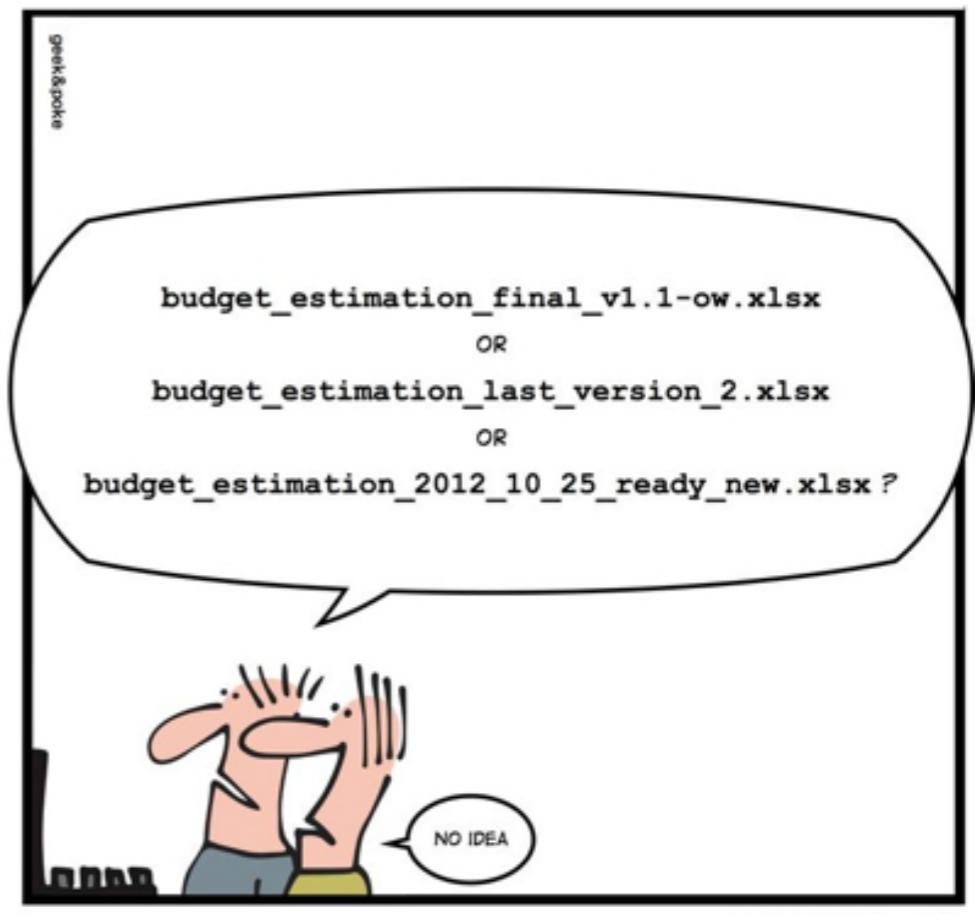
Primeiro vamos criar um novo diretório e em seguida iniciar nosso repository git:

```
mkdir NovoProjeto
```

```
cd NovoProjeto
```

```
git init
```

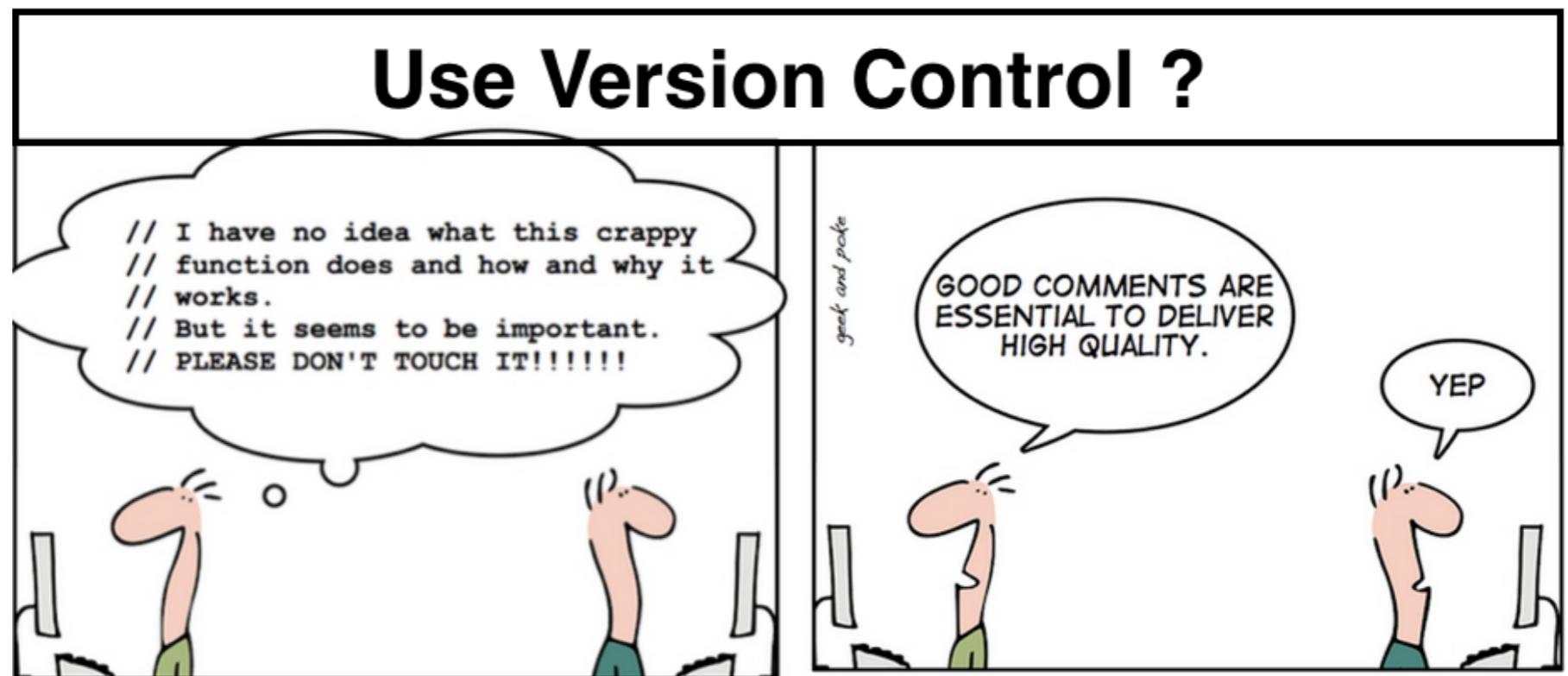
*Todos os arquivos dentro do diretório NovoProjeto faz parte do nosso repository*



VERSION CONTROL

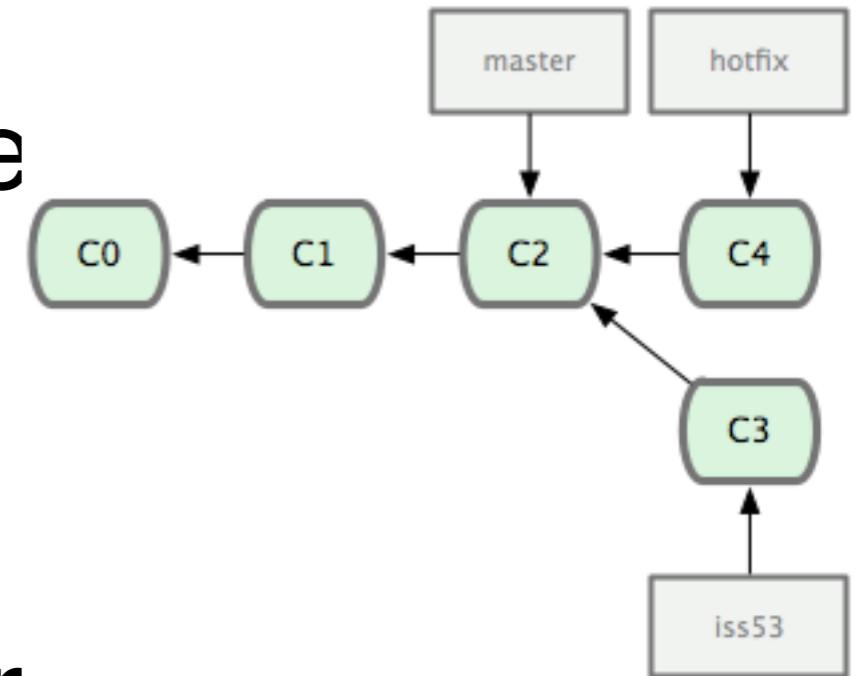
## GITLAB e Devops

```
git config user.name "Joao"  
git config user.email "email@email.com"
```



## GITLAB e Devops

O git é inteligente o suficiente para apenas realizar commits se detectar alguma alteração nos arquivos que foram indicados pelo (git add), dessa forma o único commit que ele realiza será alguma alteração é o primeiro.

