

HACK ATHION

The logo for 'idp' is a bright green circle with the lowercase letters 'idp' in black. It is surrounded by several concentric green circles of varying thicknesses, some of which have small dots at their ends, giving it a dynamic, digital feel.

BEM VINDOS!

Hack [programação] + Thon [maratona]



Informações Gerais

8 equipes

28 horas totais

6 mentores

28 estudantes

24 horas programando

3 refeições

0 horas de sono

Programação

1º Dia 25/5	2º Dia 26/5	3º Dia 27/5
18:00 - 18:15 Credenciamento ☐	18:00 - 20:00 Maratona* 💻	07:00 - 09:00 Café da Manhã ☕
18:15 - 18:30 Abertura	20:00 - 21:00 Jantar 🍔	08:00 - 12:00 Maratona* 💻
18:30 - 19:30 Workshop	21:00 > 😓 Maratona 💻	12:00 - 13:00 Almoço 🍽️
19:30 - 20:00 Coffee Break ☕		14:00 - 17:00 Maratona* 💻
20:00 - 21:00 Reunião de alinhamento das equipes		17:00 - 18:00 Demo 🚩
		18:30 > Resultados e premiações 🏆

*Horários com mentores disponíveis

Escala Mentoria

26/5 18h-21h	27/5 09h-12h	27/5 15h-18h
Jeremias	Daniel	Jeremias
Nilson	Eduardo	Álvaro
Álvaro	Jeremias	Lucas Rodrigues
		Eduardo

Equipes

Byteiros

Augusto Soares
Emerson Vitor
Matheus Antonio
Thiago Gough

Computaria

Gabriel Magioli
Luan Ludry Souza
Lucas Rocha
João Marcelo

Garotos de Programa

Petrus de Oliveira
Vitor Souza
João Gabriel Macedo
Tiago Chehab

Loading

Rafael Cândido
Guilherme Abade
Igor Max
Luiz Eduardo

Marpi

Pedro Imanichi
Filipe Marra
Kauã Leão
Caleb

Os Conquistadores da Disrupção

João Vitor Martins
João Victor Távora
Lucas da Silva Rocha
Rafael Nogueira

Trem das Onze

Rafael Manggini
Amanda Pina Azeredo
José Guilherme
Marco Antonio Barem

O que pode e o que não pode?

Puff, sofás, barracas etc
Salas 7 e 8



O que pode e o que não pode?

Pedir comida

Usar lab/pc do idp

Sair do IDP



Ajuda externa

Softwares piratas



idp

Desafio

Alimentação



Gestão de ambientes e equipamentos



Desafio

Alimentação

- Filas
- Demora
- Pagamento
- Feedbacks

Gestão de ambientes e equipamentos

- Confuso
- Falar com quem?
- Disponibilidade
-

Entrega da Equipe

Repositório GIT



Demo
(5 min)



Critérios de Avaliação da Solução

- Criatividade, inovação e impacto da solução apresentada;

- Clareza, objetividade e qualidade da apresentação da solução;

- Qualidade da arquitetura e da implementação do código, incluindo padrões de escrita, clareza nas linhas de código e organização;

- Usabilidade, funcionalidade, eficiência e desempenho da solução desenvolvida;

- Uso correto de tecnologias e ferramentas adequadas;

- Trabalho em equipe e colaboração entre os membros da equipe;

- Capacidade de resposta e solução de problemas apresentados pelos avaliadores durante a apresentação;

- Adaptação às regras estabelecidas pela organização do evento;

- Testes e validação adequados do código desenvolvido;

- Documentação clara e completa do código.

Premiação



Cupons na KaBuM!

1º lugar – R\$ 600,00

2º lugar – R\$ 250,00

3º lugar – R\$ 150,00

API e GIT

API para o desafio de Restaurante

API RESTful com dois cenários:

- Lojista (dono do restaurante)
- Cliente (alunos)

API do Lojista:

- Gestão do cardápio por categoria de produtos
- Gestão de pedidos:
 - Fila de pedidos novos
 - Pedidos em preparo
 - Pedidos prontos
 - Pedidos finalizados e cancelados

API do Cliente:

- Gestão de dados pessoais e formas de pagamento
- Acompanhamento de pedido
- Histórico de pedidos

API para o desafio de Reserva de Espaços e Equipamentos

API RESTful com dois cenários:

