

Git

Definição

Git é um sistema de controle de versão distribuído que rastreia alterações em qualquer conjunto de arquivos de computador. Seus objetivos incluem velocidade, integridade de dados e suporte para fluxos de trabalho distribuídos e não lineares.

VCS(Software de controle de Versão) - Git

Versionamento: uma máquina do tempo que você pode sempre voltar em qualquer momento do seu projeto.

Existem dois tipos de VCS:

- Centralizado/Linear
- Distribuído

Principais Vantagens de utilizar um VCS

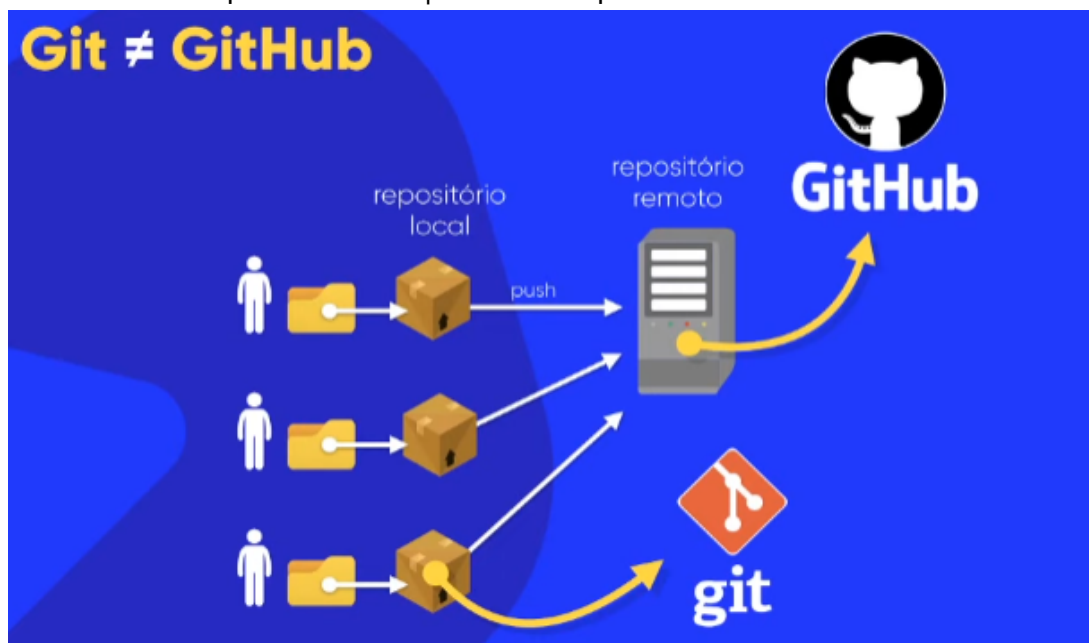
- Controle de Histórico
- Trabalho em equipe
- Ramificações do projeto
- Segurança
- Organização

Github

Plataforma de rede social para programadores - Github

Vantagens:

- Repositórios ilimitados
- Hospedagem de código-fonte
- Característica de Rede Social
- GitHub Pages integrado
- Forks
- Git > Repositório Local | Github > Repositório Remoto