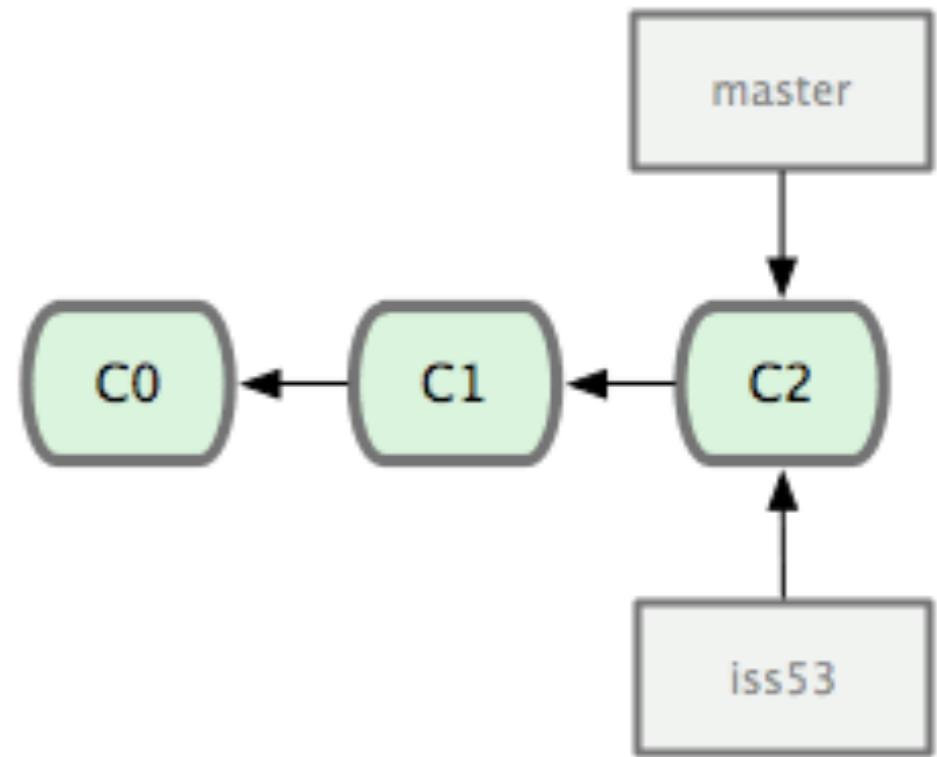


## GITLAB e Devops

`git commit -a -m "Nosso primeiro commit"`

Passamos o parâmetro `-a` para informar que todos os arquivos alterados devem ser incluídos do commit.

Passamos o parâmetro `-m` para adicionar uma mensagem explicativa para o commit.



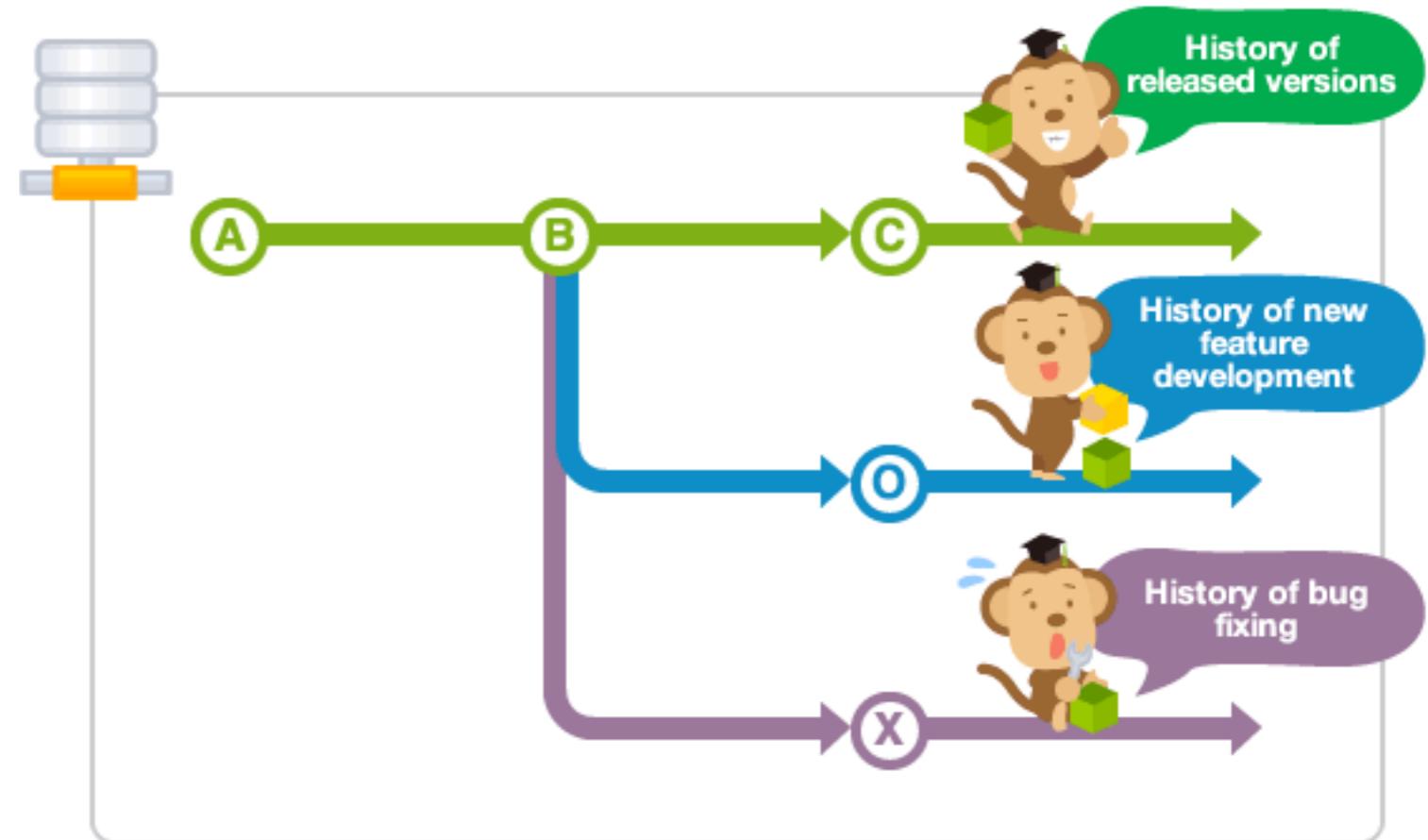
## GITLAB e Devops

Depois que fizemos nosso primeiro commit existem alguns comandos que podemos utilizar para verificar o mesmo, são eles:

`git log` // listar todos os commits já realizados.

`git status` // mostrar os arquivos alterados desde o último commit.

`git show` // mostrar as alterações realizadas no último commit.



## GITLAB e Devops

Todo novo repositorio git tem um branch default chamado “master”.