- Administrador (equipe do IDP)
- Alunos

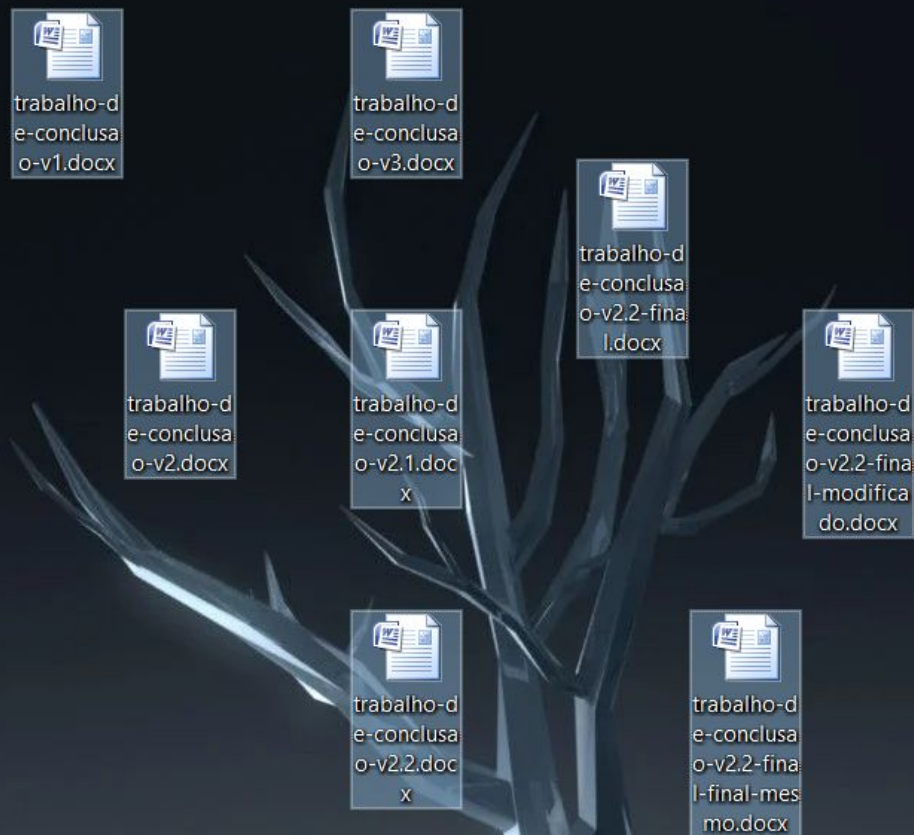
API do Administrador:

- Gestão de Espaços e Equipamentos
- Gestão de reservas

API do Aluno:

- Gestão de dados pessoais
- Gestão de reservas de espaços
- Gestão de reservas de equipamentos
- Histórico de reservas

VERSIONAMENTO [motivação]



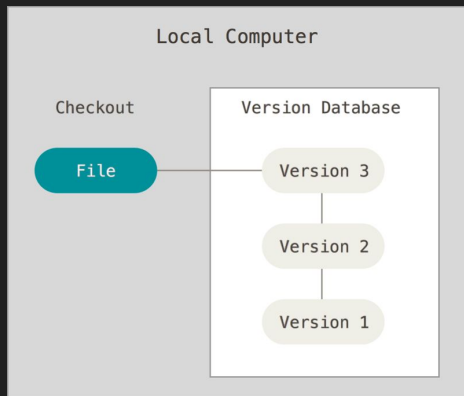
VERSIONAMENTO [motivação]

- O que é "controle de versão", e porque eu deveria me importar?
 - Registro de alterações em arquivos
 - Recuperação de versões funcionais
 - Salvaguarda de conteúdos
 - Facilita colaboração

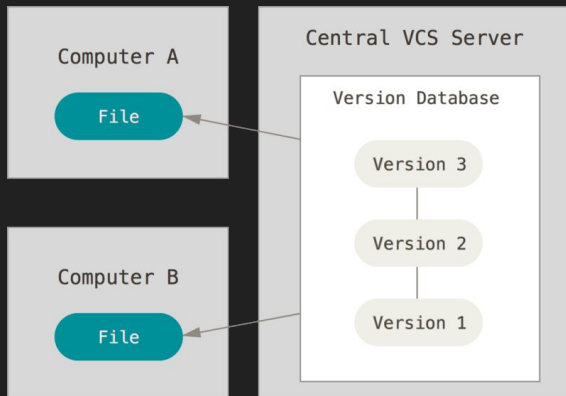
VERSIONAMENTO [histórico]

- O que é "controle de versão", e porque eu deveria me importar?

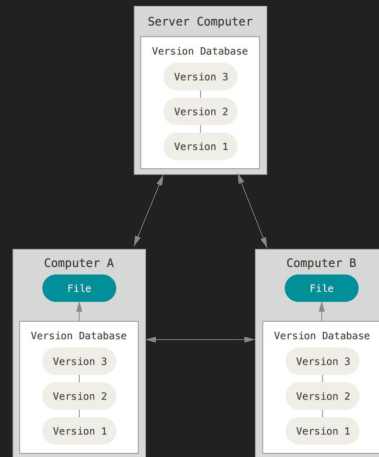
Controle Local



Sistema Centralizado de Controle de Versão



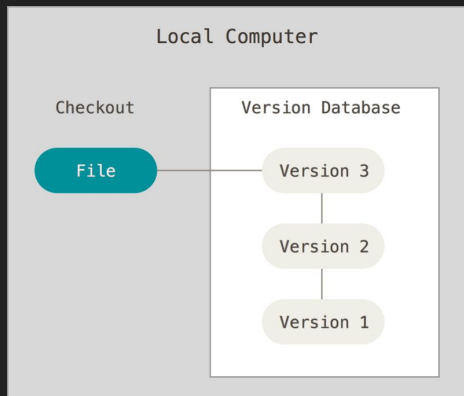
Sistema Distribuído de Controle de Versão



