Introdução ao Git

Quem sou eu?

Maria Luísa Moreno

Líder na área de Tecnologia na Escola DNC

Minha formação

Bacharel Interdisciplinar em Ciência e Tecnologia pela UNIFESP.

Project Management Full Stack - Formação em Projetos.

Minha carreira

Iniciação Científica na área de dados.

Embraer - Estagiária na área de Engenharia

DNC - Desenvolvedora Full Stack

DNC - Líder na área de Tecnologia

Introdução ao Git

Conteúdo do Curso

Introdução ao Git

O que é versionamento de código?

O que é git?

Introdução ao GitHub

O que é GitHub?

Criar uma conta no GitHub

Instalação do git

Comandos Básicos - pt1

Configurar o git

Vincular o git ao Colab

Principais comandos

Comandos Básicos - pt2

Juntar modificações

Ver histórico de modificações

Reverter modificações

Introdução ao Git

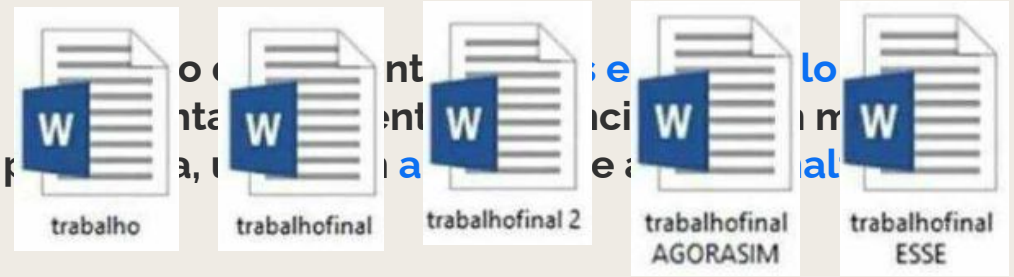
O que é versionamento de código?

O versionamento consiste em **estratégias para gerenciar as diferentes versões de um código, de um sistema ou de um modelo**. É uma forma de administrar as mudanças que são feitas e de garantir mais segurança na transição de uma versão para outra.

Garante mais segurança

Melhora a qualidade de código

Potencializa o trabalho em equipe



O que é git?

É um Sistema de Controle de Versionamento

1º

Mais utilizado no mundo

2005

Projeto de código aberto criado por Linus Torvalds.

Introdução ao GitHub

O que é GitHub?

- Serviço de colaboração de projetos pessoais e comerciais.
- Os desenvolvedores e contribuintes do projeto podem trocar ideias e sugestões para o andamento do código em questão.
- Como se fosse uma “rede social para programadores”.

“Rede social para programadores”

65 milhões

de desenvolvedores utilizam a plataforma

2008

Propriedade da Microsoft desde 2018.

Criar uma conta no GitHub

Passo-a-passo:

1. Acesse o site do GitHub.
2. Clique sobre “Sign up” localizado no topo à direita.
3. Preencha seus dados pessoais no formulário.
4. Clique sobre “Create account” após preencher os dados na etapa 3.
5. Acesse a página de login.
6. Insira suas credenciais cadastradas na etapa 3.
7. Clique em “Sign in”.

Pronto! Você tem uma conta no GitHub =)

Instalação do git

Antes de começar a usar o git, você tem que torná-lo disponível em seu computador.
Ele está disponível para Windows, Linux e Mac. Neste curso, faremos o passo-a-passo para instalar no Windows.
Demais sistemas, acesse a documentação aqui.

Passo-a-passo para instalar no Windows:

1. Acesse a página Download for Windows do próprio git.
2. Clique em “Click here to download”.
3. Clique no arquivo executável que foi baixado em seu computador.
4. Siga as instruções.

Instalação do Visual Studio Code

O Visual Studio Code é um **editor de código aberto** desenvolvido pela Microsoft. Ele está disponível para Windows, Mac e Linux.

A princípio ele é uma ferramenta muito simples, mas ele **possui uma loja de extensões imensa**, e que continua crescendo.

