**Ferramentas Git e Github**

Grupo: Vinicius Morais, Vitor França e Rafae

- O que é Git?

O Git é o coração da ferramenta GitHub e é usado como um sistema de controle de versão sendo desenvolvido por Linus Torvalds (o mesmo criador do sistema operacional Linux)

Quando falamos em "sistema de controle de versão" devemos entender que aquele programador e /desenvolvedor em conjunto pode administrar o código-fonte do projeto e seu histórico de modificações utilizando "ferramentas de linha de comandos de Git**".**

# L., A. (27 de Janeiro de 2023). Oque é Github e como usá-lo? Acesso em 7 de Junho de 2023, disponível em Hostinger Tutoriais: https://www.hostinger.com.br/tutoriais/o-que-github#:~:text=Em%20suma%2C%20o%20GitHub%20%C3%A9,registro%20detalhado%20do%20seu%20progresso.

**Comandos básicos do Git**

| git config

Esse é um dos códigos mais usados pelos usuários do Git e é aplicado para definição dos valores de configuração específicos do usuário como e-mail, nome, o formato de arquivo, etc.

Exemplo:

git config --global user.email

sam@google.com

| git init

Este comando seria específico para a criação de um novo repositório Git

OBS - Os repositórios Git eles são responsáveis pelo armazenamento dos arquivos do seu projeto, podendo ser possível guardar códigos, imagens e áudios.

Exemplo:

git init

| git status

O git status é um comando git que nos auxilia a identificar a situação de um determinado arquivo em específico, caso algum desses estiverem por exemplo fora da staging area, se alterado, será exposto que aquele foi modificado e necessita ser comitado.

Exemplo:

git status <NomeDoArquivo>

| git add

Nesse caso esse comando é utilizado para acrescentar e /adicionar arquivos ao índice. TODOS os modelos de arquivos que tiveram o uso do git add, estão obrigatoriamente na Staging Area que seria nada mais do que um espaço aonde você deixaria uma prévia do arquivo e quando você tivesse certeza, ele seria jogado para o repositório com outra forma de comando.

Exemplo:

git add <NomeDoArquivo>

ou

git add .

| git commit

O comando git commit é usado para mudanças na "cabeça" do projeto. Entretanto não causa nenhum tipo de alteração no repositório remoto (local da nuvem ou servidor). Ele serve como um meio de “registrar todos os dados do seu projeto”, passa da Staging Area ao Repository (diretório)

Exemplo:

git commit -m "Adicione uma mensagem aqui"

Caso você já tenha usado o comando git add no arquivo que está sendo comitado, você pode mandá-lo diretamente ao repositório sem necessariamente passar pela Staging Area.

| git push

O git push é uma maneira de enviar todo nosso e trabalho e /projeto de commits ao repositório remoto. Ele atualiza o remoto e envia a informações localmente, fazendo que outros usuários possam analisar todas as alterações feitas por você.

| git rm

Esse é responsável pela remoção dos dados do projeto, ele é aplicado quando você especifica o tipo de arquivo na qual quer ser deletado. Abaixo, temos um modelo prático.

git rm <NomeDoArquivo>

| git reset

O git reset é o comando sempre devemos lembrar da ideia de “resetar”, ou seja do conceito de voltar ou retornar a um determinado ponto. Quando aplicamos esse comando git nós podemos desfazer alterações que fizemos no passado em diferentes tipos de situações. Supondo que você comitou um arquivo que foi alterado, porém você quer deletar esse commit e voltar ao último registro, desejando apagar também a modificação interna dele. Você pode utilizar o seguinte comando:

Exemplo:

git reset –hard <IdDoCommit>

| git branch

A git branch é um comando que pode ser utilizado de diversas maneiras. Você pode usá-lo para listar todas as branches presentes na sua aplicação e até mesmo criar novas.

Antes disso precisamos separar os tipos de branches (principal e alternativa)

A branch principal seria a linha de percurso do seu projeto, isto é, é aonde você vai aplicar as informações do projeto sendo única e “protagonista”.