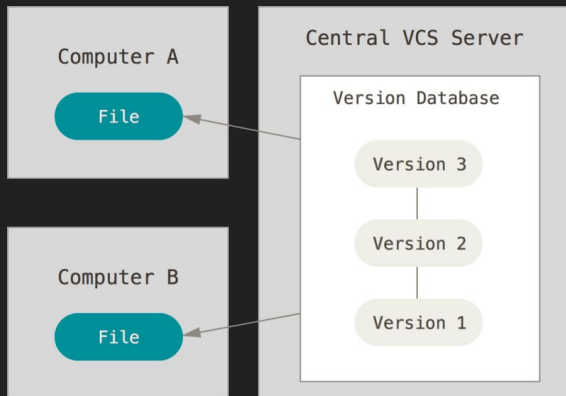
VERSIONAMENTO [histórico]

- O que é "controle de versão", e porque eu deveria me importar?

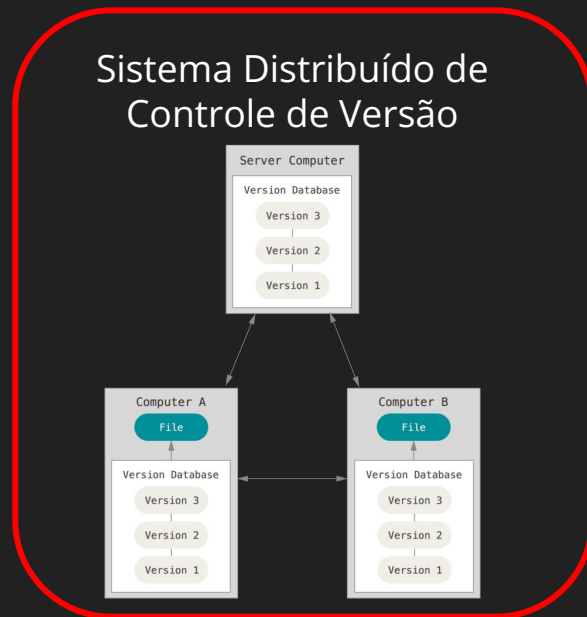
Controle Local



Sistema Centralizado de Controle de Versão



Sistema Distribuído de Controle de Versão



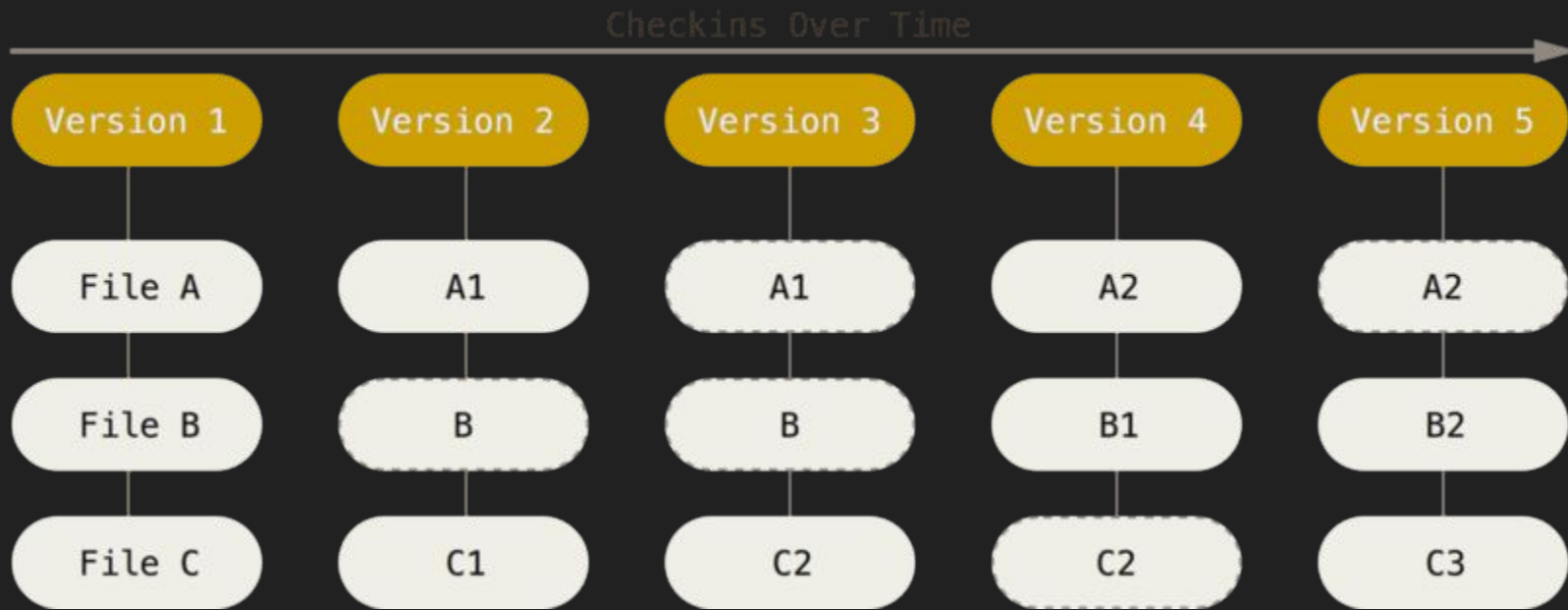
Git, Mercurial,
Bazaar, Darcs

GIT [histórico]

- Surgiu por uma necessidade do [Kernel do Linux](#) (2005)
 - Por problemas com o BitKeeper
 - Necessidades de velocidade e simplicidade
 - Facilidade de desenvolvimento não-linear
 - Capacidade de escala

GIT [básico]

- Trabalha com a ideia de *snapshots* (estados do sistema)



GIT [básico]

- Por quê utilizar GIT?
 - Sincroniza trabalhos locais e remotos
 - Permite experimentação
 - Modificar sem medo de errar
 - Registro de modificações ao longo do tempo
 - Pode ser utilizado para além de código (documentação, relatórios, etc)