Para listar todos nossos branchs:

git branch

Se vc não tiver mais branchs a saida será:

\*master



## GITLAB e Devops



# GitLab

- OpenSource
- 700 desenvolvedores voluntários



IDEA



DEFINE



PLAN



CREATE



REVIEW

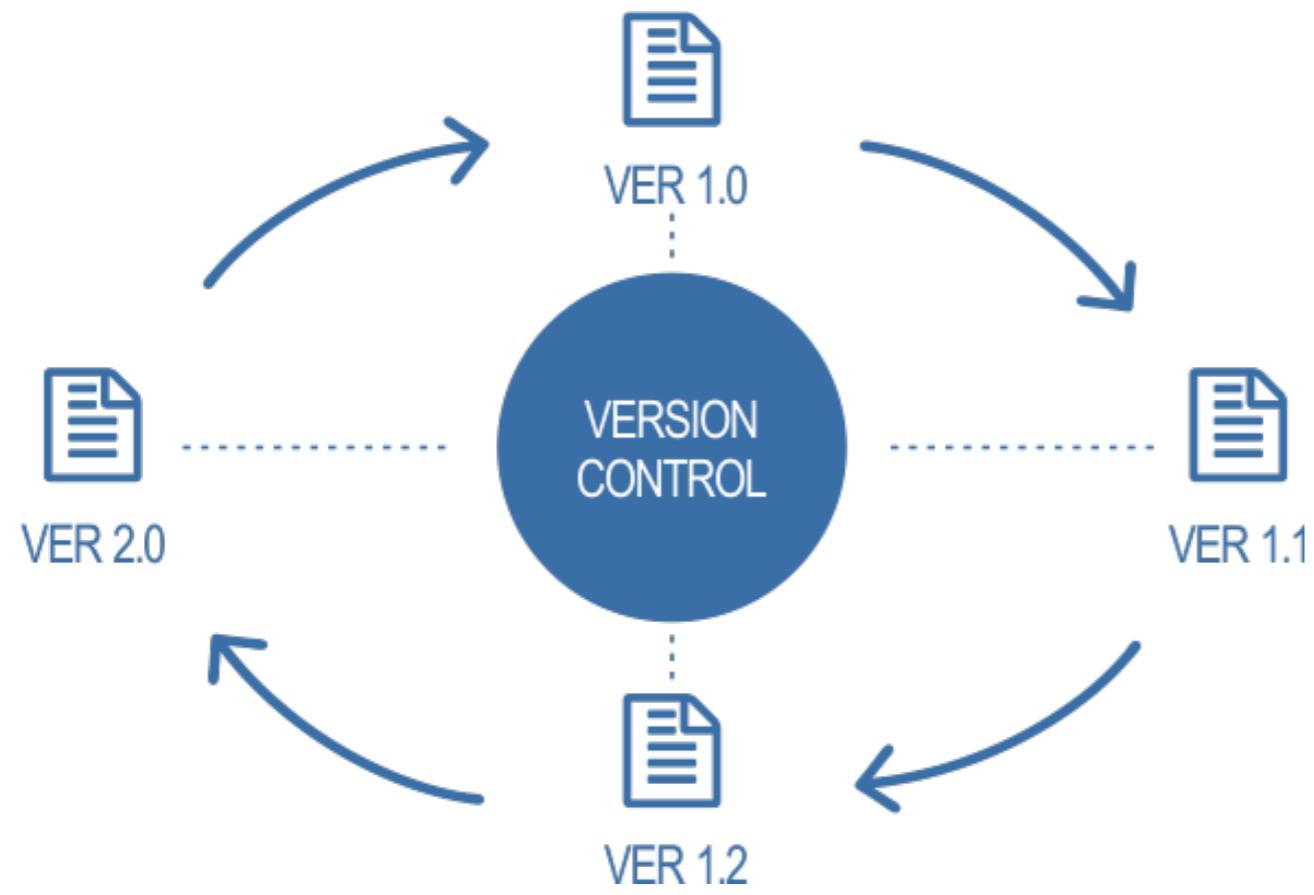


RELEASE

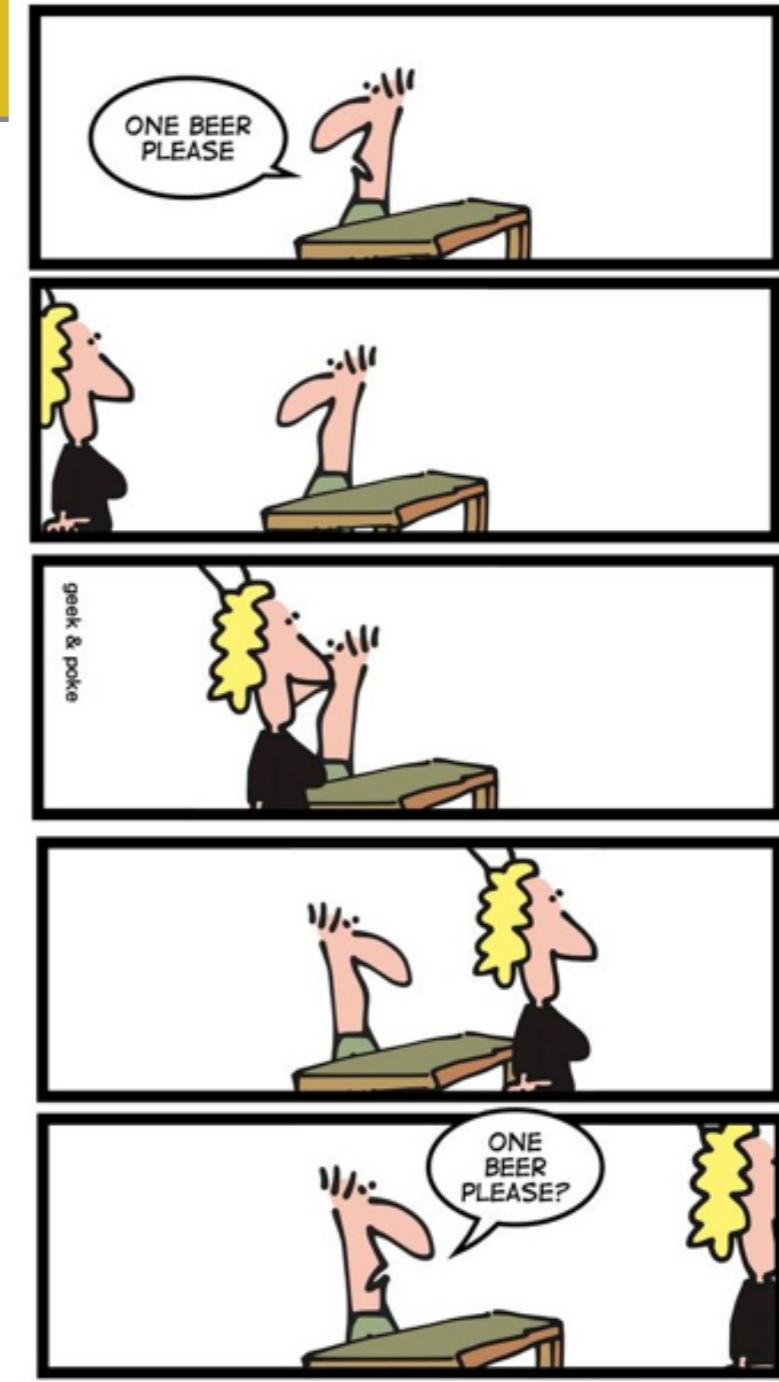


MEASURE

## GITLAB e Devops



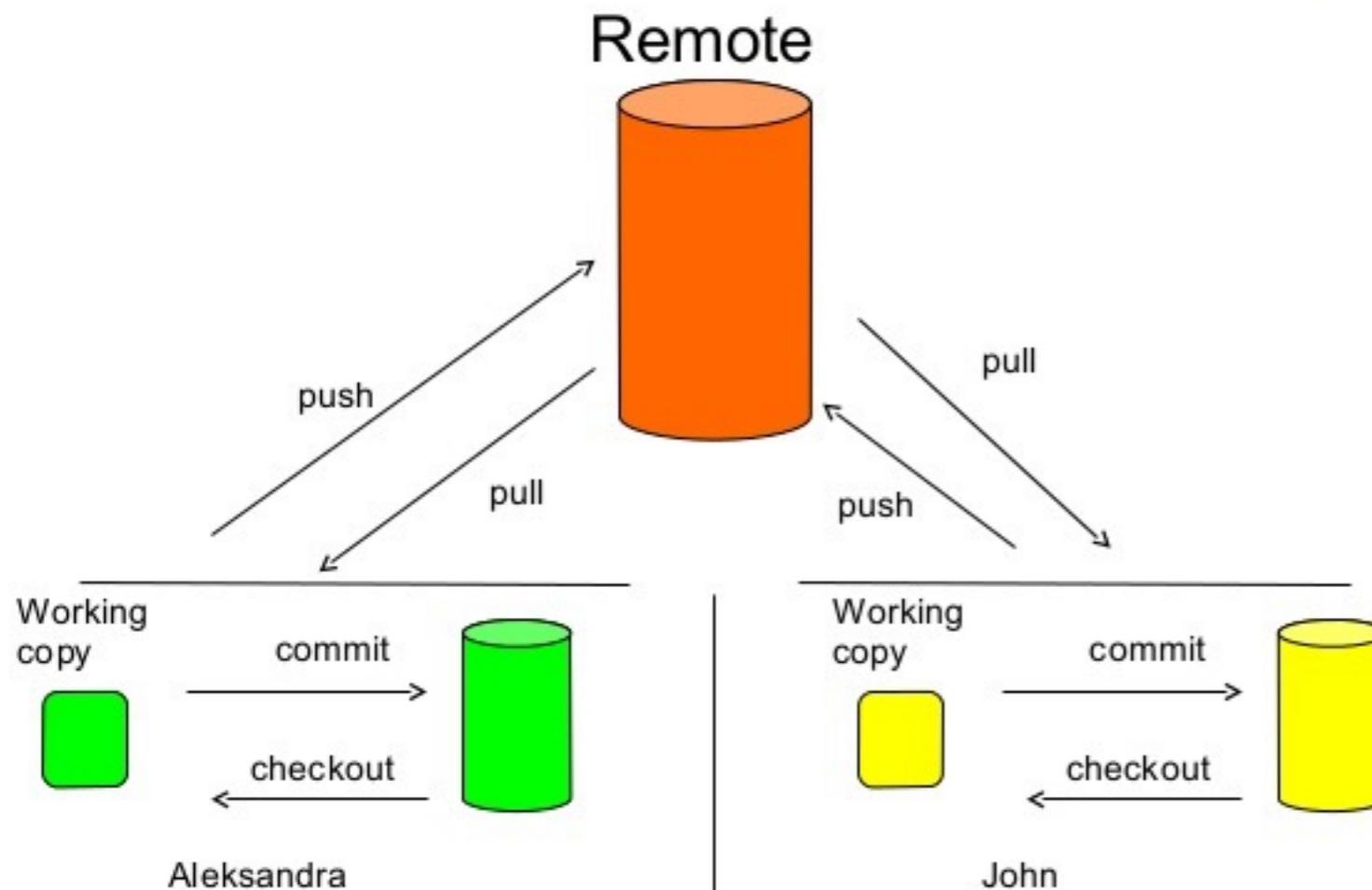
SIMPLY EXPLAINED



## GITLAB e Devops

Working with remote repository

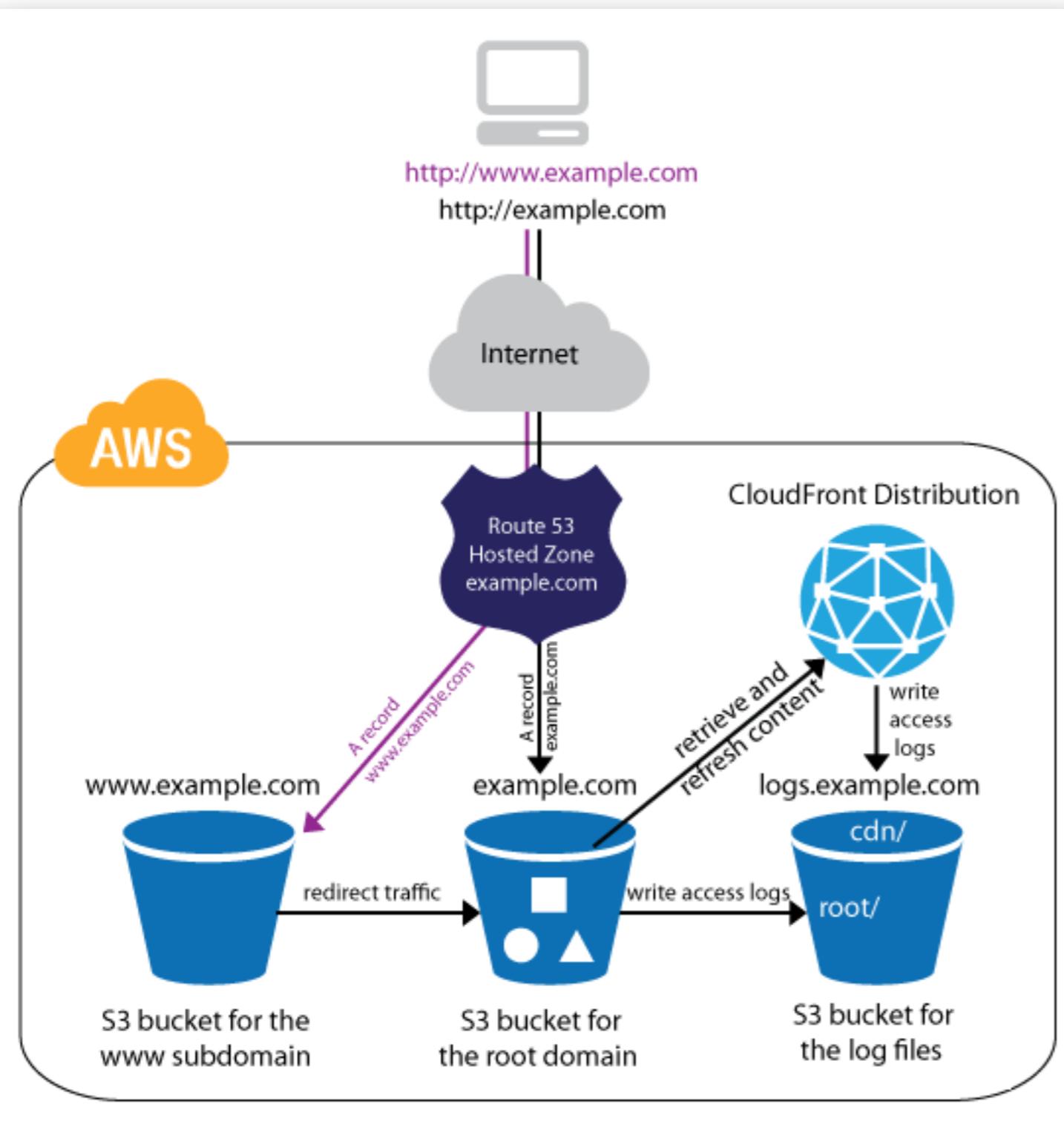
software carpentry



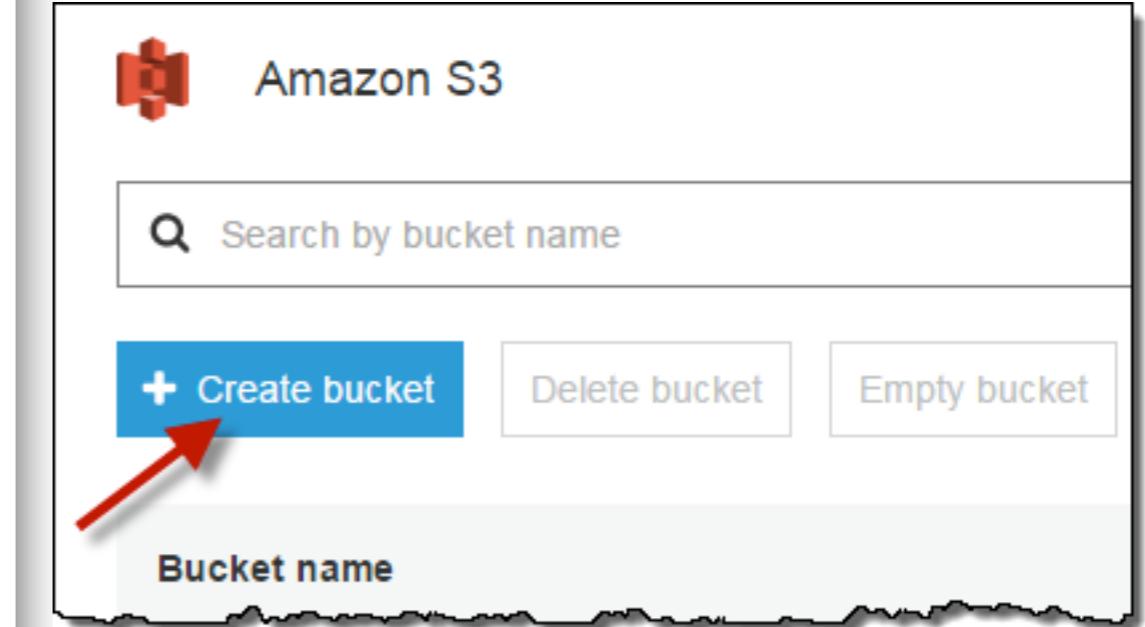
<http://software-carpentry.org>



## Site estático no Amazon AWS



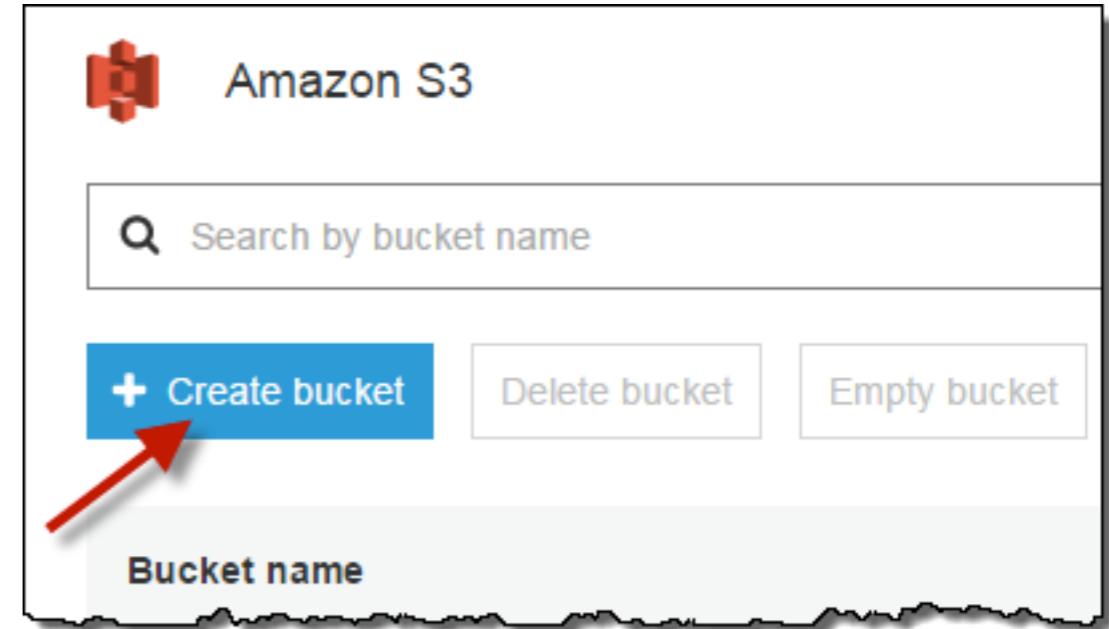
### Criando um bucket



## Site estático no Amazon AWS

**Buckets are globally unique containers for everything that you store in Amazon S3.**

### Criando um bucket



## Site eletônico no Amazon AWS

**freenom**  
A Name for Everyone



Services

Login

Email Address:

Password:

Remember Me

**Login**

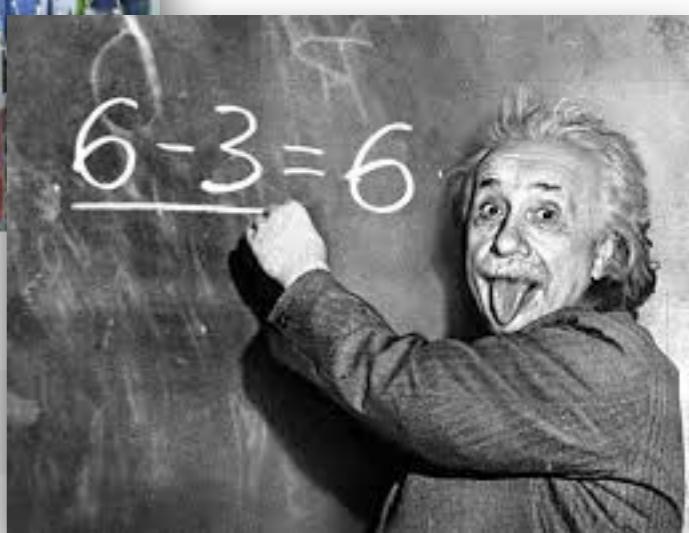
