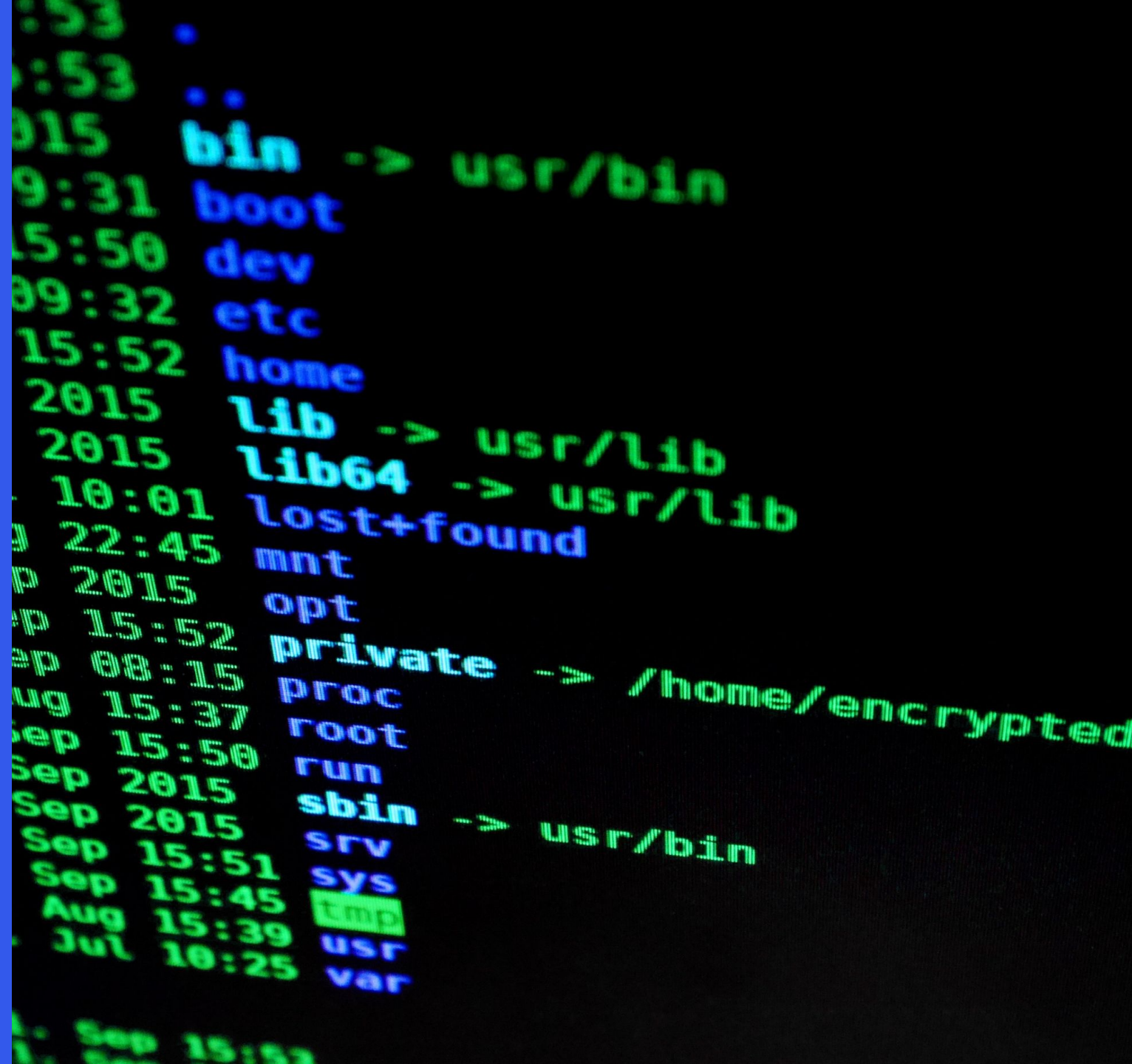




# Git Terminal

Controle de versão

< thefutureisblue.me />



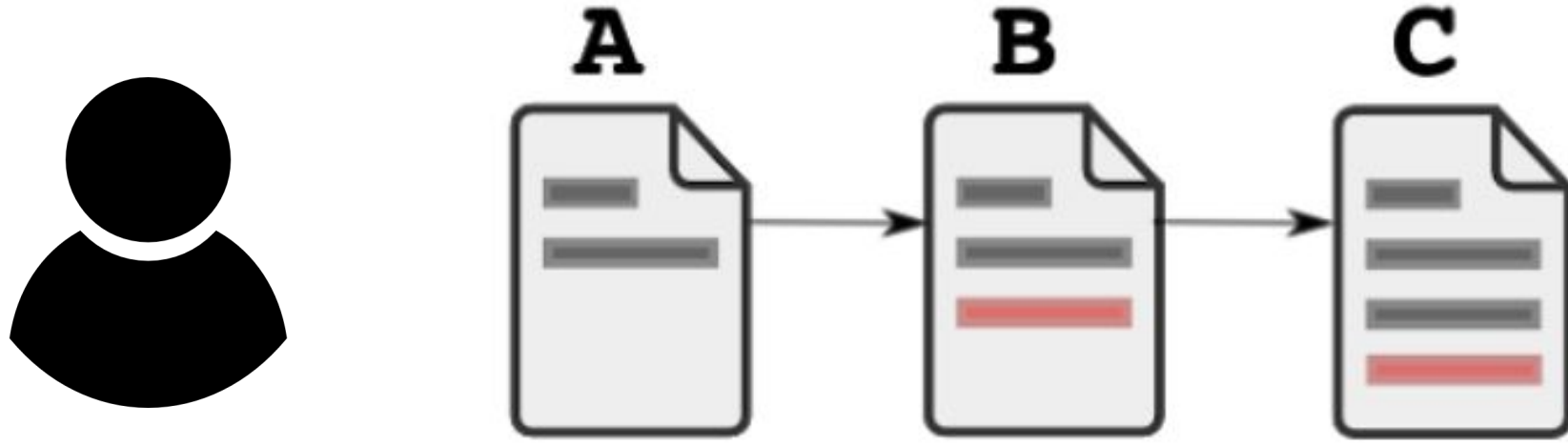
# Antes, vamos lembrar o que é **versionamento**.

Como o próprio nome sugere, **versionamento** é a tarefa de **manter e controlar algo por meio de versões**. Como **versão** podemos entender: um determinado conteúdo (seja código ou não) em determinado momento. Exemplos de versionamento:

**Histórico de um documento.**

**Versão de atualização de um jogo  
(já ouviram falar em patch?).**

**Atualização de segurança de um aplicativo.**



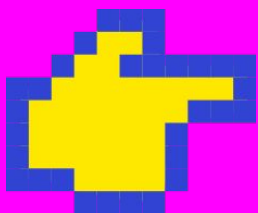
# Por que **versionar** é tão importante?

Olá, essa tela está igualzinha à próxima! Qual delas será mantida?

Alguns dos benefícios de **versionar** um conteúdo:

- Histórico altamente detalhado e descritivo. Podemos saber quem alterou o conteúdo, além de quando e como a alteração foi realizada;
- Ter diversas versões de um mesmo conteúdo e também realizar trabalhos e experimentos em paralelo;
- Nos traz muito mais segurança quanto ao conteúdo que estamos versionando;
- Nos permite até viajar no tempo.