GIT [básico]

- Como funciona
 - Cada repositório consiste de um conjunto de arquivos e diretórios
 - Para cada registro de modificação (`commit`), é guardado uma imagem (*snapshot*) de todos os arquivos
 - Arquivos inalterados em uma nova imagem não são duplicados
 - Apenas uma referência à imagem anterior

GIT [básico]

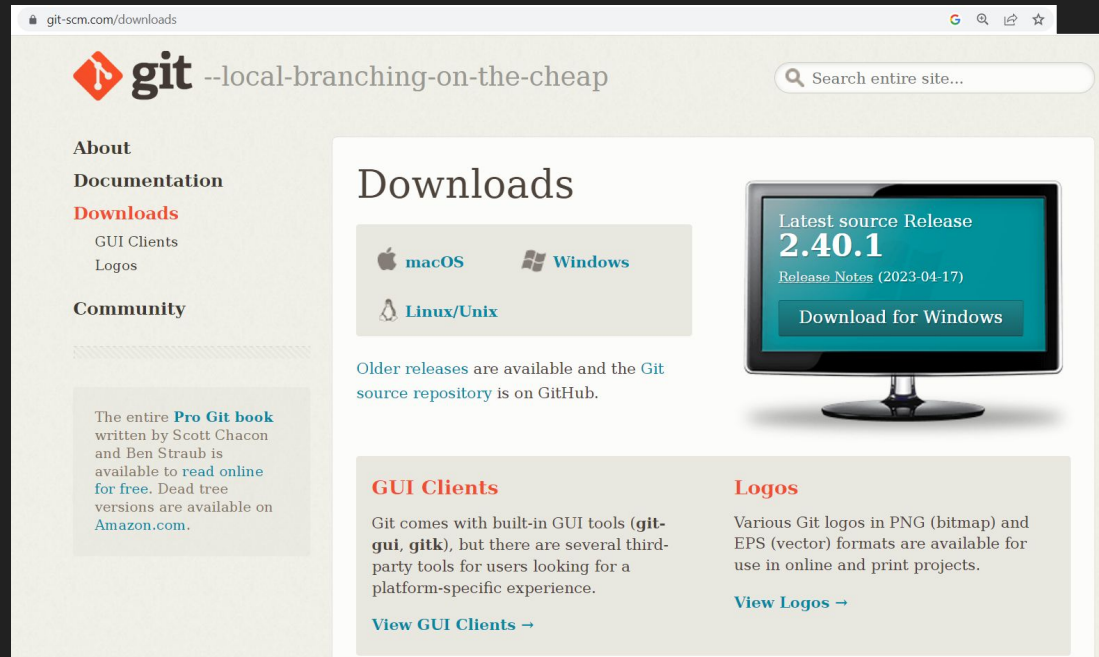
- Integridade
 - Associa um hash a cada imagem
 - Exemplo: 11ca8244562b5d5849cd1d1d6d90ecb0c6764b14
 - Garante que o conteúdo não foi modificado ou corrompido
 - Auxilia na identificação de imagens

GIT [básico]

- Repositórios Locais e Remotos
 - O protocolo git funciona em sua maioria de maneira **local**
 - Inicializar repositório
 - Acrescentar, modificar ou remover arquivos
 - Listar histórico
 - Há a possibilidade de sincronizar um repositório de maneira remota (em um servidor)