Request a Password Reset

---

Use social sign in

**Sign in as Nauber Gois**

[switch accounts](#)



# Site estático no Amazon AWS

## Bucket criado

The screenshot shows the AWS S3 console interface. At the top left is the Amazon S3 logo. On the right, there is a blue button labeled "Discover the new console". Below the logo is a search bar with the placeholder "Search for buckets". Underneath the search bar are three buttons: "+ Create bucket" (blue), "Delete bucket" (gray), and "Empty bucket" (gray). To the right of these buttons, it says "1 Buckets" and "1 Re". The main area displays a table with three columns: "Bucket name", "Region", and "Date created". The first row shows a bucket named "nauber.ml" in the "US West (Oregon)" region, created on "Sep 19, 2017 12:22:03 PM".

Bucket name	Region	Date created
nauber.ml	US West (Oregon)	Sep 19, 2017 12:22:03 PM

## Site estático no Amazon AWS

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
    <h1>Welcome to my website</h1>
    <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

+ Create bucket    Delete bucket    Empty bucket

Bucket name ↑☰

- admin-created
- admin-created-one

## Guia de uso do S3 no Amazon AWS

Vamos realizar o upload do conteúdo

The screenshot shows the AWS S3 console interface. On the left, a 'Pictures library' window displays a list of files and folders: Sample Pictures, cats, cool\_monkey, dogs, and example-pic. The 'cool\_monkey' folder is selected and highlighted with a red box. A large red arrow points from this selection towards the S3 upload interface on the right. The S3 interface has tabs for Objects, Properties, Permissions, and Management. It includes a search bar, an 'Upload' button, and a 'Create folder' button. Below these are buttons for 'All' and 'Deleted objects'. The main area lists objects with columns for Name, Last modified, and Type. The 'cool\_monkey' folder is listed under 'Name' with a last modified date of Mar 13, 2017. The 'example-pic.jpg' file is also listed under 'Name' with a last modified date of Mar 30, 2017.