Para instalar:

1. Acesse a página de Download do Visual Studio.
2. Selecione seu sistema operacional.
3. Execute o arquivo baixado.

Comandos Básicos - parte 1

Configurar o Git

Utilizamos o comando ***git config*** para configurar seu nome e email no git.

Exemplo:

$ git config --global user.name "Seu nome para exibição"

$ git config --global user.email "seu-email@email.com"

Para vincular com algum repositório criado em seu GitHub, basta executar o comando:

$ git remote add origin URL_REPOSITORIO_REMOTO origin master

Configurar a chave SSH

Para enviar arquivos com seu login do GitHub e seu computador local, temos que configurar uma chave de acesso SSH.

Vá até um terminal do seu computador e digite:

$ cd ~/.ssh

$ ls

Se aparecer um arquivo chamado **id_rsa.pub**, você já tem uma chave cadastrada! Caso não, cria uma com o comando:

$ ssh-keygen -o -t rsa -C "your@email.com"

Configurar a chave SSH

Após gerada a chave, abra o arquivo com o comando:

$ cat ~/.ssh/id_rsa.pub

Copie seu conteúdo e vá até o GitHub e siga os passos:

1. Clique no seu perfil no canto superior direito e procure por “Settings”.
2. Clique em “SSH and GPG keys” no menu do lado esquerdo.
3. Clique me “New SSH key”.
4. Adicione um nome para sua chave em “Title”.
5. Adicione o conteúdo copiado anteriormente no campo “Key”.
6. Clique em “Add SSH key”.

Criar um repositório

Utilizamos o comando ***git init*** para inicializar um repositório como um projeto git.

Após este comando, o repositório **será reconhecido pelo git** como um projeto e responderá aos seus demais comandos.

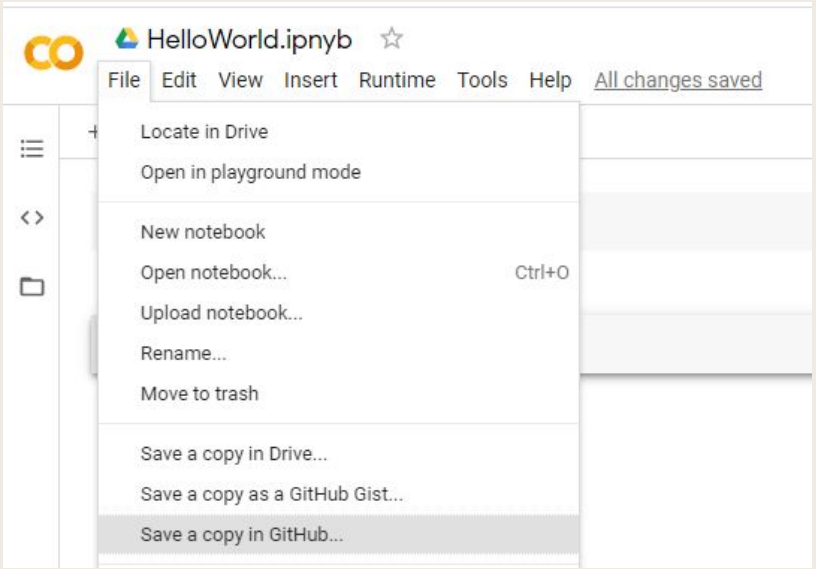
Exemplo:

$ git init

Vincular o Git ao Colab

Para conectar o seu Colab ao GitHub:

1. Clique em “File” no menu superior esquerdo.
2. Selecionar a opção “**Save a copy in GitHub**”.
3. Uma tela vai abrir com a solicitação para conectar ao seu GitHub. Clique em “Authorize googlecolab”.