GIT [[instalação](#)]

- Windows



GIT [[instalação](#)]

- Linux (sistemas baseados em debian)
 - `sudo apt install git`

```
[9513] j3r3mias@serenity:tmp > sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.8).
git set to manually installed.
The following packages were automatically installed and are no longer required:
  docker-scan-plugin libllvm11 libsasl2-modules
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
1 not fully installed or removed.
Need to get 0 B/157 MB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n]
```

GIT [git help]

- git help <comando>

```
[9518] j3r3mias@serenity:hackathon > git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path=<path>] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        [--super-prefix=<path>] [--config-env=<name>=<envvar>]
        <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status
```

GIT [git config]

- git config - Define informações de autoria

```
[16167] j3r3mias@bebop:hackathon > git config --global user.name "Jeremias Gomes"
[16168] j3r3mias@bebop:hackathon > git config --global user.email "jeremias.gomes@idp.edu.br"
[16169] j3r3mias@bebop:hackathon > █
```

GIT [git init]

- git init
 - Cria um diretório .git de configurações para armazenar metadados do repositório

```
[16174] j3r3mias@bebop:hackathon > git init
Initialized empty Git repository in /tmp/hackathon/.git/
[16175] j3r3mias@bebop:hackathon|master > █
```


GIT [git status]

- git status - verifica status dos arquivos no repositório

```
[9526] j3r3mias@serenity:hackathon|master > ls
lista.txt
[9527] j3r3mias@serenity:hackathon|master > cat lista.txt
computador
teclado
mouse
[9528] j3r3mias@serenity:hackathon|master > █
```

```
[9528] j3r3mias@serenity:hackathon|master > git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    lista.txt

nothing added to commit but untracked files present (use "git add" to track)
[9529] j3r3mias@serenity:hackathon|master > █
```

GIT [git add]

- `git add <arquivo>` - adiciona arquivo ao projeto (staging area)
 - O arquivo passa a ser “observado”

```
[9530] j3r3mias@serenity:hackathon|master > git add lista.txt
[9531] j3r3mias@serenity:hackathon|master > git status
On branch master

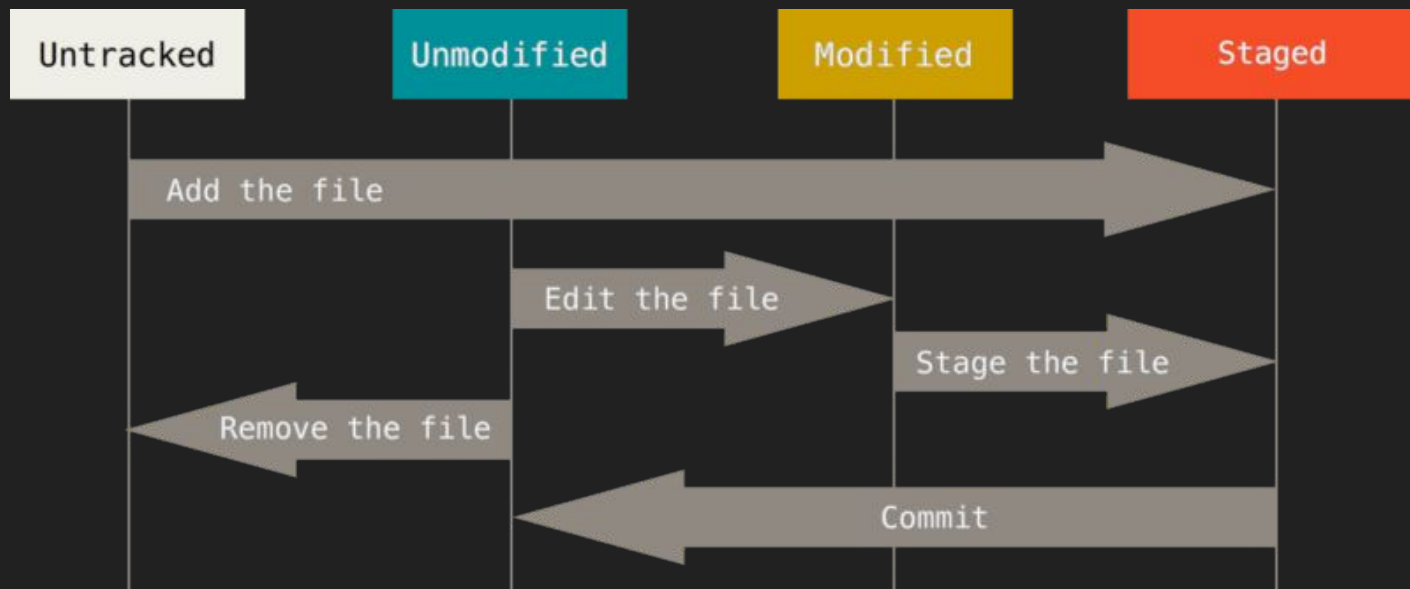
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   lista.txt

[9532] j3r3mias@serenity:hackathon|master > █
```