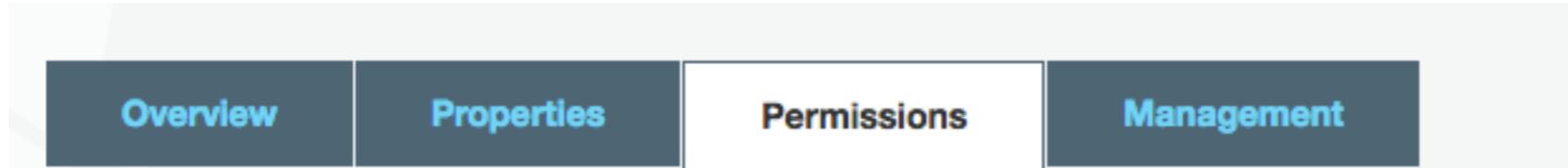
Name	Last modified	Type
cool_monkey	Mar 13, 2017	Folder
example-pic.jpg	Mar 30, 2017	Image

## Site estático no Amazon AWS

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {"Sid": "PublicReadGetObject",  
     "Effect": "Allow",  
     "Principal": "*",  
     "Action": ["s3:GetObject"],  
     "Resource": ["arn:aws:s3:::example.com/*"]  
    }  
  ]  
}
```

**Attach the following bucket policy to the example.com bucket substituting the name of your bucket for example.com. For step-by-step instructions to attach a bucket policy, see How Do I Add an S3 Bucket Policy? in the Amazon Simple Storage Service Console User Guide.**

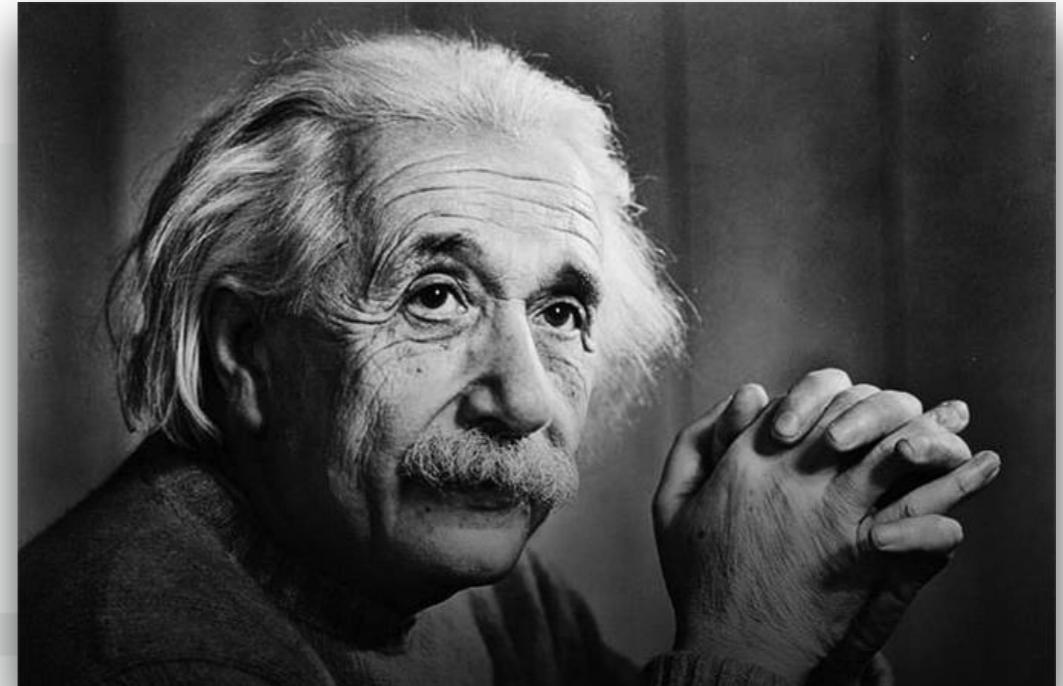
# Site estático no Amazon AWS



## Adicionando Permissões

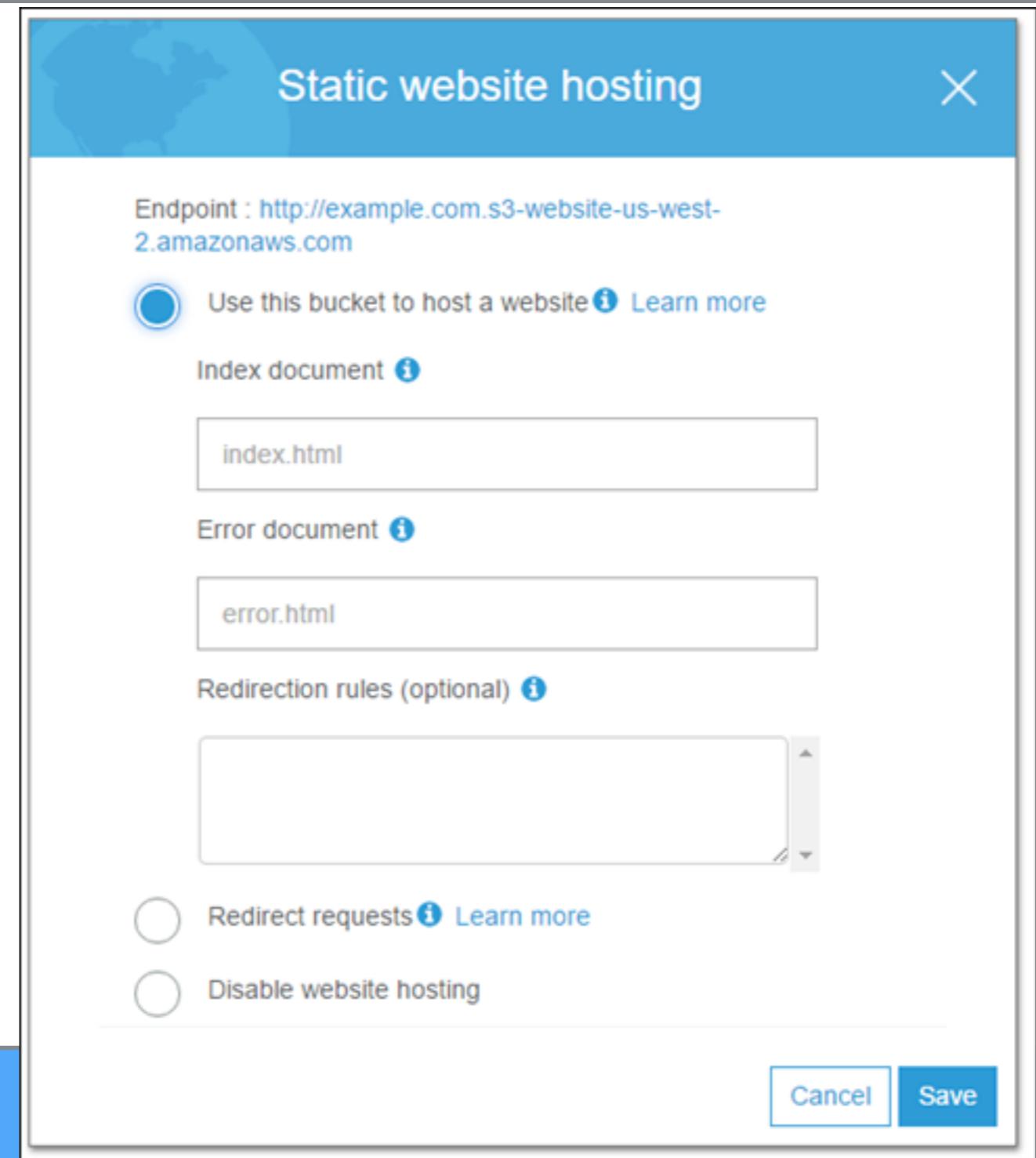
Bucket policy editor ARN: arn:aws:s3:::nauber.ml  
Type to add a new policy or edit an existing policy in the text area below.

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {"Sid": "PublicReadGetObject",
5       "Effect": "Allow",
6       "Principal": "*",
7       "Action": ["s3:GetObject"],
8       "Resource": ["arn:aws:s3:::nauber.ml/*"]
9     }
10   }
11 ]
12 }
```



