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

Aula 96

Visão geral de Git e Github

https://devsuperior.com.br

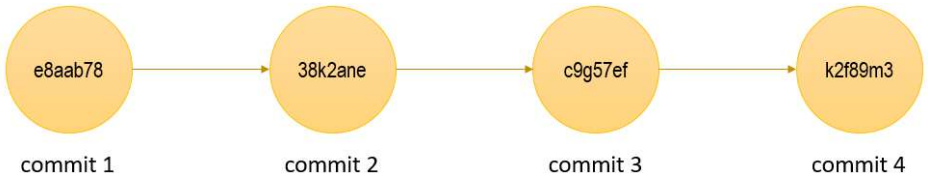
Prof. Dr. Nelio Alves

3

Aula 97

Git

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



Versões no Git são chamadas de commit

4

about.gitlab.com

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.

ATLASSIAN Bitbucket

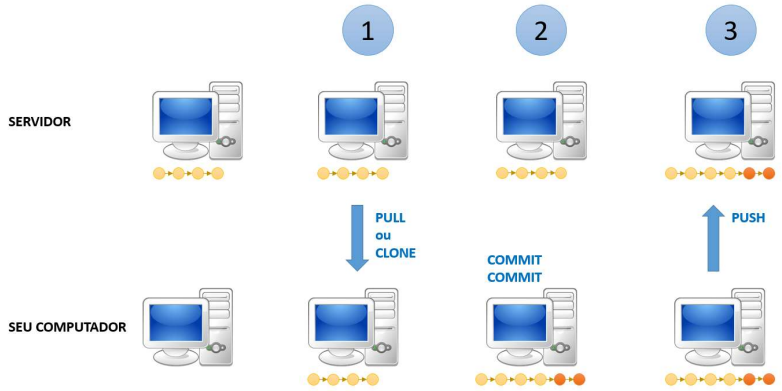
5

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.



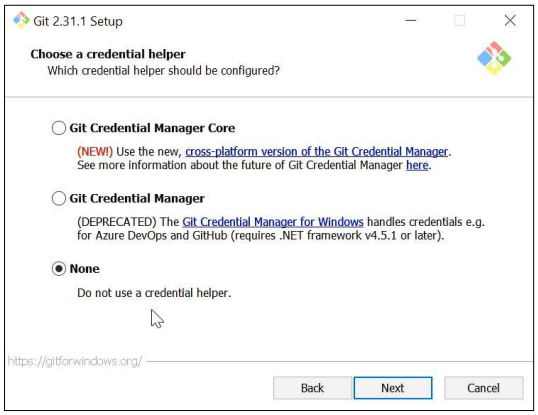
6

Aula 98

Instalação do Git no computador

https://git-scm.com/downloads

Importante: não escolha Gerenciador de credenciais



7

Aula 99

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
```

8

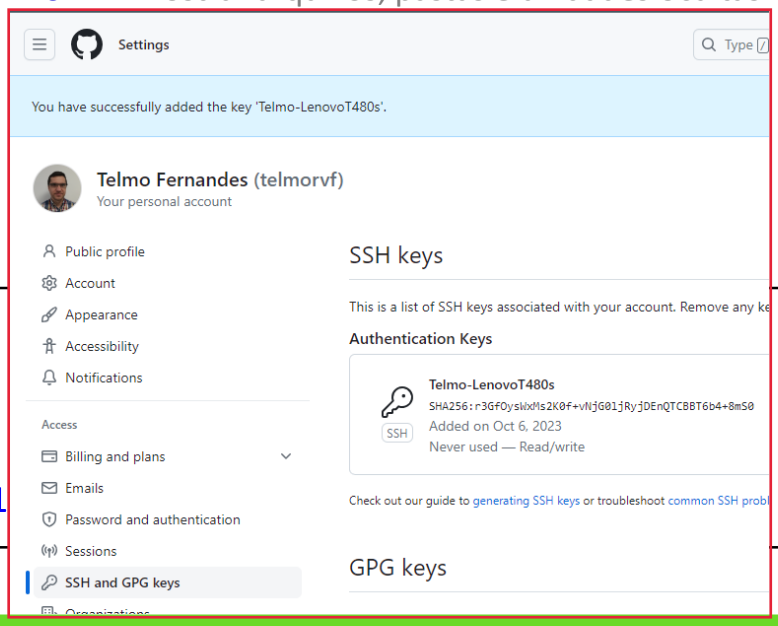
Aula 100

Configuração para ver arquivos ocultos (Windows)

Iniciar -> Opções do explorador de arquivos

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

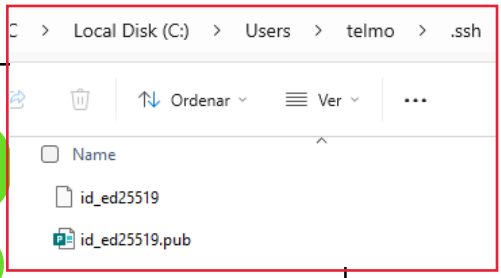
MARCAR: "Mostrar arquivos, pastas e unidades ocultas"



9

Aula 101

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: https://docs.github.com/pt/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent

(1) Gerar uma chave SSH no seu computador ssh-keygen -t ed25519 -C "your_email@example.com"
(2) Cadastrar essa chave no seu Github

10

Aula 102

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                                  =>  Initialized empty Git repository in C:/

git add .                                  => Add projecto do stage area

git commit -m "Mensagem explicativa"      => create new version "commit no github.com"

git branch -M main                         =>

git remote add origin git@github.com:telmorvf/Github.git

git push -u origin main    => Envia dados para o Github só na primeira vez "-u origin main"
```