GIT [git]

- Estados de um arquivo



GIT [git commit]

- git commit - adiciona arquivos observados ao repositório
 - Entra para o histórico

```
[9532] j3r3mias@serenity:hackathon|master > git commit -am 'Adicionando a lista ao repositório.'
Warning: commit message did not conform to UTF-8.
You may want to amend it after fixing the message, or set the config
variable i18n.commitencoding to the encoding your project uses.
[master (root-commit) a9e2182] Adicionando a lista ao repositório.
 1 file changed, 3 insertions(+)
 create mode 100644 lista.txt
[9533] j3r3mias@serenity:hackathon|master > git status
On branch master
nothing to commit, working tree clean
[9534] j3r3mias@serenity:hackathon|master > ls
lista.txt
[9535] j3r3mias@serenity:hackathon|master > █
```

GIT [git log]

- git log - exibe o histórico de commits

```
[16179] j3r3mias@bebop:hackathon|master > git log
commit 64588c74235b8146160d858352be4d1bc4c11bdb (HEAD -> master)
Author: Jeremias Gomes <jeremias.gomes@idp.edu.br>
Date: Thu May 25 18:23:31 2023 -0300
```

Adicionando lista ao repositório.

```
[16180] j3r3mias@bebop:hackathon|master > █
```

```
[16180] j3r3mias@bebop:hackathon|master > git log --oneline
64588c7 (HEAD -> master) Adicionando lista ao repositório.
[16181] j3r3mias@bebop:hackathon|master > █
```

GIT [git diff]

- git diff [commit_01] [commit_02] - lista as diferenças
 - Por padrão, as modificações atuais com o commit anterior

```
[16183] j3r3mias@bebop:hackathon|master > echo 'monitor' >> lista.txt
[16184] j3r3mias@bebop:hackathon|master > cat lista.txt
computador
teclado
mouse

monitor
[16185] j3r3mias@bebop:hackathon|master > git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   lista.txt

no changes added to commit (use "git add" and/or "git commit -a")
[16186] j3r3mias@bebop:hackathon|master > git diff lista.txt
diff --git a/lista.txt b/lista.txt
index 3d56593..f4212d6 100644
--- a/lista.txt
+++ b/lista.txt
@@ -2,3 +2,4 @@ computador
teclado
mouse

+monitor
[16187] j3r3mias@bebop:hackathon|master > █
```

GIT [git checkout]

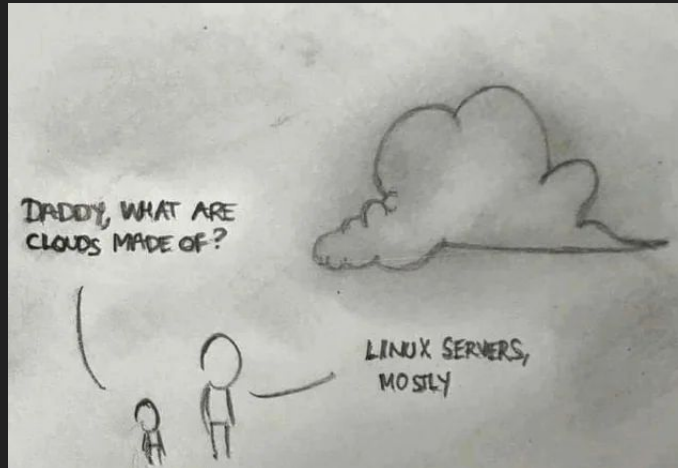
- git checkout [commit] - navega por um histórico
 - Ou desfaz alterações

```
[16188] j3r3mias@bebop:hackathon|master > git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   lista.txt

no changes added to commit (use "git add" and/or "git commit -a")
[16189] j3r3mias@bebop:hackathon|master > git checkout lista.txt
Updated 1 path from the index
[16190] j3r3mias@bebop:hackathon|master > git status
On branch master
nothing to commit, working tree clean
[16191] j3r3mias@bebop:hackathon|master > █
```

GIT [desenvolvimento colaborativo]

- Uma das vantagens do uso de git é o versionamento coletivo
 - Desenvolvimento colaborativo por repositórios remotos
- Um repositório remoto é apenas um repositório em outra máquina




GIT [github.com e gitlab.com]

- Existem serviços que disponibilizam servidores para realizar armazenamento colaborativo
 - Github e Gitlab são os mais famosos, para o protocolo git
 - Possuem contas gratuitas
 - Incentivam o desenvolvimento open-source
 - Repositórios podem ser públicos ou privados


GIT [github.com]

- Criação de repositório

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

 Set up in Desktop or


HTTPSSSHgit@github.com:j3r3mias/hackathon-teste.git



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a README, LICENSE, and .gitignore.