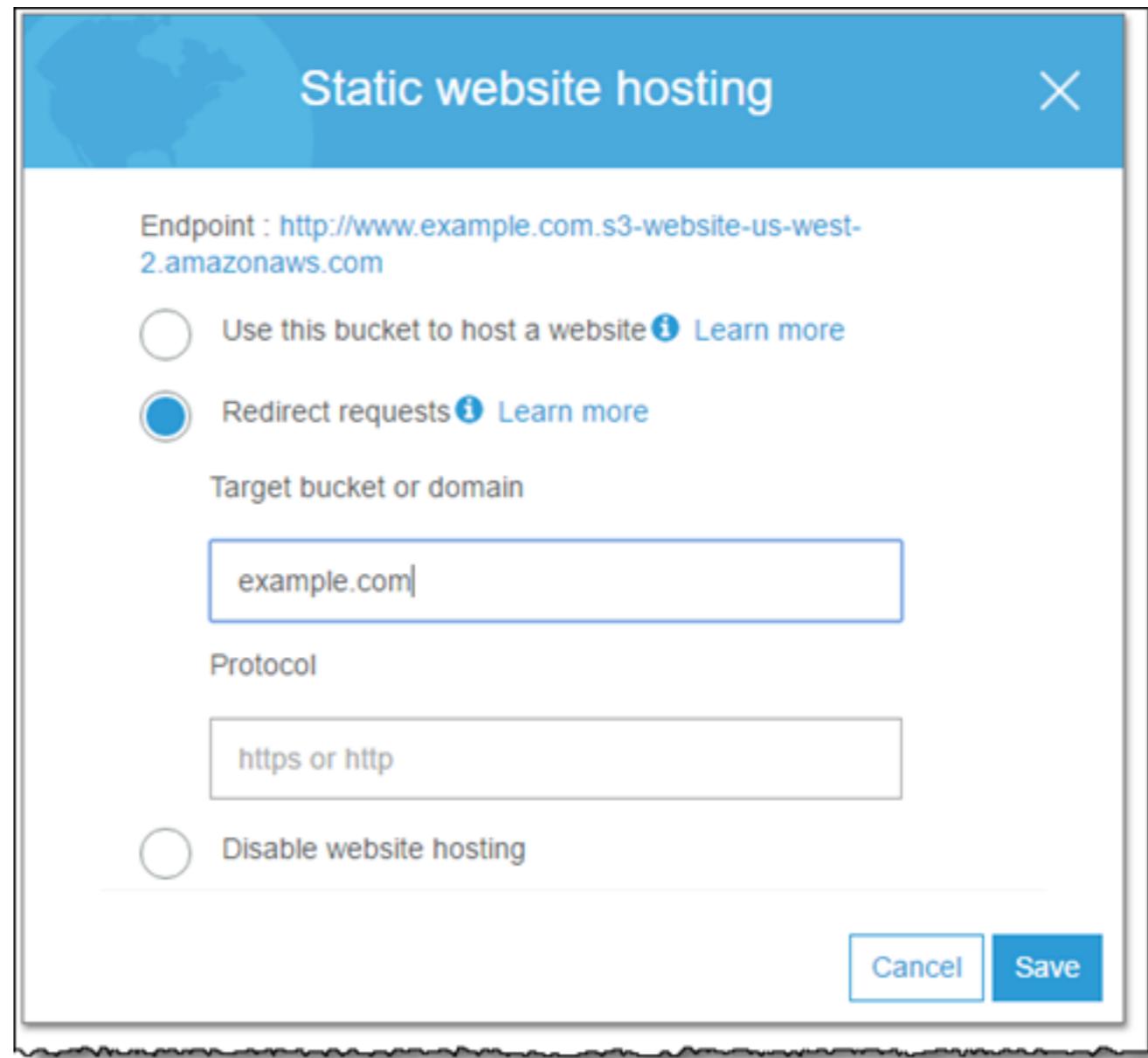
# Site estático no Amazon AWS

## Configurando website estático



# Site estático no Amazon AWS

Configurando  
website  
estático



## Comando Printf

```
$ printf "%d\n" 5  
5  
$ printf "%f\n" 5  
5.000000
```



## namei

The `namei` (*name inode*) command lists all the components in a path, including any symbolic links.

```
$ namei files
f: files
l files -> /home/ken/bash/scripts
d /
```



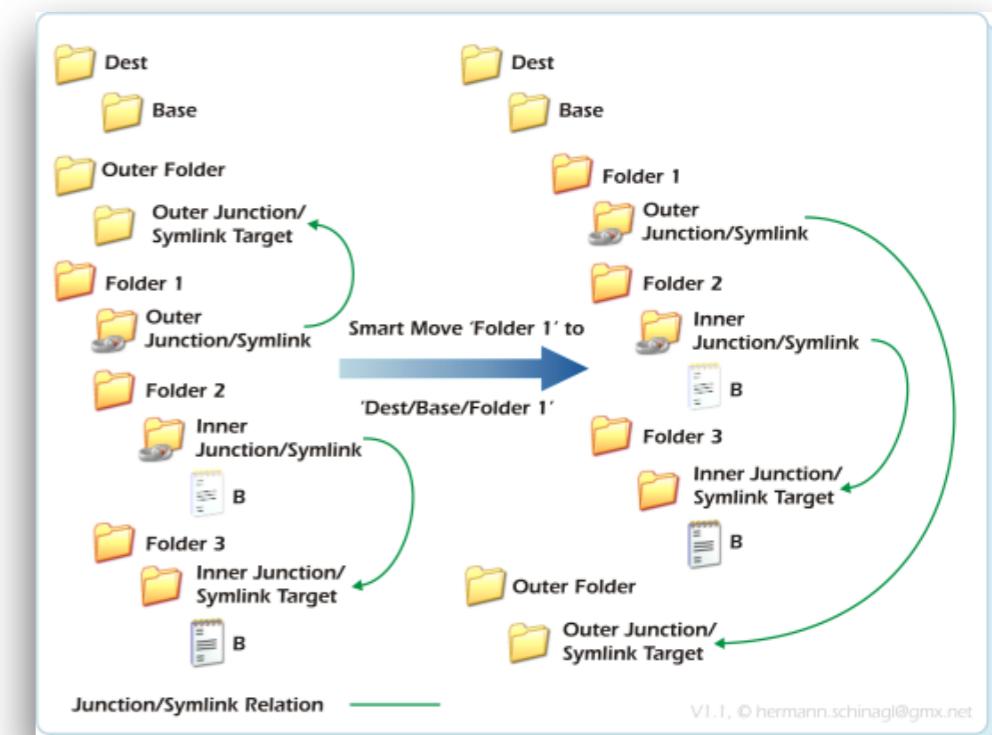
## Links Simbólicos

### LINK SIMBÓLICO E HARDLINK

O link é um mecanismo que faz referência a outro arquivo ou diretório em outra localização. Os links são arquivos especiais e podem ser identificados com um "l" quando executado o comando: "ls -la".

Exemplo:

```
$ ls -la  
lrwxr-xr-- 1 roberto roberto 5 2006-10-12 22:40 link ->  
teste.txt
```



## Links Simbólicos

### In - Linkando arquivos

`In -s <arquivo_origem> <link simbólico>`

Este comando é usado para gerar links simbólicos, ou seja, que se comportam como um arquivo ou diretório, mas são apenas redirecionadores que mandam seu comando para outro arquivo ou diretório, por exemplo:

```
$ In -s /manual /home/linux-manual
```

Este comando criará o link `/home/linux-manual`, se você der um `ls -l` você verá que o diretório `/home/linux-manual` está apontando para `/manual`. Se você ir para o `/home/linux-manual`, você na verdade estará no `/manual`, mas como é um link, não há diferença.



## Links Simbólicos

### TIPO SIMBÓLICO

No link tipo simbólico, o link é um arquivo especial de disco do tipo link, que tem como conteúdo o caminho para chegar até o arquivo alvo.

Características:

Pode-se fazer links simbólicos em arquivos e diretórios;

O link simbólico e o arquivo alvo não precisam estar na mesma partição de disco;

Se o link simbólico for apagado/movido.

Somente o link será apagado/movido;

Qualquer usuário pode criar/desfazer um link simbólico (respeitando as permissões).

### TIPO HARLINK

No link tipo hardlink, o link é apontado para o mesmo inode do arquivo alvo, sendo assim, os dois arquivos serão o mesmo.

