


# O que é Git? O que é versionamento?

<input checked="" type="checkbox"/> Finalizada?	<input checked="" type="checkbox"/>
 Comentários	O que é Git? O que é GitHub? O que é VCS? O que é controle de versão? O que é versionamento de software?

## Pra quê que serve o Git e Github?

### Git

- Software de controle de versão (VCS)
- Trabalha então, com o versionamento:
  - Programadores trabalham constantemente com várias versões ao mesmo tempo;
  - Ter um VCS é como ter uma máquina do tempo em que eu consigo voltar o tempo de todas as modificações no projeto.

### Github

- Plataforma de Rede Social para Programadores

## Histórico dos VCS

### Centralizado/ linear

- Primeiro VCS lançado em 1972
  - Tudo o que é feito em um projeto é feito e logo em seguida “commitado”

### Distribuído

- Versão mais moderna do primeiro
- Ao invés de commitar diretamente para o repositório central (seja remotamente ou pela internet), os



*Commit* é o termo para enviar uma versão do software o repositório central. Para isso, é necessário ter conexão constante com o repositório central.

usuários commitavam para um repositório local



*Repositórios locais* são softwares instalados no computador que atuam como um repositório central.

- Quem vai fazer essa organização (do que era e do que ficou) não é o usuário, mas sim o servidor que está cadastrado o código.

## Como que os repositórios locais trocam informações com outros repositórios?

- Com os modelos distribuídos, temos os **repositórios remotos**, que são como se fosse o Google Drive do seu projeto.
- Ao finalizar um projeto, ou uma versão daquele dia, é necessário que haja um **push** que é enviar para o repositório remoto a versão que está sendo trabalhada e atualizada no repositório local.

## Por que usar softwares de controle de versão?

- **Controle de histórico:** é a máquina do tempo. A todo momento conseguimos retornar a qualquer ponto do código, quais foram todas as modificações que foram feitas ao longo do tempo, dos dias, etc.
- **Trabalho em equipe;**
- **Ramificação de projeto:** dessa forma, se consegue criar e separar diferentes áreas do projeto para ser implementado; por exemplo, dividir equipes para trabalhar no front, no back-end do mesmo projeto. Ao final, com o **merge** consigo mesclar todas as alterações na master do projeto.
- **Segurança:** com as ramificações, o cara de um projeto não se interfere no projeto de outro.

- **Organização;**