• INTRODUÇÃO:

- Iremos iniciar um novo projeto de construção de software e para isso resolvemos utilizar o sistema **Git** para mantermos várias versões do projeto e trabalhando em equipe a partir da plataforma **GitHub**.

1) CRIANDO AS PASTAS (LOCAL & REMOTA):

- Inicialmente criaremos 2 pastas que irão se comunicar na realização do compartilhamento remoto:
- **Pasta remota:** No GitHub criaremos o repositório chamado **SiteEtecAssisRepo**
- **Pasta local:** No Computador criaremos um diretório chamado **SiteEtecAssisDir**

Exemplo:

O repositório remoto (nuvem) terá que ter um nome padrão para acesso de todos que participarão do projeto, já o diretório local poderia ter qualquer nome.

2) VERIFICANDO AS CREDENCIAIS NO COMPUTADOR:

- Para não gerar conflitos, verifique se em seu computador **não existe nenhuma conta do GitHub já conectada**:
  Busque no computador por Credential Manager (Gerenciador de Credenciais) → Windows Credentials → Se houver alguma credencial com o nome GitHub, delete-a.

3) PERMITINDO O USO DO GIT NO DIRETÓRIO LOCAL:

- Abra o diretório do projeto no terminal utilizando o caminho por cd e execute:
  git init

4) INSIRA AS CREDENCIAIS DO DESENVOLVEDOR:

- Com o diretório do projeto aberto execute:
  git config --global user.name "<Seu Nome de Usuário no GitHub>"
  git config --global user.name "<Seu Email Vinculado ao GitHub>"

5) VERIFICAR SE ESTÁ CADASTRADO CORRETAMENTE:

git config --global --list

6) ADICIONAR A URL QUE LIGARÁ O DIRETÓRIO LOCAL AO REPOSITÓRIO REMOTO:

git remote add origin "<Inserir a URL do Projeto>"

7) VERIFICAR SE O REPOSITÓRIO REMOTO ESTÁ CORRETO:

git remote -v

8) VISUALIZAR SE O REPOSITÓRIO REMOTO ESTÁ CORETO:

- Caso a URL verificada acima (Passo 1.7), remova a **URL** e adicione novamente (Passo 1.6):
  git remote remove origin

9) PUXE OS DADOS QUE ESTÃO NO REPOSITÓRIO REMOTO:

git pull origin main

10) ALTERE A BRANCH PARA A MESMA QUE ESTÁ NO REPOSITÓRIO REMOTE:

- No repositório remoto alocado na nuvem a **branch** padrão se chama **main**:

git branch main

git checkout main

11) ABRIR O DIRETÓRIO LOCAL NO VISUAL STUDIO CODE:

- Abra o diretório no Visual Studio Code para iniciarmos as alterações.

12) ABRIR O DIRETÓRIO LOCAL NO VISUAL STUDIO CODE:

- Adicione um arquivo de index.html com o HTML Skeleton.

13) EMPACOTAR AS ALTERAÇÕES REALIZADAS:

git add .

git commit -m “<Mensagem>”

14) ENVIAR AS ALTERAÇÕES PARA A NUVEM:

git push -u origin main

- Loga no GitHub por code

15) VAMOS ADICIONAR BOOTSTRAP NO PROJETO:

- Adicione as 2 linhas de configuração do bootstrap no código (link e script):

https://getbootstrap.com/docs/5.3/getting-started/introduction/

16) EMPACOTAR AS ALTERAÇÕES REALIZADAS E ENVIE PARA A NUVEM:

git add .

git commit -m “<Mensagem>”

git push origin main

17) ADICIONE UM CARROUSEL DO BOOTSTRAP NO PROJETO E ENVIE PARA A NUVEM:

- Adicione as linhas de código de carrousel bootstrap:

git add .

git commit -m “<Mensagem>”

git push origin main

18) ADICIONE BOTÕES AZUIS DO BOOTSTRAP NO PROJETO E ENVIE PARA A NUVEM:

- Adicione 4 botões azuis com Bootstrap na parte superior do site:

git add .
git commit -m "<Mensagem>"
git push origin main

19) CRIE 4 PÁGINAS WEB, UMA PARA CADA BOTÃO:

- Adicione 4 páginas em branco no projeto, pagina1, pagina2, pagina3 e pagina4. Vincule as páginas aos botões e realize os commits:

git add .
git commit -m "<Mensagem>"
git push origin main

20) ALTERE AS CORES DOS BOTÕES PARA VERMELHO:

- Altere as cores dos botões.

git add .
git commit -m "<Mensagem>"
git push origin main

21) REVERTA O ÚLTIMO COMMIT:

- Remova as alterações feitas no commit anterior

git revert HEAD ou git revert <Número HASH>
git add .
git commit -m "<Mensagem>"
git push origin main

22) CRIE UMA NOVA BRANCH:

git branch modificacoes_paralelas
git checkout modificacoes_paralelas

23) BOTÕES VERDES NA NOVA BRANCH:

- Altere as cores dos botões para verde e commit.

git add .
git commit -m "<Mensagem>"
git push origin main

24) VOLTE PARA A BRANCH PRINCIPAL:

git checkout main

25) JUNTE AS ALTERAÇÕES DA BRANCH EM PARALELO:

git merge modificacoes_paralelas

26) ENVIE TUDO PARA A NUVEM:

git push -u origin main

27) PUBLIQUE O SITE NA WEB:

- No GitHub vá em Pages, em Main e Save.

28) SALVE A URL DO REPOSITÓRIO EM UM ARQUIVO TXT:

- Salve um arquivo de texto com a URL na pasta tarefas.