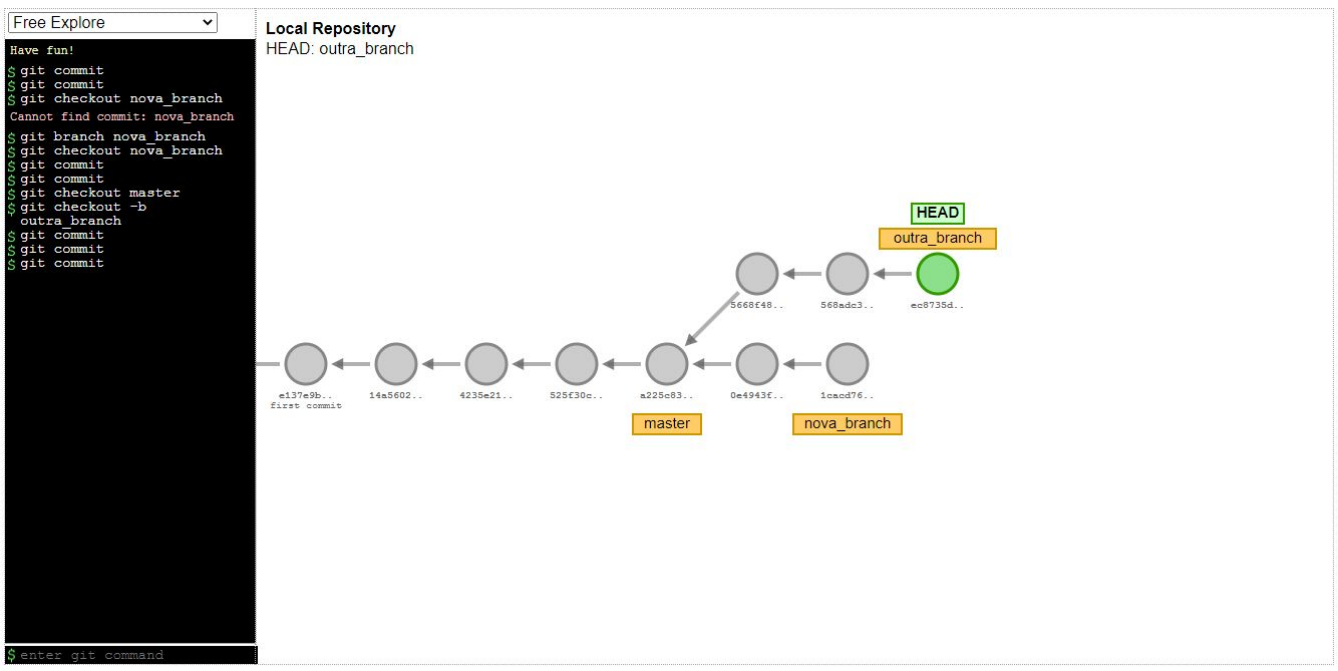


Vincular o Git ao Colab

4. Selecione o repositório e branch do GitHub.
5. Em **file path** encontramos o nome do arquivo, e como ele será chamado no GitHub.
6. Em **commit message**, temos a mensagem que será exibida ao publicar o notebook no GitHub.
7. Em **Include a link to Colaboratory** irá adicionar um link no arquivo para direcionar o usuário para o Notebook no Colab.
8. Clique em “OK”.

Visualizing git

https://git-school.github.io/visualizing-git/



git add

Esse comando Git adiciona os arquivos especificados de código ao seu repositório, sejam arquivos novos ou arquivos anteriores que foram alterados. Oferece diferentes possibilidades de sintaxe.

Exemplo:

$ git add seu_arquivo (esse comando irá adicionar o arquivo em específico ao repositório)

$ git add . (esse comando irá adicionar todos os arquivos novos e/ou modificados ao repositório)

git commit

É fundamental se estabelecer uma diferença entre git add e git commit:

- git add adiciona seus arquivos modificados à fila para serem submetidos a um commit posteriormente. Os arquivos não passaram por um commit.
- O git commit executa o commit dos arquivos que foram adicionados e cria uma nova revisão com um log. Por outro lado, se você não adicionar nenhum arquivo, o git não fará o commit de nada.