# Por que versionar é tão importante?



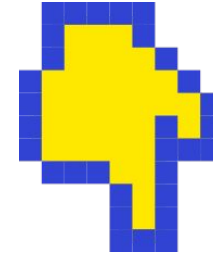
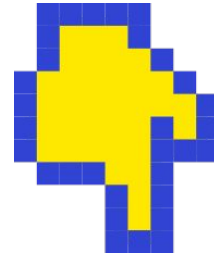
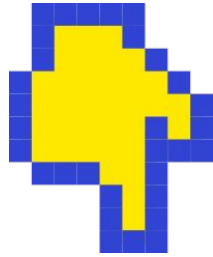
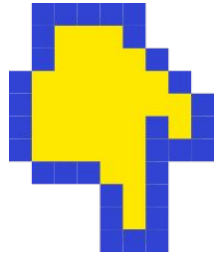
Alguns dos benefícios de versionar um conteúdo:

- Histórico altamente detalhado e descritivo, podemos saber quem alterou o conteúdo, além de quando e como a alteração foi realizada;
- Ter diversas versões de um mesmo conteúdo e também realizar trabalhos e experimentos em paralelo;
- Nos traz muito mais segurança quanto ao conteúdo que estamos versionando;
- Nos permite até viajar no tempo.

# Viajar no tempo?







**Git** é uma ferramenta que nos permite  
trabalhar com **versionamento!**





Você pode  
baixá-la  
através do  
site oficial:



Imagem: Freepik.com

<https://git-scm.com/downloads>

É só baixar o executável e fazer o download. O processo de instalação é padrão e já conhecemos: next, next e finish. As opções padrão são o que você precisa para começar.



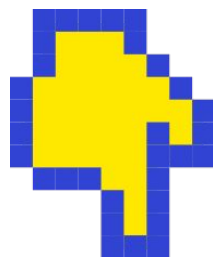
# Como **Git** funciona?

É uma **ferramenta de linha de comando**, que nos permite trabalhar com versionamento de código.

