DEVSUPERIOR

Formação Desenvolvedor Moderno
Módulo: Git e Github

Capítulo: Introdução ao Git e Github

https://devsuperior.com.br

1

Favor instalar o Visual Studio Code

https://code.visualstudio.com/download

Vídeo instalação Windows:
https://youtu.be/ZIoHacwWjLI

Vídeo instalação Ubuntu:
https://youtu.be/THdaC99oNxw

2

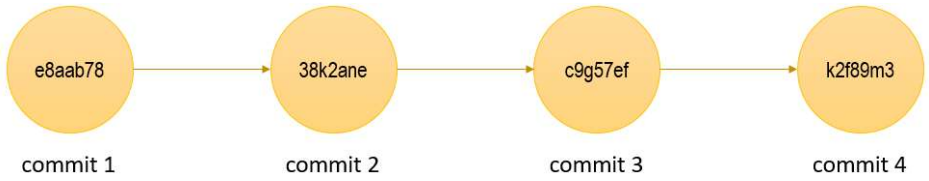
Visão geral de Git e Github

https://devsuperior.com.br

Prof. Dr. Nelio Alves

Git

GIT - é um sistema de versionamento: você controla as modificações de um projeto por meio de versões chamadas "commits".



Github

É um serviço online de hospedagem de repositórios Git remotos.

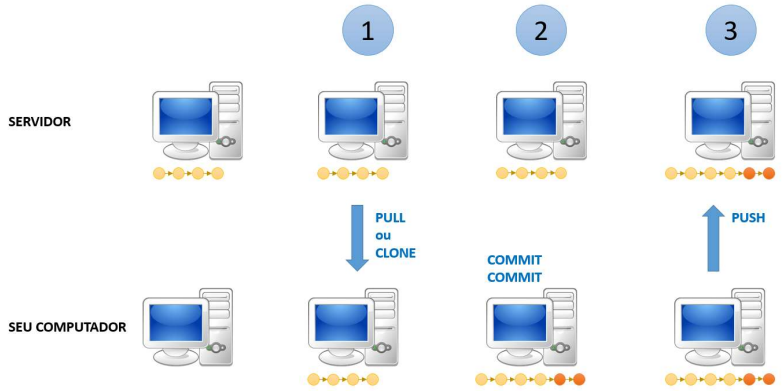
- Possui uma interface gráfica web: github.com
- É uma plataforma social (usuários, página de perfil, seguidores, colaboração, etc.). Dica: currículo!
- Maior serviço do mundo de hospedagem de projetos de código aberto
- Modelo de cobrança: gratuito para projetos de código aberto, pago para projetos privados
- Alternativas: BitBucket, GitLab, etc.

Repositório remoto e local

Um projeto controlado pelo Git é chamado de **repositório** de versionamento.

Tipicamente uma cópia "oficial" do repositório fica salvo em um **servidor** (**repositório remoto**).

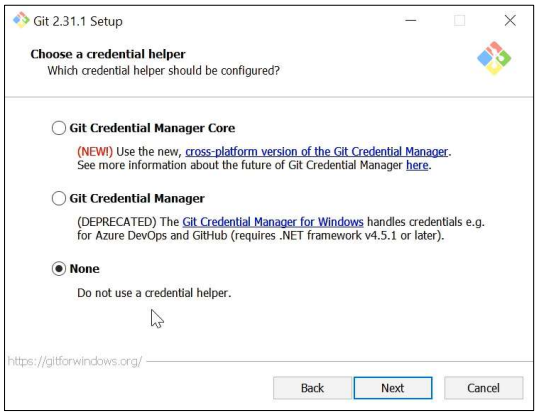
Cada pessoa que trabalha no projeto pode fazer uma cópia do repositório para seu computador (**repositório local**). A pessoa então faz suas alterações no projeto (novos commits) e depois salva as alterações no servidor.



Instalação do Git no computador

https://git-scm.com/downloads

Importante: não escolha Gerenciador de credenciais



Configurando sua identificação no Git

```
git config --global user.name "Seu nome"
git config --global user.email "Seu email de cadastro do Github"

git config --list
```

Configuração para ver arquivos ocultos (Windows)

Iniciar -> Opções do explorador de arquivos

DESMARCAR: "Ocultar as extensões dos tipos de arquivos conhecidos"

MARCAR: "Mostrar arquivos, pastas e unidades ocultas"

Configurar chave SSH para o Github

SSH é um protocolo para comunicação de dados com segurança.