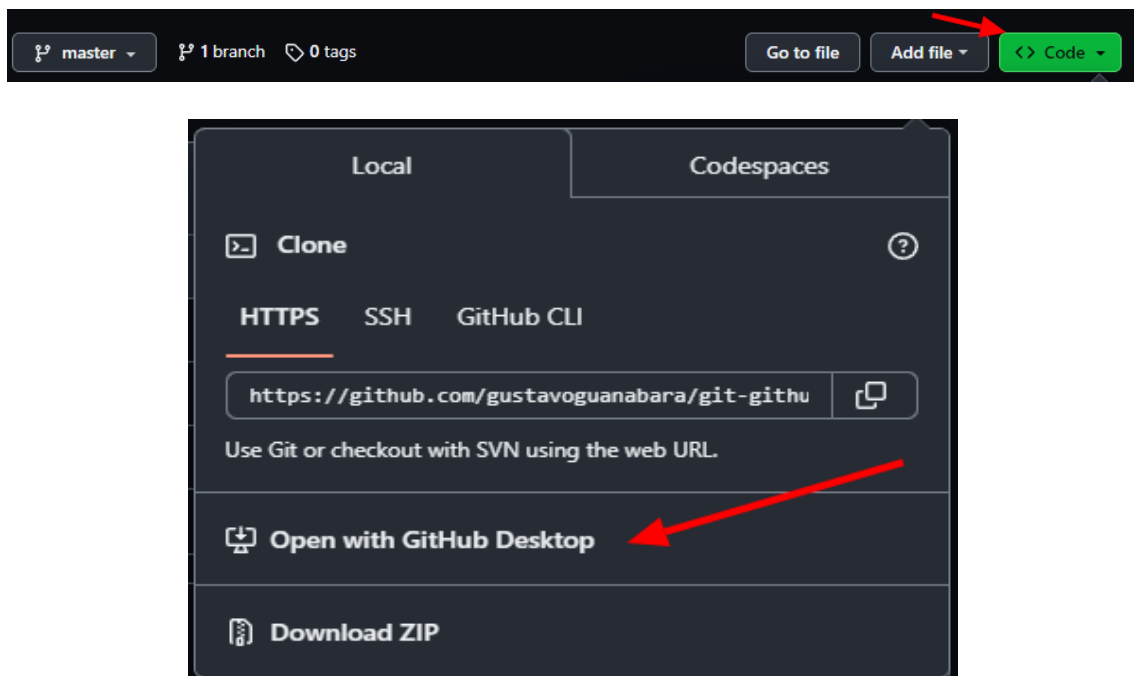
Software utilizados:

- ❖ Git
- ❖ Github
- ❖ Vscode

Clonando um Repositório

Repositório é onde o código está versionado, pode ser público ou privado.

Clonagem de Repositório: trazer os arquivos do servidor remoto para meu pc e acessar os arquivos de forma local por causa do clone do github.



Versionamento de Projetos Antigos

Cria um **Novo repositório local**, publica-o no github, depois encaminha os **uploads** no **repositório local** e faz uma **push** após.

Issues

Um issue no GitHub é uma espécie de rastreador de bugs para um projeto que tem sua própria seção em cada repositório. Ele pode ser usado para discussões que incluem uma variedade de coisas, como relatar um bug nesse repositório, fazer perguntas ou até mesmo pedir um recurso específico. É um ótimo lugar para manter um controle de tarefas e melhorias do projeto.

Issues: são um meio para os programadores que estiverem no mesmo projeto possam interagir e resolver dentro da aba issues no repositório que esteja trabalhando.

Utiliza-se o @ para fazer referência a algum usuário que tenha proposto ou aberto uma issue.

Linguagem Markdown

Linguagem de marcação; .md;

Onde aplica markdown? readme.md; issues; pull request

Comandos básicos:

- Transforma a palavra ou frase em negrito; **** **** ou **__ __**
- Transforma a palavra ou frase em itálico; *_ _* ou ** **
- Deixam a palavra riscada: ~~~~ ~~~~
- Dependendo da quantidade de # pode mudar o nível do título: # Titulo <- Titulo de nível 1; ## Titulo <- Titulo de nível 2
- Gera uma linha *******
- Lista numerada
- Lista demarcada
- Lista de Tarefas
- Tabela: segue o mesmo padrão porém remove os valores

```
nome|numero|nota
---|---|---
Guilherme|2|6,7
Virgilio| 6 |10
```

