

Técnico em Desenvolvimento de Sistemas



Programação Front-end

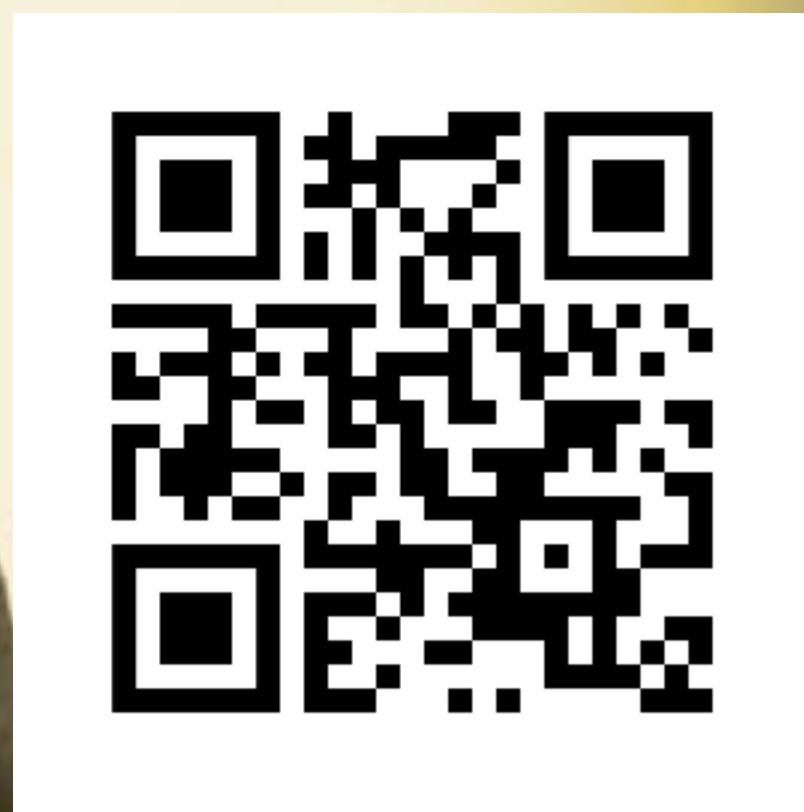
Em instantes iniciaremos nossa aula

Professor João Choma Neto



O que nós vimos na aula passada?

<https://github.com/JoaoChoma/frontend/tree/main/SEMANA-RECUPERACAO-PROVA-05>



REACT

<https://pt-br.react.dev/>



<https://codesandbox.io/>



E ESSA SEMANA?



LIÇÃO 18: VERSIONAMENTO

<https://sites.google.com/view/tec-ads-programacao-front-end/3%C2%BA-trimestre/li%C3%A7%C3%A3o-18>

OBJETIVO DA AULA

O **objetivo** da aula de hoje é conhecer a ferramenta **GIT**.

Ferramenta para **versionamento** de arquivos

VERSIONAMENTO

O versionamento de arquivos é o processo de rastrear e gerenciar diferentes versões de um arquivo ao longo do tempo

VERSIONAMENTO

Acompanhar as alterações feitas em um arquivo, permitindo que os usuários mantenham um histórico das mudanças e revertam para versões anteriores, se necessário.

VERSIONAMENTO

Colaboração: Permite que várias pessoas trabalhem em um mesmo arquivo sem sobrescrever as alterações umas das outras. Cada alteração é registrada como uma nova versão.

VERSIONAMENTO

Histórico: Mantém um histórico das mudanças feitas em um arquivo, o que pode ser útil para rastrear quando e por quem uma alteração específica foi feita.

VERSIONAMENTO

Recuperação: Permite a recuperação de versões anteriores de um arquivo em caso de erros, perda de dados ou necessidade de voltar a uma versão anterior.