O Github aboliu a autenticação somente com usuário e senha.

A ideia básica é cadastrar previamente quais computadores podem acessar o Github em seu nome. Outros computadores não conseguem acessar.

Para isto você deve:

(1) Gerar uma chave SSH no seu computador
(2) Cadastrar essa chave no seu Github

Passo a passo: salvar primeira versão de um projeto no Github

Considerando que agora seu ambiente já está todo configurado (usuário e email, visualização de arquivos ocultos, chave SSH), sempre que você criar um novo projeto, os passos básicos serão estes (troque os parâmetros em **azul** pelos seus dados):

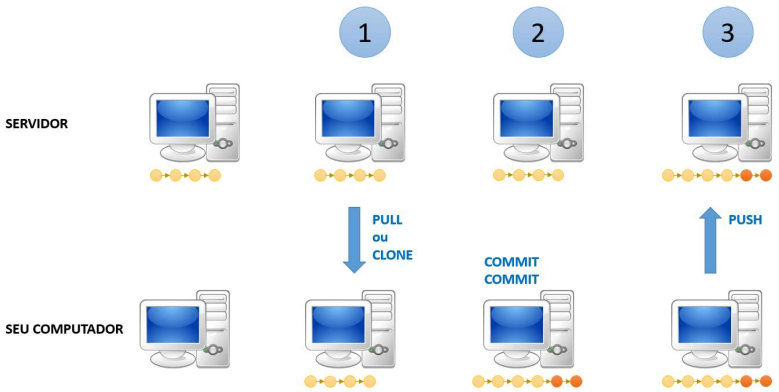
```
git init

git add .

git commit -m "Mensagem explicativa"

git branch -M main

git remote add origin git@github.com:seuusuario/seurepositorio.git

git push -u origin main
```


Passo a passo: salvar uma nova versão

```
git status

git add .

git commit -m "Mensagem explicativa"

git push
```

Demo: clonar e modificar um projeto de um repositório remoto que você tem permissão para alterar

```
git clone git@github.com:seuusuario/seurepositorio.git
git add .
git commit -m "Mensagem explicativa"
git push
```



Verificando o histórico de versões

```
git log
```

Listagem resumida:

```
git log --oneline
```

Git status, git add e stage

Git diff

- Comando que mostra a diferença entre arquivos modificados
- Dica: utilizar o VS Code, que mostra graficamente as diferenças

Git checkout

- Permite modificar temporariamente os arquivos do projeto ao estado de um dado commit ou branch

- Código do commit, HEAD
  - Cada commit possui um código, que pode ser utilizado para referenciar o commit
  - O último commit do histórico do branch corrente também pode ser referenciado pela palavra HEAD
  - É possível referenciar um commit N versões antes de HEAD usando ~N, por exemplo:
    - HEAD~1 (penúltimo commit)
    - HEAD~2 (antepenúltimo commit)

- IMPORTANTE: antes de fazer o checkout para voltar para HEAD, certifique-se de que não haja mudanças nos arquivos. Se você acidentalmente mudou alguma coisa, desfaça as modificações usando:

```
git reset
git clean -df
git checkout -- .
```


Arquivo .gitignore

- É um arquivo que indica o que NÃO deve ser salvo pelo Git.

- Geralmente o arquivo .gitignore fica salvo na pasta principal do repositório. Mas também é possível salvar outros arquivos .gitignore em subpastas do repositório, para indicar o que deve ser ignorado por cada subpasta.

Casos comuns de arquivos que não devem ser salvos pelo Git:

- Arquivos compilados
  Linguagens compiladas (C, C++, Java, C#, etc.) geram arquivos de código compilado para executar o programa localmente.
- Arquivos de bibliotecas externas usadas no projeto
  Projetos reais utilizam bibliotecas externas (programas prontos disponíveis na Internet). Por exemplo, projetos JavaScript com NPM tipicamente salvam uma subpasta "node_modules" na pasta do seu projeto.
- Arquivos de configuração da sua IDE
  IDE's podem salvar uma subpasta com arquivos de configuração na pasta do projeto (exemplo: .vscode).
- Arquivos de configuração do seu sistema
  Por exemplo, sistemas Mac podem gravar uma subpasta .ds_store na pasta do projeto.

Exemplos de projetos com .gitignore

https://github.com/acenelio/composition1-java

https://github.com/acenelio/dsmovie