list files: $ ls

11

```
$ echo "# Github" >> README.md
$ git init  =>  Initialized empty Git repository in C:/
$ git add .
$ git status
$ git commit -m "mensagem explicativa"
$ git status
$ git branch -M main
$ git remote add origin git@github.com:telmorvf/Github.git
$ git push -u origin main (só na primeira vez "-u origin main")
```

Aula 103

Passo a passo: salvar uma nova versão

```
$ git status

git status

git add .

git commit -m "Mensagem explicativa"

git push
```

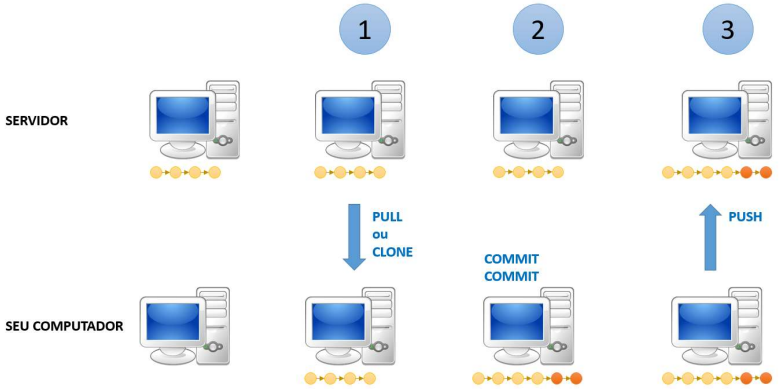
```
Set the user name for the current repository to:
$ git config user.name "w3schools-test"
```

12

Aula 104

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

```
git clone git@github.com:seuusuario/seurepositorio.git => Cloning into 'Github'...  (ir ao Github e copy)
git add .                                               => Coloca na área de stage
git commit -m "Mensagem explicativa"                    => Salva a nova versão
git push                                                => Envia para o GitHub
```



13

```
$ code .      => Abre o VScode a partir da bash do Git
$ Git log     => Show all commits
$ clear       => Limpar o ecrã
$ git status  => mostra status
```

Aula 105

Verificando o histórico de versões

```
git log
```

Listagem resumida:

```
git log --oneline
```

```
C:\Projects\001-Udemy\002-C#CursoCompletoPOO\Github>git log --oneline
daa8dae (HEAD -> main, origin/main, origin/HEAD) Foi criada a pagina Blog
2e64728 Acrescentado valor das vendas vendas.html
f0ad5db added pages index e vendas
```

14

Aula 106

Git status, git add e stage

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    blogue.html
        modified:   index.html
        modified:   vendas.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        about.html

no changes added to commit (use "git add" and/or "git commit -a")
```

```
telmo@DESKTOP-K2SKTVO MINGW64 /c/Projects/001-Udemy/002-C#Cur
$ git log --oneline
f586e5d (HEAD -> main) delete blog, update index
b76eb7f added file about.html
daa8dae (origin/main, origin/HEAD) Foi criada a pagina Blog
2e64728 Acrescentado valor das vendas vendas.html
f0ad5db added pages index e vendas
```

$ git add .
=> adiciona todas as
alterações à área de Stage

$ git reset => to unstage the changes after reset

$ git log --oneline => log em formato simples

$ git commit -m "added file about.html"