**repositório:** local onde os arquivos de um projeto são mantidos.

**commit:** ponto histórico em que uma alteração foi feita nos arquivos de projeto.

# Alguém disse comandos?

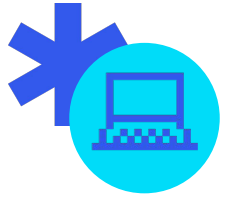


- **git --help**: mostra uma lista dos comandos disponíveis para utilização;
- **git init**: inicia um repositório no seu projeto;
- **git status**: mostra o "status" dos modificados do seu projeto;
- **git add *nome-do-arquivo***.: adiciona os arquivos para a "staging" área;
- **git commit -m 'mensagem'**: "commita" os arquivos (criando um novo histórico de edição);
- **git log**: exibe o histórico dos commits.



- **git revert HEAD**: desfaz a última alteração (commit);
- **git revert *hash-do-commit***: reverte para o estado desejado;
- **git status**: mostra o "status" dos modificados do seu projeto;
- **git branch**: lista as "branch" criadas;
- **git checkout *nome-da-branch***: navega para a branch escolhida; pode criar uma nova com o parâmetro **-b**
- **git merge *nome-da-branch***: mescla a branch desejada com a atual. Pode ser necessário resolver conflitos.





# Exercício

Utilizando o Git, você deverá:

Iniciar um repositório na pasta terminal , criada na aula anterior.

Realizar um commit para subir os arquivos criados.