Características:

Não é possível fazer um hardlink para um diretório;

Somente é possível fazer hardlink em arquivos que estejam em uma mesma partição de disco;

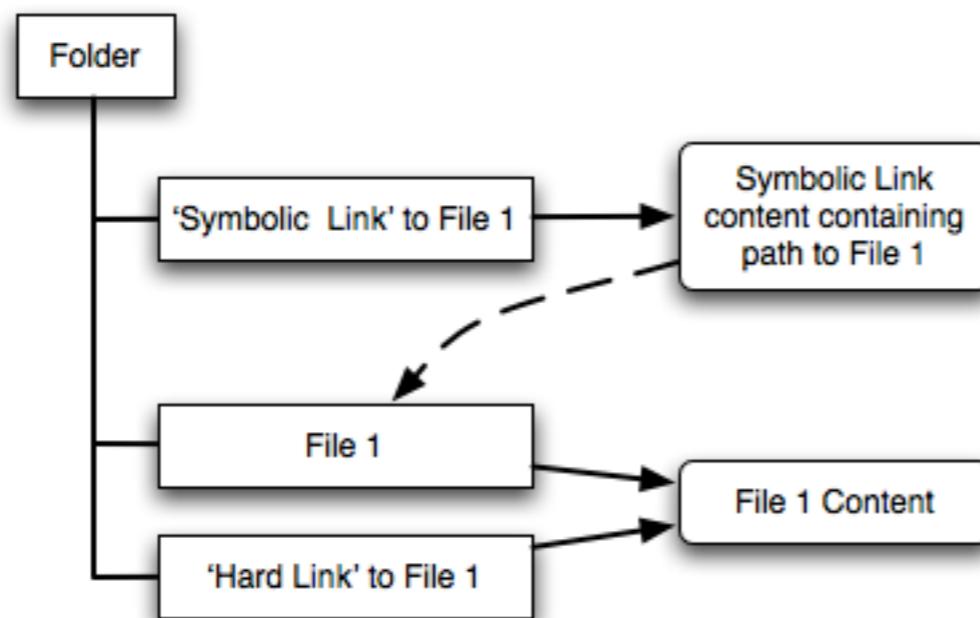
Se o hardlink for apagado/movido, você estará apagando/movendo o arquivo alvo;

Somente o usuário root pode criar/desfazer hardlinks.

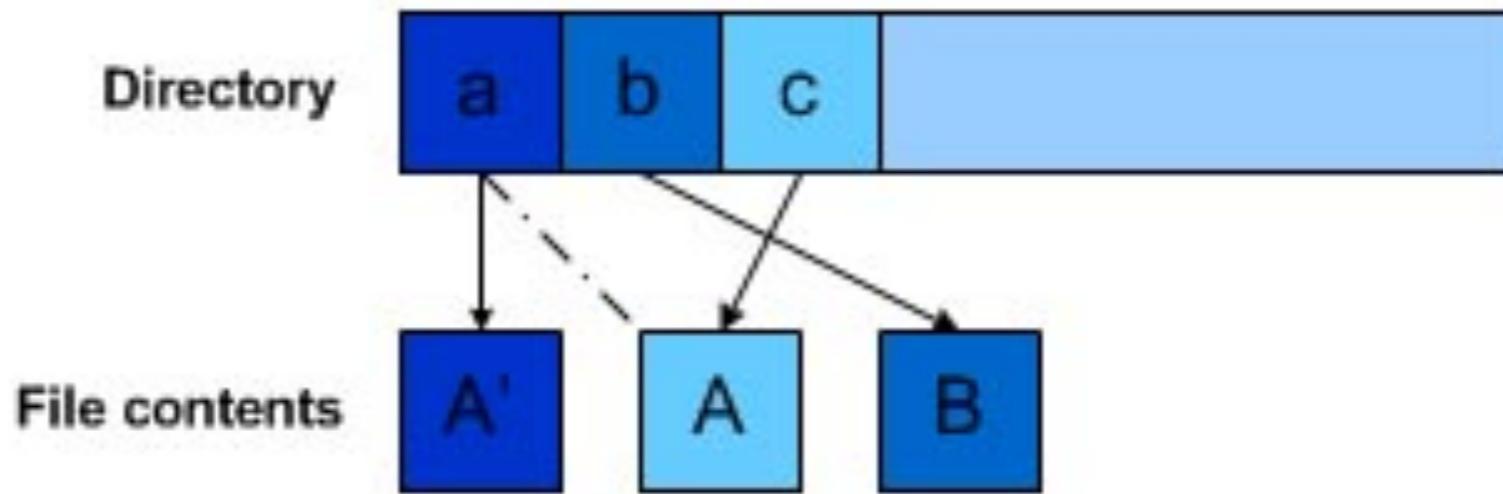
## namei

The `namei` (*name inode*) command lists all the components in a path, including any symbolic links.

```
$ namei files
f: files
l files -> /home/ken/bash/scripts
d /
```

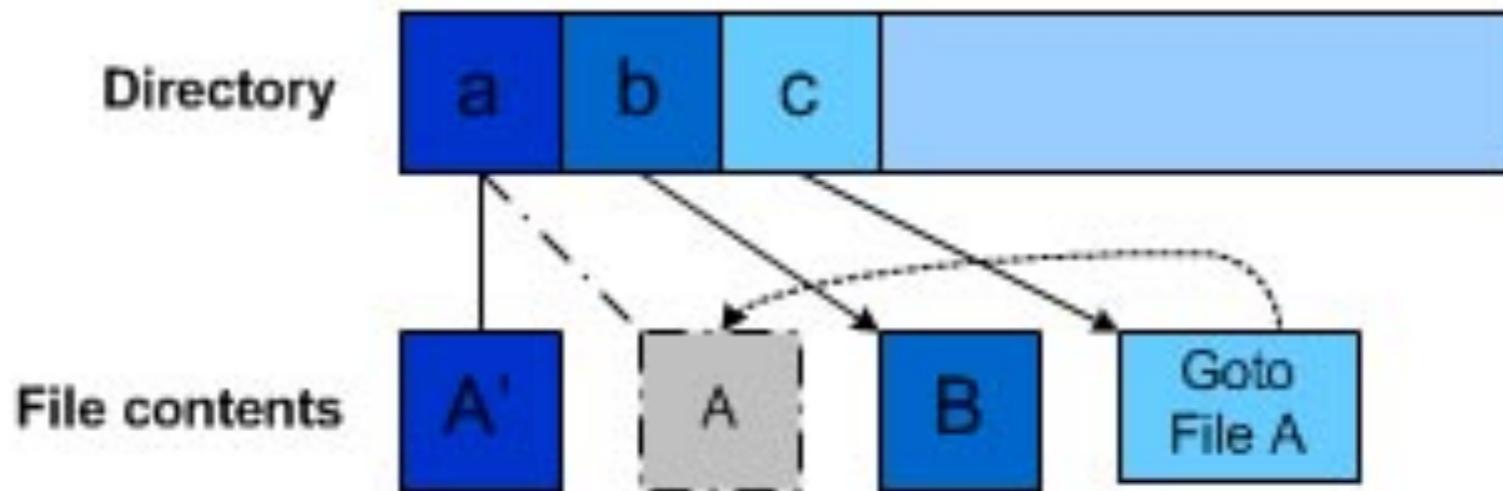


## Hard Links

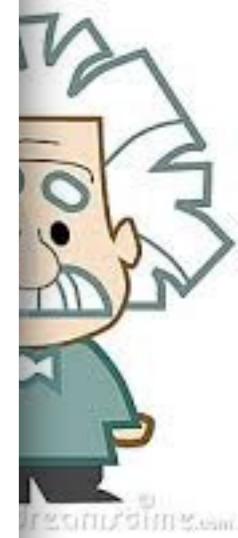


Deleting filename "a" doesn't delete file  
A if c is also hard-linked to it.

## Symlinks



After deleting A , the symbolic link through filename "c"  
is no longer available -> dangling-link problem



## ALias

## Shell Aliases

An *alias* is a short form of a command. The built-in alias command creates simple abbreviations for the current Bash session.

To create an alias, use the alias command to assign a command and its switches a name.

```
$ alias lf='ls -qFl'  
$ lf  
-rw-r----- 1 kburtsch devgroup 10809 Apr  6 11:00 assets.txt  
-rw-r----- 1 kburtsch devgroup 4713 Mar  9 2000 mailing_list.txt
```

Typing the alias command by itself, or with the -p switch, lists the current aliases.

```
$ alias  
alias lf='ls -qFl'
```

Bash interprets an alias only once, allowing the aliasing of a command with its own name.

```
$ alias ls='ls -qF' # Bash isn't confused
```





**Francisco Nauber Bernardo Gois**  
**Email: naubergois@gmail.com**