Git e Github

Git

**Git** é um sistema de controle de versão open-source, ou seja, gratuito. Ele é **utilizado para a criação de um histórico de alterações em código-fonte de projetos de desenvolvimento de software**. Foi desenvolvido por Linus Torvalds, o criador do sistema operacional Linux.

Por meio de sua utilização, podemos saber quais foram as alterações realizadas, quem fez cada uma das alterações e baixar essas mudanças em nossa máquina. Se necessário, revertê-las para uma versão anterior.

github

O Github é um **repositório remoto, criado como um serviço online de hospedagem de repositórios do Git.**

Funciona como um servidor que agrega todas as modificações realizadas por cada uma das pessoas envolvidas em um projeto, unificando as diferentes versões de código e seus históricos, permitindo compartilhamento entre as equipes.

Outro ponto interessante é a sua utilização como currículo. Muitas empresas pedem na [entrevista](https://blog.betrybe.com/carreira/como-se-preparar-para-uma-entrevista/) o link do GitHub para verificar seus projetos e entender suas capacidades de [programação](https://blog.betrybe.com/carreira/programacao-para-iniciantes/).

Os conceitos do Git

São vários conceitos dentro do Git que nos permitem manipular o código-fonte dentro de suas inúmeras versões. Vamos listar os principais deles para que você possa se familiarizar.

Repositório

O primeiro é o repositório, local físico onde os arquivos e suas cópias ficarão armazenados. O repositório pode ser local ou remoto, podendo salvar não apenas arquivos de texto, mas também imagens, áudios e outros elementos relacionados ao projeto.

Branch

Branches são os ramos, cópias do código original que podem ser manipuladas de forma livre pela pessoa que trabalha em programação, sem afetar as funcionalidades em produção no código-fonte.

Isso **permite que todas as alterações sejam realizadas de forma segura**, sem que erros ocorram na cópia principal do projeto.

Merge

Após a finalização de um trabalho em um Branch, é necessário realizar o Merge, fundir a cópia e seus arquivos modificados com o ramo principal do projeto. Isso acontece apenas no repositório local para que ajustes possam ser feitos.

Push Request

O Push Request é o envio das modificações após o Merge para o repositório central, para que todas as outras pessoas que atuam no desenvolvimento possam atualizar suas cópias e revisar o código criado, verificando conflitos com seus próprios trabalhos.

Pull Request

O Pull Request é utilizado quando outra [pessoa que atua no desenvolvimento](https://blog.betrybe.com/carreira/desenvolvedor-de-aplicativos/) muda o ramo principal no repositório central, puxando as modificações realizadas para a sua máquina, fundindo a nova versão com o seu código local.

Fork

Trata-se do comando de cópia de um repositório remoto para a máquina local, realizado sempre que vamos começar a trabalhar em um projeto que já existe. Também é usado para pegar um código público para posterior modificação e utilização em um programa interno.

## **Principais comandos Git**

Se você estiver familiarizado com alguns dos principais comandos do GIT, seus primeiros passos na ferramenta podem se tornar mais descomplicados.

Para isso, trouxemos abaixo uma lista dos comandos mais utilizados pelos programadores(as) e o que eles significam.

* **Init**: este comando dá origem a um repositório novo, local ou remoto, ou reinicializa um repositório já existente;
* **Clone**: este comando clona o código de um repositório para sua manipulação em outro ambiente;
* **Commit**: este comando move os arquivos da state area para um repositório local;
* **Add**: este comando adiciona um arquivo alterado a uma staging area, ou seja, o prepara para ser vinculado a um commit;
* **Push**: este comando envia arquivos de um repositório local para um repositório remoto. No GitHub, por exemplo;
* **Pull**: ao contrário do push, este comando traz um arquivo do repositório remoto para o repositório local.
* **Merge**: este comando serve para unir arquivos alterados ao arquivo original de um projeto. Em outras palavras, é ele quem une os branchs as commits.
* **Log**: este comando permite a visualização do histórico de commits de um arquivo ou usuário, ou o acesso de uma versão específica.

Instalar <https://desktop.github.com/>

Criar conta

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Vc pode baixar algum repositório pronto(repositório e onde tem o projeto de alguém)

Clone – copia de alguém

Create ou add – são seus projetos

No clone

Tela de um aparelho eletrônico

Descrição gerada automaticamente

Vc tem que colocar o endereço da url e o local onde vc quer gravar a pasta

Após tem que criar uma nova “Branch” para poder alterar o projeto e salvar