```
telmo@DESKTOP-K2SKTVO MINGW64 /c/Projects/001-Udemy/002-C#CursoCompletoPOO/Github (main)
$ git add about.html

telmo@DESKTOP-K2SKTVO MINGW64 /c/Projects/001-Udemy/002-C#CursoCompletoPOO/Github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   about.html

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    blogue.html
        modified:   index.html
        modified:   vendas.html
```

```
telmo@DESKTOP-K2SKTVO MINGW64 /c/Projects/001-Udemy/002-C#CursoCompletoPOO/Github (main)
$ git commit -m "added file about.html"
[main b76eb7f] added file about.html
 1 file changed, 5 insertions(+)
 create mode 100644 about.html
```


Aula 107

Git diff

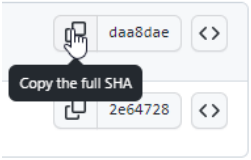
• Comando que mostra a diferença entre arquivos modificados:

$ git diff vendas.html

```
telmo@DESKTOP-K2SKTVO MINGW64 /c/Projects/001-Udemy/002-C#Cur
$ git diff vendas.html
diff --git a/vendas.html b/vendas.html
index e6d73d1..0785b6e 100644
--- a/vendas.html
+++ b/vendas.html
@@ -8,8 +8,10 @@
     </head>
     <body>
         <h1>Página de Vendas</h1>
-        <p>Vendas = 5.000,00 € </p>
         <a href="index.html">Ir para a página de vendas</a>
+        <h2>Relatório de vendas</h2>
+        <p>Venda 1</p>
+        <p>venda 2</p>
     </body>
     </html>
 </html>
```

• Dica: utilizar o VS Code, que mostra graficamente as diferenças

Aula 108



Git checkout

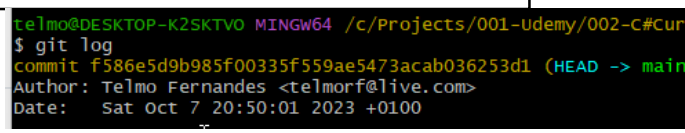
- Permite modificar temporariamente os arquivos do projeto ao estado de um dado commit ou branch

- 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 -- .  => este checkout serve para limpar modificações
```

17



$ git log
$ git checkout <SHA> => "SHA - código que identifica o commit já efetuado"
$ git checkout main => passo para o último commit, do branch main
$ git chekout HEAD~1 => posso também HEAD~N n indica o nr de commits a baixo do atual Head

Aula 109

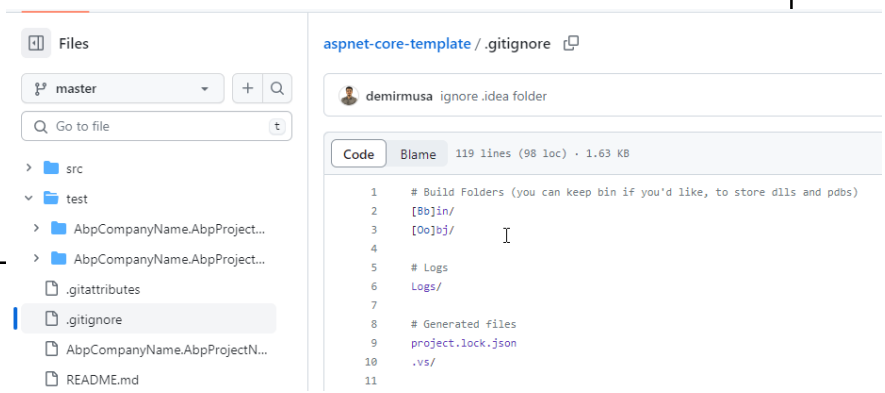
Arquivo .gitignore

- Ficheiro com os arquivos que não devem ser gavados no github
- Ir à net e pesquizar modelos de ficheiros para cada tipo de linguagem de programação

Template
https://github.com/aspnetboilerplate/aspnet-core-template/blob/master/.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.



18

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