É possível combinar as duas ações em um único comando: **$ git commit -a**

Também é possível adicionar uma mensagem para a execução de um commit. Exemplo:

$ git commit -m “seu comentário”

git pull

O comando Git pull baixa o conteúdo do que foi alterado no repositório remoto para o seu repositório local e imediatamente atualiza seu conteúdo para a última versão.

Exemplo:

$ git pull <URL>

git push

Esse comando serve para subir suas modificações para um repositório remoto conectado anteriormente com git remote.

Exemplo:

$ git push –set-upstream <origin> <nome_do_branch>

git branch

É possível listar todas as branches criadas até o momento.

Exemplo:

$ git branch

Também é possível criar um novo branch.

Exemplo:

$ git branch nome_do_branch

git checkout

Exemplo:

$ git checkout <nome_do_branch>

Também é possível combinar operações, criando e fazendo o checkout de um novo branch com um único comando:

$ git checkout -b <nome_do_branch_novo>

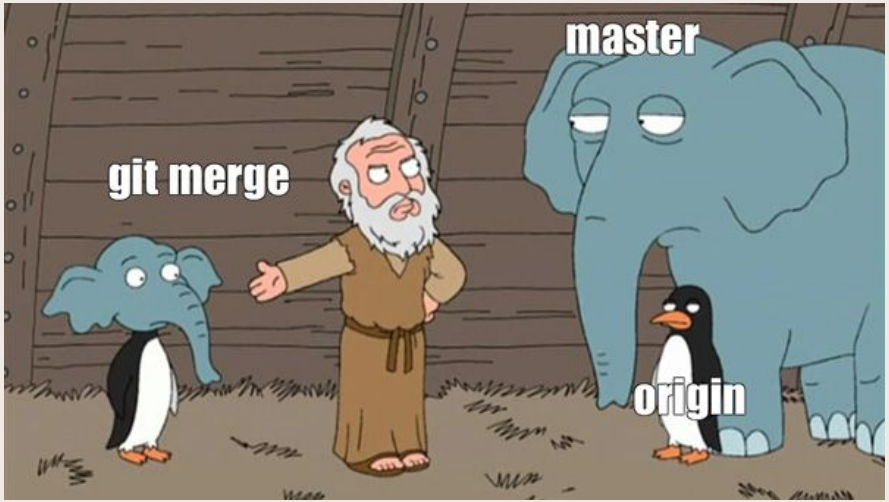
Comandos Básicos - parte 2

git merge

Esse comando Git integra as mudanças de dois branches diferentes em um único branch. Ele precisa ser iniciado a partir de um branch já selecionado, que será mesclado com outro, com o nome passado por parâmetro.

Exemplo:

$ git merge <nome_do_branch>



git rebase

Git rebase a princípio parece fazer o mesmo que um comando git merge: ele integra dois branches em um branch único. Porém, esse comando refaz o histórico de commits, tornando-o linear. É o mais indicado para consolidar múltiplos branches.

Exemplo:

$ git rebase <base>

git log

Podemos acessar um log de modificações feitas no projeto.

Exemplo

$ git log

git revert

Para desfazer uma modificação é muito simples. Ao executar o comando que vimos anteriormente, o ***git log***, nós temos uma codificação vinculado ao commit, chamado ***hash***.

Exemplo:

$ git revert <hash_commit>

Boas práticas para o seu Portfólio no GitHub

Boas Práticas

Um bom projeto necessita de **um bom README**. Um arquivo README bem escrito é um incrível atrativo para projetos Open Source.

Estrutura de um bom README:

- Informar ferramentas utilizadas no projeto;
- Imagem demonstrativa do projeto juntamente como o nome do projeto;
- Texto com resumo da proposta do projeto;
- Lista dos requisitos do projeto;
- Demonstração da Aplicação;
- Instruções para visualizar o projeto;
- Referências e observações.

Exemplos

https://github.com/EduardooPV/portifolio

https://github.com/marialuisamoreno/case-dinamica