VERSIONAMENTO

Você provavelmente já usou algum sistema de **versionamento de arquivos** mesmo antes de estudar **tecnologia**

trabalho_V1.docx
trabalho_V2.docx
trabalho_final.docx

VERSIONAMENTO

Temos a falta de controle sobre alterações em arquivos em um projeto, especialmente em projetos **colaborativos** que envolvem várias pessoas trabalhando no **mesmo conjunto de arquivos**.

VERSIONAMENTO

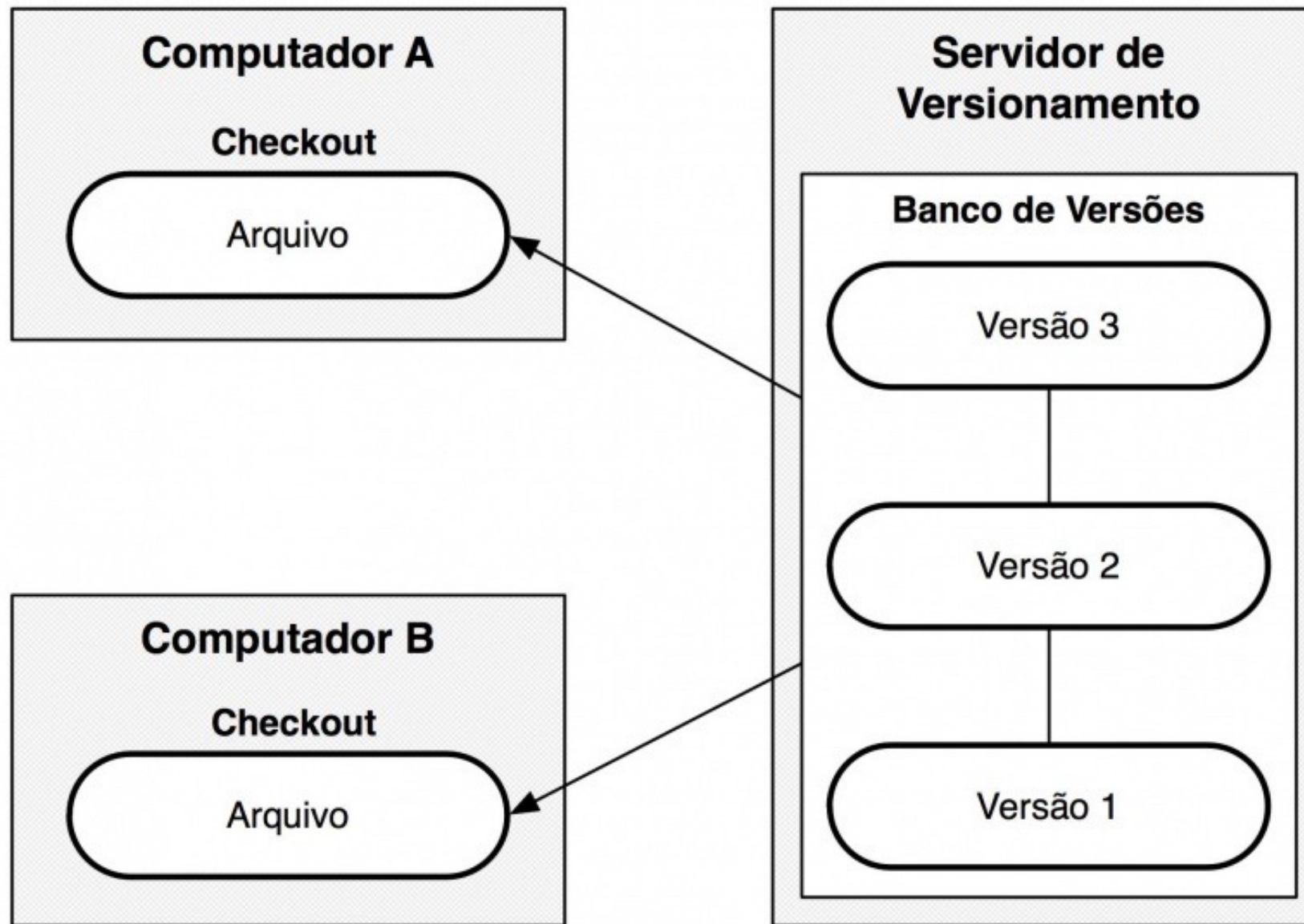
O versionamento de arquivos, todas as **alterações** feitas em um arquivo são **registradas** e mantidas em um **repositório** centralizado, permitindo acompanhar e entender facilmente as mudanças realizadas.

