* GIT
  + Software gerenciador de versões
* GITHUB
  + Plataforma que hospeda projetos versionados com GIT

###################### INÍCIO ######################

**GIT**

* DOWNLOAD GIT
  + Como saber se o Windows é 32 ou 64 BITS
    - Até 01:58
      * <https://youtu.be/jG0yFfZn1Y0?t=36>
  + Site para fazer download do GIT
    - <https://git-scm.com/downloads>
    - Standalone Installer
      * Clicar em 32-bit ou 64-bit
  + Como instalar o Git no Windows em 2023
    - Até 03:51
      * <https://youtu.be/d5d7DsQ5uXA?t=78>

###################### FIM ######################

* VERSIONAR UM PROJETO COM GIT
  + Crie uma pasta chamada projeto1 na sua Área de Trabalho
  + Abra a pasta projeto1 no vscode
  + Abra o terminal do vscode
    - **CTRL + ‘ (aspas)**
  + OPCIONAL > Deixar terminal colorido > Git Bash
    - Até 03:30
      * <https://youtu.be/iKR3ZX2w9vg?t=46>
  + Para versionar o projeto, ou seja, controlar as versões do projeto com git
    - **git init (ETAPA 00)**
    - Vai criar uma pasta com o nome .git (pasta oculta)
    - Todos os arquivos da pasta projeto1, podem ser versionados
  + CONFIGURAÇÕES DE USUÁRIO
    - Controlar quem está trabalhando no projeto
      * Todos os usuários do computador
        + - -system
      * Nosso usuário em todos os projetos que usem git
        + - -global
      * Apenas para esse projeto
        + - - local
    - Definir nome
      * **git config - -global user.name usuario01**
    - Definir e-mail
      * **git config - -global user.email** [**usuario01github@gmail.com**](mailto:usuario01github@gmail.com)
    - Listar últimas configurações no terminal
      * **git config - -global -l**
* COMANDOS GIT
  + Saber o status
    - **git status**
  + Adicionar o arquivo na stage (ETAPA 01)
    - **git add arquivo.txt**
  + Adicionar todos os arquivos na stage (ETAPA 01)
    - **git add .**
  + Tirar a foto > fazer o commit (ETAPA 02)
    - **git commit -m ‘mensagem’**
  + Adicionar na stage e dar commit (depois do primeiro git add .)
    - **git commit -am ‘mensagem’**
  + Histórico de commits
    - **git log**
  + Ver arquivo em determinado commit
    - **git checkout código commit**
  + Voltar ao estado atual
    - **git checkout master**

###################### INÍCIO######################

**GITHUB**

* SITE GITHUB
  + <https://github.com>
* CRIAR CONTA NO GITHUB
  + Clique em SIGN UP
  + Digite seu e-mail
  + Clique em CONTINUE
  + Digite sua senha
  + Clique em CONTINUE
  + Digite seu nome de usuário
  + Clique em CONTINUE
  + Gostaria de receber atualizações de produtos e anúncios por e-mail?
  + Digite y para sim ou n para não
  + Clique em CONTINUE
  + Clique em VERIFICAR
  + Fazer o teste
  + Clique em CREATE ACCOUNT
  + Digite o código recebido por e-mail
  + Marcar quantas pessoas trabalharão com você
  + Marcar se é estudante ou professor
  + Clique em CONTINUE
  + Marcar os recursos que está interessada em usar
  + Clique em CONTINUE
  + Clique em CONTINUE FOR FREE
* CONFIGURAÇÕES
  + Clique na foto (parte superior direita)
  + Clique em SETTINGS
  + Pode alterar nome, digitar sua biografia e colocar foto
* CRIAR NOVO REPOSITÓRIO
  + Clique no +
  + Clique em NEW REPOSITORY
  + Digite um nome sem espaços
  + Digite uma descrição
  + Adicione README
  + Clique em CREATE REPOSITORY
* SINCRONIZAR PROJETO
  + Clique em CODE (botão verde)
  + Copie a URL
  + Acesse o vscode
  + Acesse o terminal
  + Vá até a Área de Trabalho
  + **git clone url**
  + ENTER
* COMANDO PUSH > (LOCAL para GITHUB)
  + Abra a pasta do repositório no vscode
    - Crie um arquivo chamado index.html
    - Utilize apenas a estrutura HTML5
  + Abra o terminal do vscode
    - **git config - -global user.name usuario01github**
    - **git config - -global user.email** [**usuario01github@gmail.com**](mailto:usuario01github@gmail.com)
  + Se der erro, usar o chat GPT para colar o comando
    - git config --global --add safe.directory 'C:/Users/gilap/OneDrive/Área de Trabalho/projeto01'
  + Adicione o arquivo no stage
    - **git add .**
    - **git commit -m ‘arquivo index.html criado’**
    - **git push**
* COMANDO PULL > (GITHUB para LOCAL)
  + Crie um novo arquivo no GITHUB
  + Salve as alterações
  + No terminal do vscode
    - **git pull**

###################### FIM ######################

**BRANCHES**

* BRANCH PRINCIPAL > MASTER/MAIN
* CRIAR NOVA BRANCH
  + **git branch nome da branch**
* CRIAR NOVA BRANCH E ACESSÁ-LA
  + **git checkout -b nome da branch**
* VER AS BRANCHES CRIADAS
  + **git branch**
    - asterisco indica a branch que estamos trabalhando
* MUDAR DE BRANCH
  + **git checkout nome da branch**
* VISUALIZAR HISTÓRICO DE COMMITS
  + **git log**
* EXECUTAR MERGE ENTRE BRANCHES
  + **git merge nome da branch**
* CASOS DE CONFLITOS
  + abrir os arquivos apontados nos conflitos
  + fazer os ajustes
  + salvar
  + **git commit -am ‘mensagem’**
* MINI GRÁFICO DA LINHA DE COMMITS DO GIT
  + **git log - -graph**
  + **git log - - graph - -oneline**
* EXCLUIR BRANCH
  + **git branch -d nome da branch**
* TIPOS DE MERGES
  + Fast-forward
  + Recursive strategy >> Ort strategy
    - <https://github.blog/changelog/2022-09-12-merge-commits-now-created-using-the-merge-ort-strategy/>

**ISSUES**

* QUANDO USAR UMA ISSUE?
  + Rastreio de bugs
  + Sugestão de ferramentas
  + Histórico de alterações
  + Documentação dos passos
* CRIAR UM ISSUE
  + Acessar a aba ISSUE
  + Clique em NEW ISSUE
  + Adicione um título
  + Escreva no campo de mensagem
  + Clique em SUBMIT NEW ISSUE
* FAZER COMMIT COM REFERÊNCIA DA ISSUE
  + **git commit -m ‘Adição da página de contatos #2’**
* ENVIAR OUTRA BRANCH
  + **git push origin nome-branch**
* FECHAR ISSUE COM COMMIT
  + **git commit -m 'Alteração da página de contato closes #2'**