

# Versionamento de código

# **GIT I**

Professor - Renato Naumann



# O que vamos aprender **hoje?**

- O que é o Git
- Por que devemos ter um sistema de controle de versão
- Estados de um arquivo
- Como instalar o Git no Windows
- Como configurar o Git
- Comandos básicos do Git
- Como criar nosso primeiro repositório local
- O que é o GitLab
- Como trabalhar com repositório remoto

# O que é GIT?

O Git é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo (Exemplo: alguns livros digitais são disponibilizados no GitHub e escrito aos poucos publicamente).

O Git foi inicialmente projetado e desenvolvido para o desenvolvimento do kernel Linux, mas foi adotado por muitos outros projetos.

Cada diretório de trabalho do Git é um repositório com um histórico completo e habilidade total de acompanhamento das revisões, não dependente de acesso a uma rede ou a um servidor central.



# Por que utilizar?

Um sistema de controle de versão permite a colaboração de vários desenvolvedores em uma base de código compartilhada, onde as ações podem ocorrer de forma simultânea. Devido a esta natureza dinâmica, é vital que mais de uma pessoa possa trabalhar na mesma base sem afetar o trabalho de outro indivíduo.



# Quais os estados de um arquivo?



## Modificado

Uma vez no repositório, qualquer arquivo que é **adicionado, modificado** ou **removido** é marcado como modificado. Significa que o arquivo sofreu alterações, mas ainda não foi dito que ele fará parte da próxima versão que será consolidada. Significa também que esses arquivos são diferentes quando comparados com a última versão disponível no histórico.



## Preparado

A partir do momento que o comando **git add** é executado em algum arquivo, ele assume o estado de preparado. Neste momento, o Git sabe que o arquivo foi modificado e agora está na área de preparação para ser consolidado.



## Consolidado

Após toda a preparação, os arquivos são finalmente salvos quando o comando **git commit** é executado. Se algum arquivo se encontra no estado modificado e é feito um **git commit**, estes arquivos continuarão como modificados, eles não farão parte do commit, pois nunca foram preparados.

# Como instalar o Git no Windows?



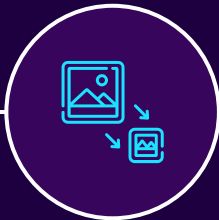
Basta acessar o site oficial para  
baixar o instalador do git:  
<https://git-scm.com/downloads>

# Como configurar o **Git**?



## Nome

\$ git config --global user.name  
**"YOUR NAME"**



## Email

\$ git config --global user.email  
**"YOUR EMAIL ADDRESS"**



1

# Vamos configurar?

Abra o terminal e vamos fazer juntos





# Comandos básicos



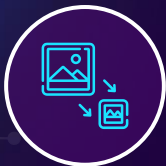
## git init

Inicializar um repositório git



## git status

Exibe o status dos arquivos no repositório



## git add

Adiciona um arquivo em uma lista de arquivos a ser monitorado

## git commit

Cria um ponto de referência com o estado atual de todos os arquivos



## git rm

Remove um arquivo da lista de arquivos monitorados



## git checkout

Restaura arquivos da árvore de trabalho



2

**Vamos ver na prática?**

Abram o terminal e vamos trabalhar com  
repositório local



# Comandos básicos



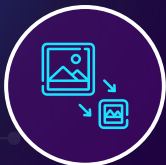
## git clone

Cria uma cópia exata de um repositório remoto



## git pull

Busca modificações do repositório remoto



## git push

Envia as modificações para o repositório remoto

## git log

Usado para visualizar o histórico



## gitk

Interface gráfica para um repositório local



## git remote -v

Exibe o endereço do repositório remoto



3

**Vamos ver na prática?**

Abram o terminal e vamos trabalhar com  
repositório local



# OBRIGADO!

Duvidas?

Nosso canal oficial de comunicação é o Slack.

academico@geradordedevs.com.br  
<https://www.geradordedevs.com.br>