CONTROLE DE VERSÃO

Controle de versão é a **prática de gerenciar alterações** em arquivos de código-fonte ao longo do tempo, permitindo que você volte a versões anteriores, se necessário.

CONTROLE DE VERSÃO

Isso é feito através do uso de um sistema de controle de versão, como o **GIT**, que permite rastrear as mudanças feitas em cada arquivo e a data em que foram feitas e, principalmente, quem as realizou.



GIT

GIT é um sistema de **controle de versão**, ou versionamento, que permite que você **rastreie alterações** em seus arquivos ao longo do tempo. Com isso, você pode criar **diferentes versões** de um arquivo e alterar entre elas facilmente.



GIT

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

GIT



Várias pessoas podem trabalhar
no **mesmo arquivo**, ao mesmo tempo,
sem se preocupar em perder o trabalho
um do outro.

GIT



O **GIT** ainda mantém registro de todas as alterações feitas em um arquivo, o que lhe permite voltar às versões **anteriores**.

GIT



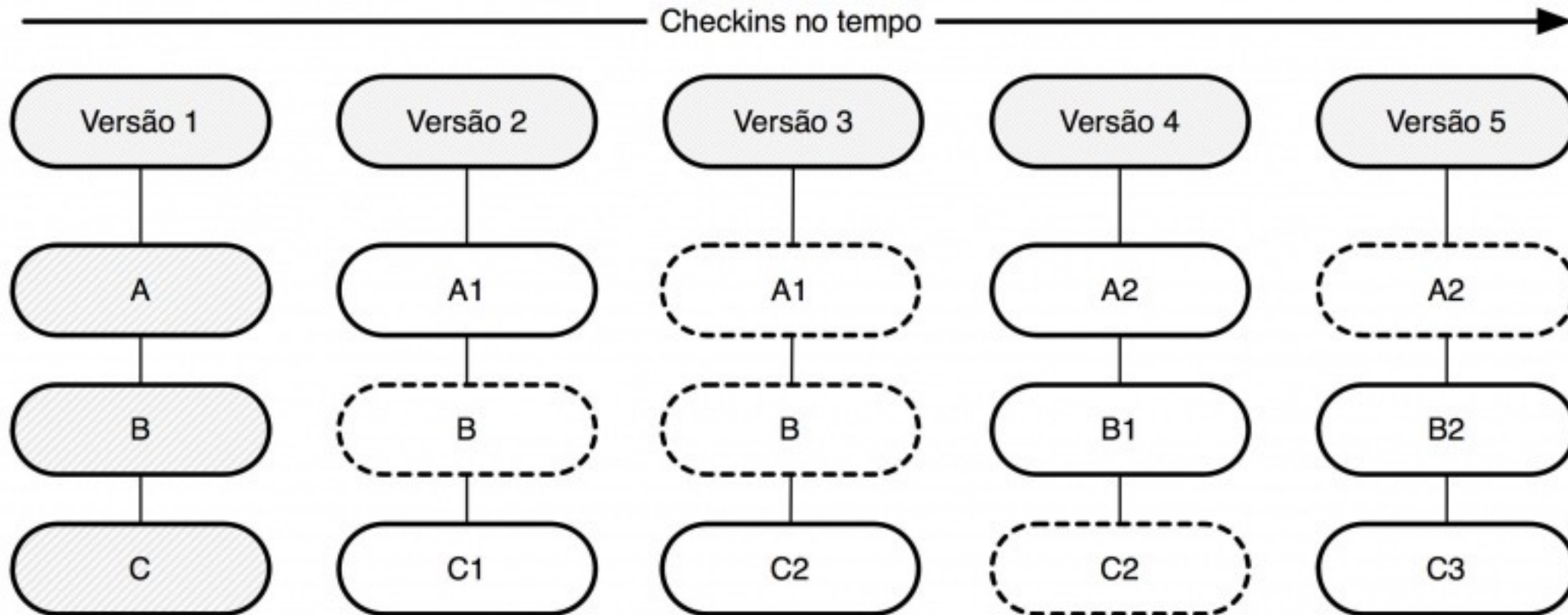
O GIT trata seus dados como um conjunto de imagens de um sistema de arquivos em miniatura