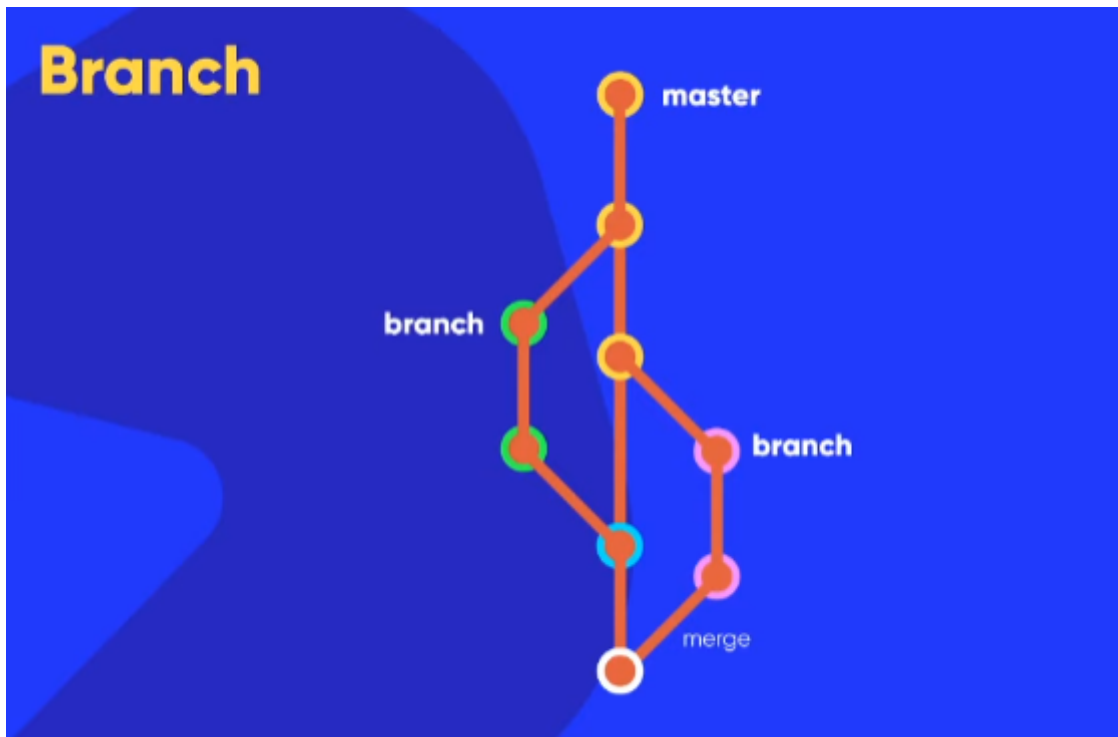
nome	numero	nota
Guilherme	2	6,7
Virgilio	6	10

- Para destacar comandos utilize duas crase ``
- Para escrever um código de uma determinada linguagem utilize crases triplas: ``` ```
- Quote reply : >

```
> Será que vai chover??
```

```
| Será que vai chover??
```

Branches ou Ramificações



Branch: são bifurcações ou ramos do seu projeto principal, visando a criação de uma nova versão sem modificar o projeto principal, modificando apenas depois que estiver tudo funcionando como deveria.

Master: é o ramo principal, é o produto final ou última versão do seu produto ou projeto.

Merge: junta o branche com o master(projeto principal)
é um evento que ocorre no desenvolvimento de software quando um colaborador/desenvolvedor está pronto para iniciar o processo de mesclagem de novas alterações de código com o repositório principal do projeto.

Macetes e Definições

- Início do trabalho clica em Fetch(buscar);
- Para saber se uma pasta estar versionada basta ativar os itens ocultos, ou adicionar no final do caminho do arquivo \.git;
- **Commit:** salva o código no servidor local; commit registra alterações em um ou mais arquivos no seu branch
- **Pull:** envia os dados do servidor remoto para meu servidor local;
- **Push:** enviar dados para o servidor remoto;
- **Branch:** ramificações do versionamento;
- **Git ignore:** gitignore informa ao Git quais arquivos devem ser ignorados ao confirmar seu projeto(push) no repositório GitHub;
- **Issues:** é uma espécie de rastreador de bugs para um projeto que tem sua própria seção em cada repositório;
- **Pull request:** As solicitações de pull permitem que você informe outras pessoas sobre as alterações por push feitas em uma ramificação de um repositório no GitHub. Após a abertura de uma solicitação de pull, você poderá discutir e revisar as possíveis alterações com colaboradores e adicionar confirmações de acompanhamento antes que as alterações sejam mescladas na ramificação base;
- **Fork:** Um fork é um novo repositório que compartilha configurações de código e visibilidade com o repositório “upstream” original. Os Forks costumam ser usados para iterar ideias ou alterações antes de serem propostas de volta ao repositório upstream, como em projetos de código aberto ou quando um usuário não tem acesso de gravação ao repositório upstream;