A branch alternativa é também uma linha do projeto paralela a principal. Quando relacionada a primária, as informações aplicadas nelas são jogadas a “master”, então para compreendermos melhor devemos entender como uma ramificação do nosso projeto, em suma, uma sub-divisão.

Exemplo:

git branch (listará todas as branches)

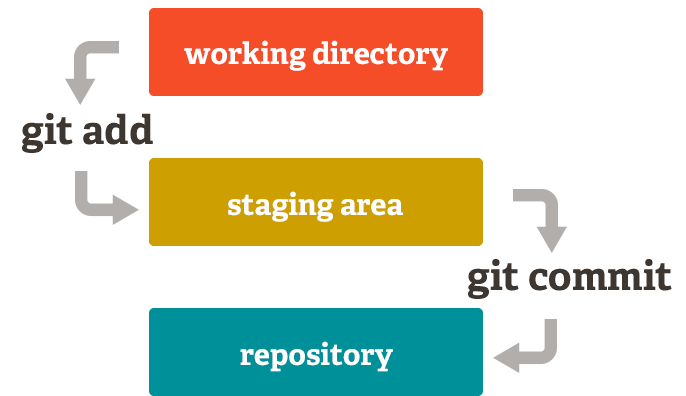
git branch <NomeDaBranch> (criará uma nova)

Caso nossa intenção é juntar todas essas branches já criadas, precisamos aplicar um outro tipo de comando conhecido como “git merge”.

| git merge

A git merge como já falado conceitualmente, é usada para o ligar ou relacionar uma branch com a outra. Esse tipo de comando é muito relevante quando por exemplo estamos desenvolvendo um trabalho em grupo utilizando um mesmo arquivo. Caso exista algum tipo de mudança, todas as alterações feitas pelos integrantes são aplicadas na branch principal (caso estejam relacionadas)

# Silva, C. L. (28 de Agosto de 2021). Top 25 comandos do Git. Acesso em 9 de Junho de 2023, disponível em CódigoFonte: https://www.codigofonte.com.br/artigos/top-25-comandos-do-git



**O que é GitHub?**

O Github tem sim muita relação com o Git. GitHub é uma plataforma para gerenciar seu código e criar um ambiente de colaboração entre devs, utilizando o Git como sistema de controle. Ele vai facilitar o uso do git, escondendo alguns detalhes mais complicados de setup. É lá que você provavelmente vai ter seu repositório e usar no dia a dia. O GitHub foi desenvolvido principalmente por Chris Wanstrath e atualmente sua dona é a Microsoft.

O sistema web que ele possui permite que você altere arquivos lá mesmo, apesar de não ser muito aconselhado, pois você não terá um editor, um ambiente de desenvolvimento e de testes. Para se comunicar com o github e mexer nos arquivos do seu repositório, você pode usar a linha de comando, usando o comando do git e suas diretivas de commit, pull e push. Parece assustador? Há uma alternativa: usar um aplicativo desktop mais intuitivo, o GitHub Desktop, veja a seguir.

# Silveira, P. (7 de Junho de 2023). *Alura*. Acesso em 9 de Junho de 2023, disponível em O que é Git e Github: como configurar e primeiros passos: https://www.alura.com.br/artigos/o-que-e-git-github

**O que é um repositorio remoto?**

Repositório, ou repo, é um diretório onde os arquivos do seu projeto ficam armazenados. Ele pode ficar em um depósito do GitHub ou em seu computador. Você pode armazenar códigos, imagens, áudios, ou qualquer outra coisa relacionada ao projeto no diretório.

# L., A. (27 de Janeiro de 2023). *Oque é Github e como usá-lo?* Acesso em 7 de Junho de 2023, disponível em Hostinger Tutoriais: https://www.hostinger.com.br/tutoriais/o-que-github#:~:text=Em%20suma%2C%20o%20GitHub%20%C3%A9,registro%20detalhado%20do%20seu%20progresso.