GIT



Toda vez que você salvar uma nova versão o que é salvo é o estado de seu projeto GIT, ele basicamente **tira uma foto de todos os arquivos** e armazena uma referência para esse conjunto de arquivos.

GIT

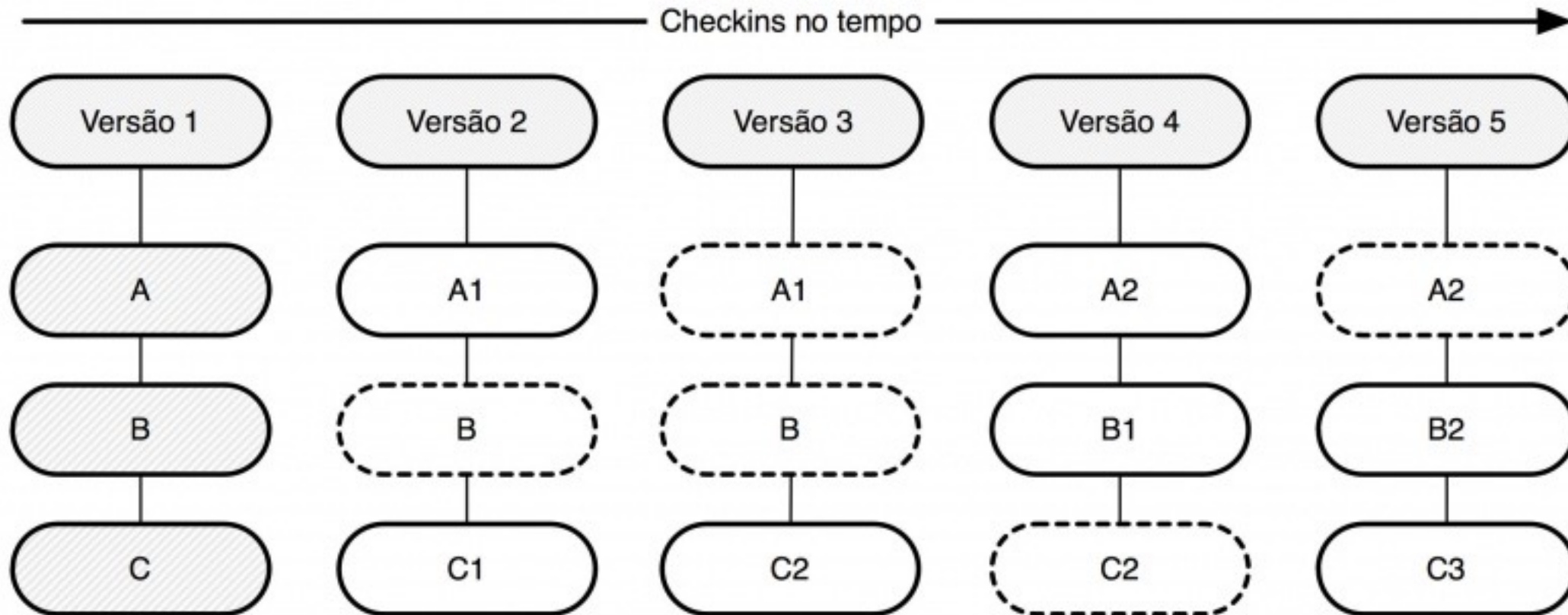


GIT



Caso o arquivo não tenha sido alterado, ele não armazena o arquivo novamente, apenas cria um link para o arquivo idêntico anterior já armazenado.