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

```
echo "# hackathon-teste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:j3r3mias/hackathon-teste.git
git push -u origin main
```



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

```
git remote add origin git@github.com:j3r3mias/hackathon-teste.git
git branch -M main
git push -u origin main
```



GIT [git clone]

- git clone <repositório> - Baixa um repositório
 - Precisa ter acesso a esse repositório

```
[16228] j3r3mias@bebop:tmp > git clone git@github.com:j3r3mias/hackathon-teste.git
Cloning into 'hackathon-teste'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
[16229] j3r3mias@bebop:tmp > cd hackathon-teste/
[16230] j3r3mias@bebop:hackathon-teste|main > ls
lista.txt
[16231] j3r3mias@bebop:hackathon-teste|main > git log --oneline
93afa43 (HEAD -> main, origin/main, origin/HEAD) Inclusão de gabinete.
d937f7b Typo na primeira palavras.
e0d0d80 Lista com três elementos.
[16232] j3r3mias@bebop:hackathon-teste|main > █
```

GIT [git pull]

- git pull - Recebe commits do repositório remoto

```
[16246] j3r3mias@bebop:hackathon-teste|main > ls
lista.txt
[16247] j3r3mias@bebop:hackathon-teste|main > git log --oneline
93afa43 (HEAD -> main, origin/main, origin/HEAD) Inclusão de gabinete.
d937f7b Typo na primeira palavras.
e0d0d80 Lista com três elementos.
[16248] j3r3mias@bebop:hackathon-teste|main > git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 274 bytes | 274.00 KiB/s, done.
From github.com:j3r3mias/hackathon-teste
   93afa43..591ae53  main       -> origin/main
Updating 93afa43..591ae53
Fast-forward
  outro-arquivo.txt | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 outro-arquivo.txt
[16249] j3r3mias@bebop:hackathon-teste|main > ls
lista.txt  outro-arquivo.txt
[16250] j3r3mias@bebop:hackathon-teste|main > █
```

GIT [git pull]

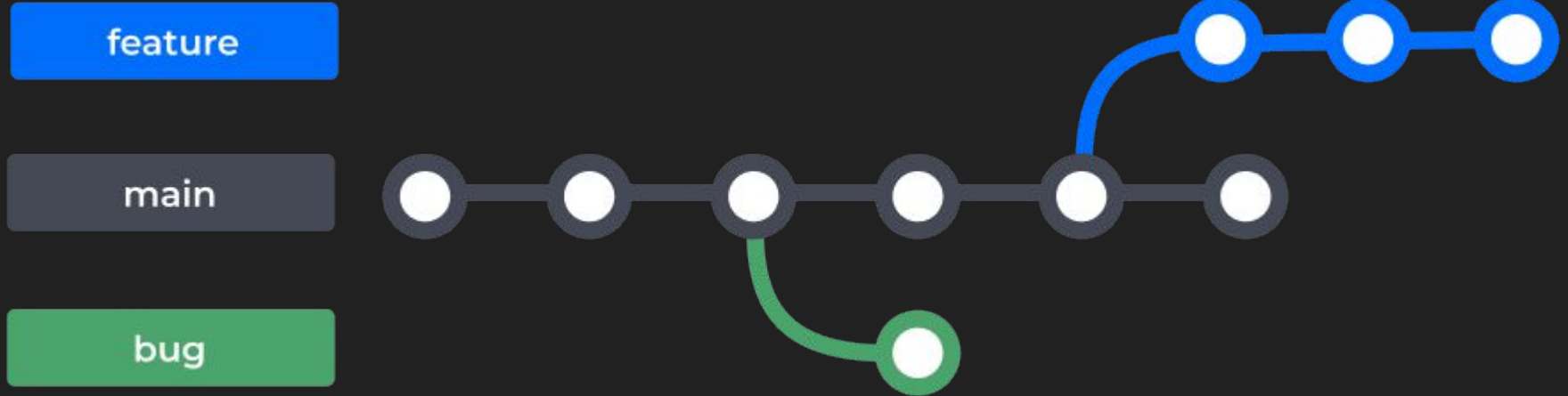
- `git pull [origem] [branch]` - Recebe commits do repositório remoto

```
[16254] j3r3mias@bebop:hackathon-teste|main > git pull
Already up to date.
[16255] j3r3mias@bebop:hackathon-teste|main > git pull origin main
From github.com:j3r3mias/hackathon-teste
 * branch          main          -> FETCH_HEAD
Already up to date.
[16256] j3r3mias@bebop:hackathon-teste|main > █
```