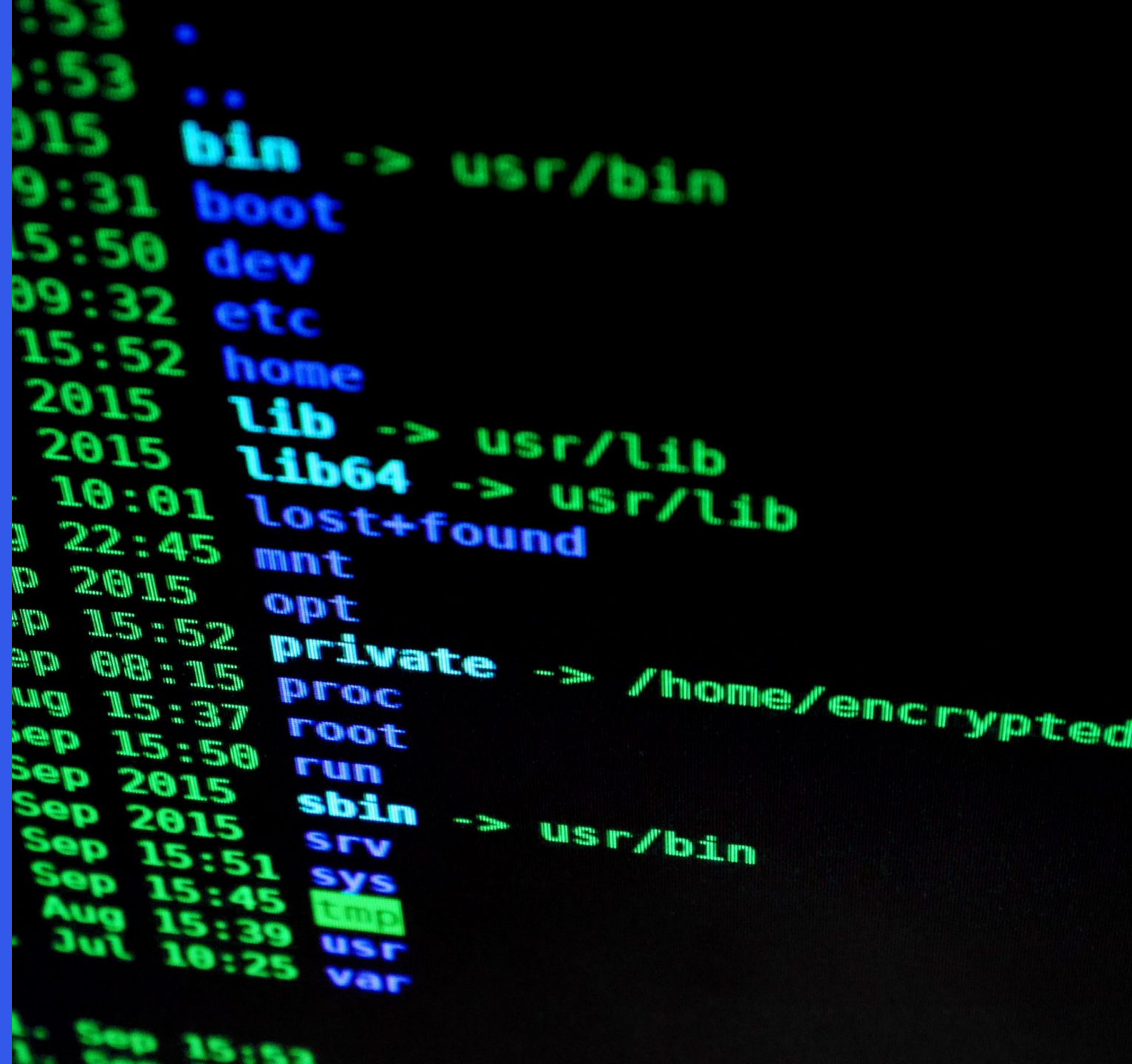




## Um pouco mais de Git

Branches  
e compartilhamento

< thefutureisblue.me />

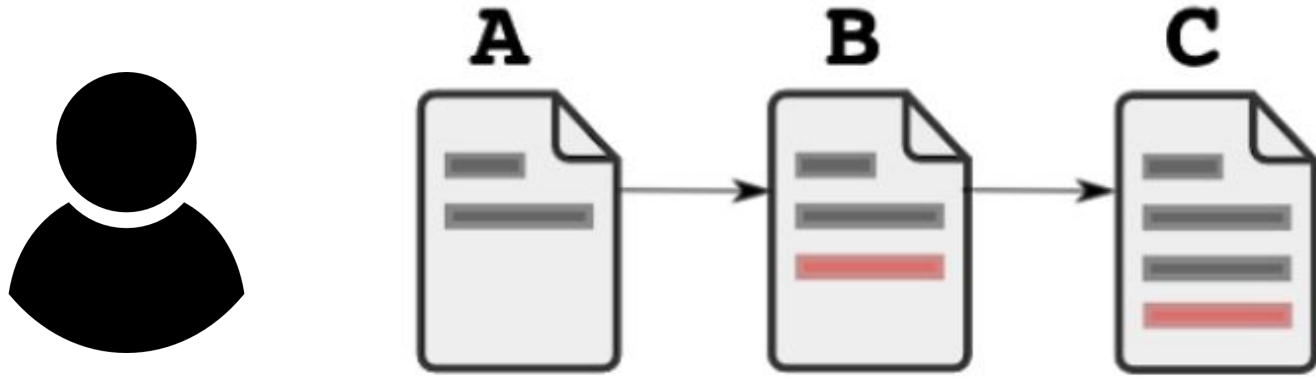


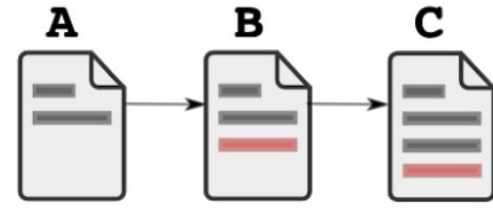
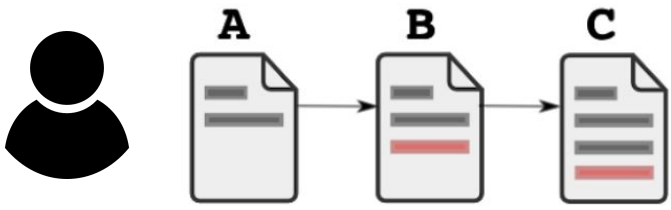
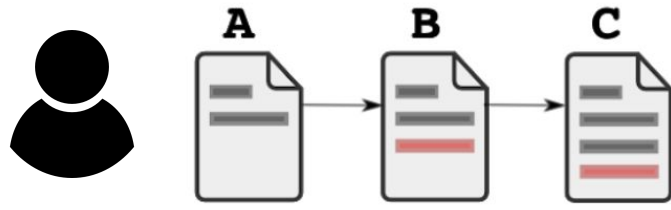
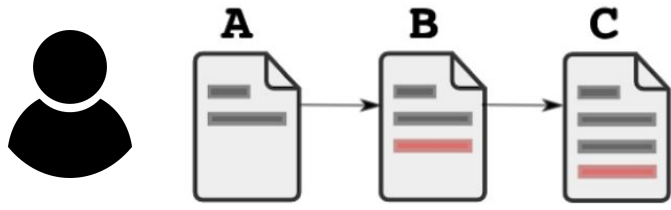


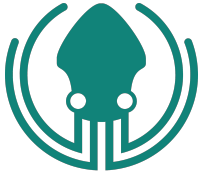
# Como funciona o versionamento em um time?

Alguém arrisca um palpite?









GitKraken



Github Desktop



SourceTree

Outros serviços de git em nuvem (fora github)



Bitbucket



GitLab

Por hoje é só!  
Obrigado! =)

Até a próxima aula!