GIT



ESTADOS DE UM ARQUIVO NO GIT

Qualquer arquivo possui **3 estados no GIT**

- ***Committed (registrado)***: significa que os dados desses arquivos já estão armazenados, versionados de forma segura pelo GIT localmente.

ESTADOS DE UM ARQUIVO NO GIT

Qualquer arquivo possui **3 estados no GIT**

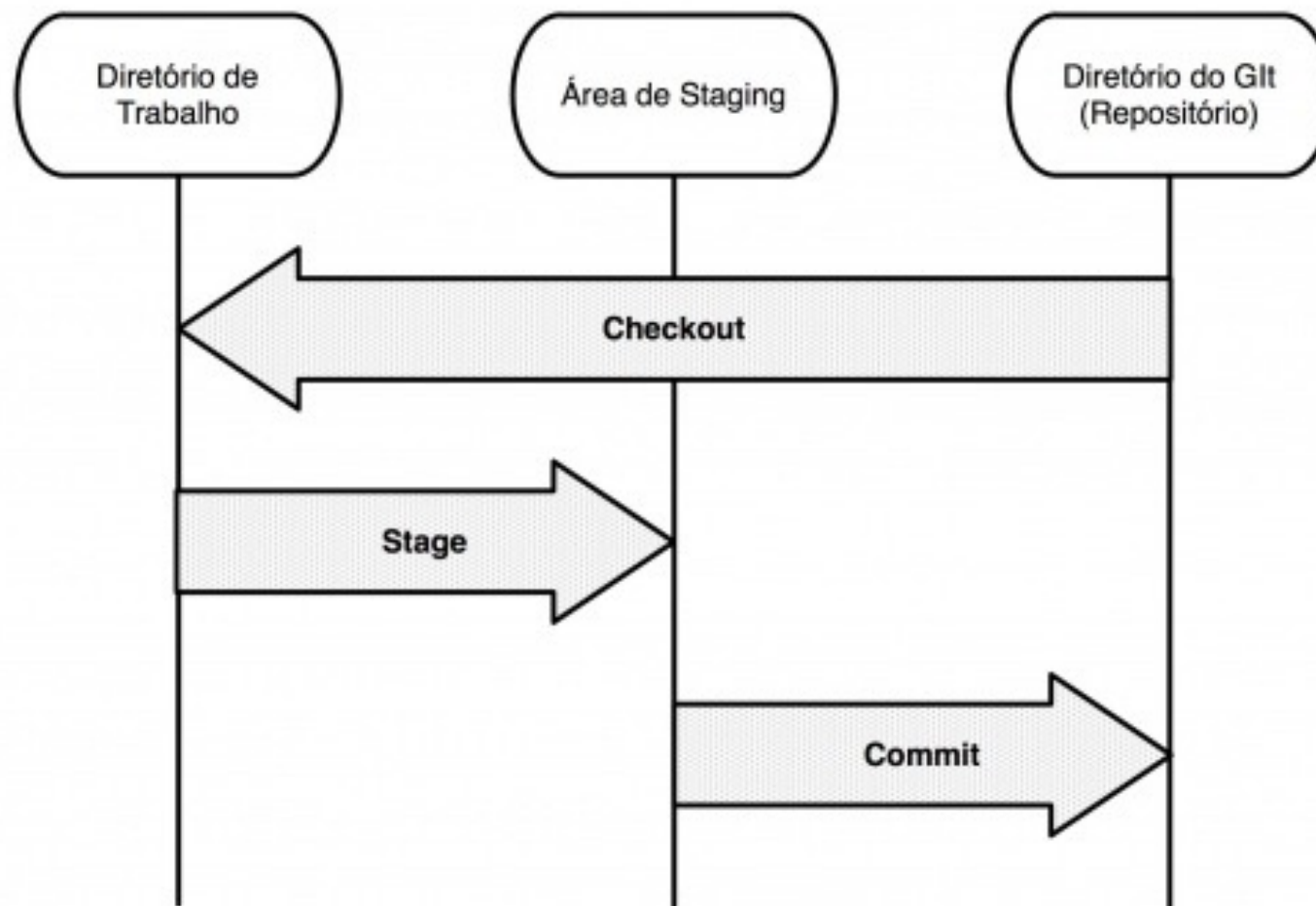
• ***Modified (modificado)***: significa que você alterou o arquivo, mas ainda não fez o *commit* no seu banco de dados.