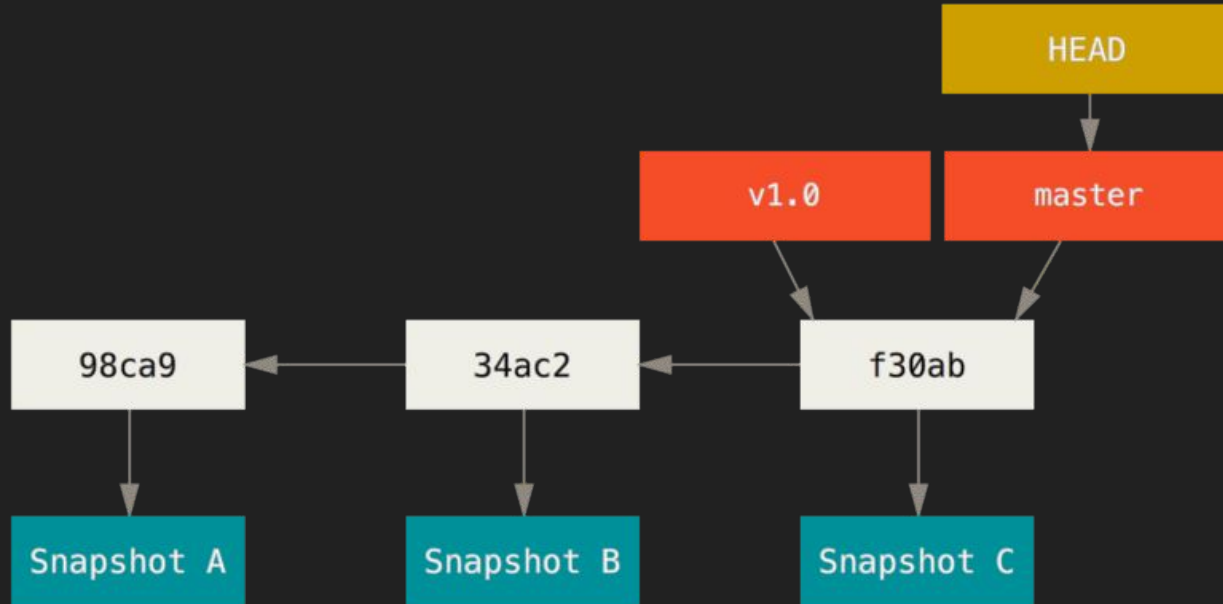
GIT [branches]

- Uma das intenções do desenvolvimento colaborativo é a possibilidade de pessoas diferentes alterarem pontos diferentes (ou não) em um mesmo projeto
- **Branches** possibilitam esse trabalho, inclusive em diferentes versões de um mesmo repositório

GIT [branches]



GIT [branches]



GIT [git branch]

- git branch <nome> - Cria uma nova branch

```
[16257] j3r3mias@bebop:hackathon-teste|main > git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
[16258] j3r3mias@bebop:hackathon-teste|main > git branch alteracao
[16259] j3r3mias@bebop:hackathon-teste|main > git branch -a
  alteracao
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
[16260] j3r3mias@bebop:hackathon-teste|main > git checkout alteracao
Switched to branch 'alteracao'
[16261] j3r3mias@bebop:hackathon-teste|alteracao > vim lista.txt
[16262] j3r3mias@bebop:hackathon-teste|alteracao > █
```

GIT [git merge]

- `git merge <nome_branch>` - Junta dois branches (atual e nomeado)
 - O merge tenta juntar os dois branches de maneira automática
 - Porém, como é possível que um mesmo arquivo contenha modificações complexas, conflitos podem ocorrer
 - Se for o caso, o usuário é responsável por resolver esses conflitos e aplicar o push

GIT [git merge]

- git merge <nome_branch>

```
[16281] j3r3mias@bebop:hackathon-teste|alteracao > git branch -a
* alteracao
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
[16282] j3r3mias@bebop:hackathon-teste|alteracao > git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[16283] j3r3mias@bebop:hackathon-teste|main > git merge alteracao
Updating 591ae53..916e26b
Fast-forward
 lista.txt | 1 +
 1 file changed, 1 insertion(+)
[16284] j3r3mias@bebop:hackathon-teste|main > git log --oneline
916e26b (HEAD -> main, alteracao) Alteração na lista via branch.
591ae53 (origin/main, origin/HEAD) Outro arquivo.
93afa43 Inclusão de gabinete.
d937f7b Typo na primeira palavras.
e0d0d80 Lista com três elementos.
[16285] j3r3mias@bebop:hackathon-teste|main > █
```

GIT [conflitos]

- Conflitos em merges são marcados no arquivo com
 - >>>>>> - Para a modificação externa (de outra branch)
 - <<<<<<< - Para a versão atual da branch (com o conflito durante o merge)
- Usuário precisa remover esses destaques e resolver o conflito

```
[9583] j3r3mias@serenity:hackathon|master > cat lista.txt
computador
teclado
mouse
<<<<<<< HEAD
monitor
=====
tela
>>>>>> branch_alteracao
fonte

[9584] j3r3mias@serenity:hackathon|master > █
```



THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



Start!

