**Software é feito de forma colaborativa.**

**Git** - criado em 2005 pelo Linus Torvalds - sistema de versionamento de código distribuído

* Ajuda a gente a criar e monitorar diferentes versões do nosso código
* Open source

**Git Bash -** o Git Bash é um terminal estendido para otimizar o uso do Git

Se você já tem um software para cuidar da parte de versionamento do código, onde você guarda? (abaixo)

**GITHUB**

- Empresa da Microsoft

* Repositório de código
* Podem ter planos pagos

GIT e GITHUB são diferentes, complementares porém diferentes.

*Comandos básicos para um bom desempenho no terminal - WINDOWS*

**dir** - listar, entender o local em que estamos - traz lista de diretórios na pasta na qual estamos situados

**cls** (clear screen) - apaga o que está na tela

**cd** (change directory) - ir para uma pasta específica - *Ex: cd windows*

**cd ..** - voltar uma ação, voltar para o repositório anterior

**mkdir** - criar uma pasta

**echo** - printa de volta o que eu digitei / Usa para criar um arquivo dentro de uma pasta *Ex: echo hello > hello.txt*

**del** **+ nome pasta** - deleta arquivos (que estão nas pastas)

**rmdir + nome pasta /S /Q** - remove o repositório (pasta)

**ls** - list

**ls -a -** Flag -a mostra repositórios ocultos

**git init -** iniciar o repositório do Git

**git status** -

**mv** - mover

**git add nomeArquivo**

**git add .**

**git add \*** - com o asterisco pega tudo o que foi modificado e add para o stage pra podermos comitar

**git commit -m** - criar “imagem”, é como se fosse uma foto do nosso código. A flag -m é para passar uma mensagem.

**git push origin master -** empurrar para o github

**git pull -** “puxar” o que está no github para a sua máquina

**git clone + url do github** - quando clona ele já vem como repositório

**git remote -v**

**GIT - Por baixo dos panos** - Git é um sistema distribuído seguro

SHA1 - Secure Hash Algorithm (40 caracteres identificador)

Algoritmo de encriptação - “vai pegar o seu arquivo e embaralhar” - Forma curta de representar um arquivo.

Objetos internos do GIT - Objetos fundamentais

Blob (bolha) - contém metadados do Git - nome objeto, tamanho, entre outros

Tree - armazenam e apontam para blobs e pode apontar para outra árvore- também contém metadados

Commit

Aponta para um tree (árvore), para um parente, para o autor

Os commits também possuem um SHA1

Commit é único para cada autor.

Se você mudar qualquer coisa em um arquivo, vai mudar na árvore e consequentemente no commit.

**Chave SSH** - forma de estabelecer uma conexão segura e encriptada entre duas máquinas.

**Token de acesso pessoal**

Markdown (.md)- forma mais humanizada de escrever HTML / estrutura para mexer no texto (deixar ele mais bonito) **Exemplo: # = h1 no html**

**QUANDO USAMOS GIT INIT - CRIA-SE UM REPOSITÓRIO**

**Inmodified - não modificado (está no repositório do git e ainda não sofreu modificação)**

**Modified - modificado**

**Staged - arquivos que estão se preparando para fazer parte de outro tipo de agrupamento**

**Quando você “commita” ele volta pra Unmodified**

Quando você faz o commit você tira os arquivos do staging e coloca como não modificado e você POPULA o seu repositório local. O repositório local só aceita COMMIT.

**CONFLITOS**

São comuns quando usamos versionamento de código.

Conflito de Merge (resolve manualmente) - você tem que resolver o conflito que ocorreu na mesma linha, por exemplo. Abre o arquivo e vê qual alteração fica.

Usar botão verde CODE - para baixar o código zipado ou selecionar o link para clonar o código.