ESTADOS DE UM ARQUIVO NO GIT

Qualquer arquivo possui **3 estados no GIT**

• ***Staged (organizar)***: significa que você marcou a versão atual de um arquivo modificado para fazer parte de seu próximo *commit*.

ESTADOS DE UM ARQUIVO NO GIT



COMANDOS GIT

git init: inicializa um novo repositório git;

git add: adiciona alterações aos arquivos ao stage;

git commit: cria um novo commit com as alterações adicionadas;

COMANDOS GIT

git status: exibe o status atual do repositório, incluindo arquivos modificados e não adicionados ao stage;

git log: exibe um histórico detalhado de commits;

COMANDOS GIT

git clone: clona um repositório remoto existente na internet ou em sua própria máquina;

git push: envia as alterações locais para um repositório remoto;

git pull: obtém as alterações mais recentes de um repositório remoto;

COMANDOS GIT

git branch: exibe uma lista de branches (ramos) no repositório atual;

git checkout: muda para um branch (ramo) no repositório atual;

COMANDOS GIT

git merge: combina as alterações de um ramo para outro;

git remote: gerencia as conexões com repositórios remotos;

git diff: exibe as diferenças entre duas versões de um arquivo.



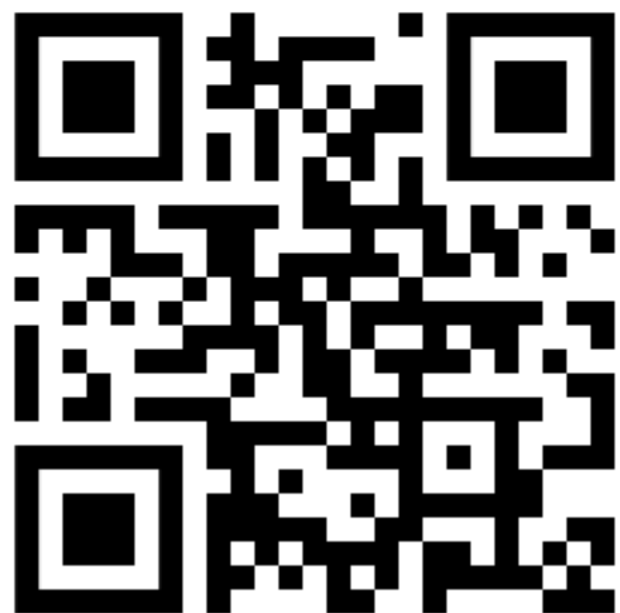
GITHUB

<https://github.com/>

COMO CRIAR CONTA NO GITHUB



https://www.youtube.com/watch?v=Prxgo_pXUI



GIT

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

SIMULADOR JOGO

https://learngitbranching.js.org/?locale=pt_BR





SIMULADOR

<https://git-school.github.io/visualizing-git/#free>



Alura

Desenvolvimento de Sistemas – Front end - João Choma Neto

<https://cursos.alura.com.br/course/html-css-praticando-html-css>



PARANÁ 
GOVERNO DO ESTADO
SECRETARIA DA EDUCAÇÃO
E DO ESPORTE