

Git & GitHub

Pra que serve?

Git - Software de Controle de Versão (VCS)

GitHub - Plataforma de Rede Social para Programadores

Principais Vantagens do Git

- **Controle de histórico**
 - ◆ Permite “manipular o tempo do código” para uma versão antiga ou mais recente.
 - ◆ Verificar quem fez alguma alteração no código.
- **Trabalho em equipe**
- **Ramificação do projeto**
 - ◆ Equipes diferentes podem trabalhar individualmente em “ramos” do projeto e depois juntar tudo em um Master (Projeto principal).
- **Segurança**
 - ◆ Equipes de ramificações diferentes não tem permissão de interferir com as outras ramificações.
- **Organização**

Características do GitHub

- Repositórios Ilimitados
- Hospedagem de código-fonte
- Características de rede social
- GitHub Pages integrado (Hospedagem Básica de sites)
- Colaboração
- Forks

Instalação das Ferramentas

-  **Git SCM (Source Code Management)**
-  **GitHub Desktop**

Dicionário do Git & GitHub

Commit - Encerra a transação salvando permanentemente todas as alterações (updates, deletes, inserts) realizadas durante a transação.

Push - Envia o versionamento de código local para o repositório remoto.

Pull - “Puxa” o conteúdo do repositório remoto para o repositório local.

Fetch Origin - Verifica se o repositório local está igual ao repositório remoto.

Issue - Pedir por ajuda para resolver um determinado problema.

Pull Request - Encontrar um problema, analisar o código, encontrar uma solução e apresentar para o desenvolvedor.

Branch - Ramo do projeto.

Master - Ramo principal do projeto.

Merge - Passa a branch para o ramo principal.

Guia da Linguagem Markdown

É uma linguagem de marcação, assim como o HTML que pode ser identificada pela extensão .md a qual o GitHub tem total suporte, visto que pode ser aplicada em três áreas do GitHub, sendo elas:

- No arquivo README.md
- Nas Issues
- Nas Pull Requests

Principais características do Markdown

- Simples de ler
- É como uma versão simplificada do HTML
- Facilidade na conversão de markdown para HTML

Configurações do Markdown

Definições de Texto

Título Principal (nível 1)

Título (nível 2)

Título (nível 3)

****Negrito**** ou **Negrito**

Itálico ou *Itálico*

~~~~Texto Tachado~~~~

--- Linha ou **\*\*\* Linha** (colocar linhas abaixo dos títulos)

### Lista numerada

Qualquer Número + .

1. Teste 1
2. Teste 2
3. Teste 3
  - a. Subitem de teste 3 (Precisa de 3 espaços na frente)

## Lista demarcada

Utilizar \* ou - no início da linha.

\* Teste

\* Subitem do item anterior (Precisa de 3 espaços na frente)

Ficaria assim:

- Teste
  - Subitem do item anterior

## Lista de tarefas

- [ ] Tarefa n1 (Incompleta)

- [ ] Tarefa n2 (Incompleta)

- [ ] Tarefa n3 (Incompleta)

- [x] Tarefa n4 (Completa)

## Adicionar Links

[Texto](Link)

[Google](<https://google.com>)

## Tabelas

titulo1 | titulo2 | título 3

---|---|---

test1 | test2 | test3

## Comandos

```
`print("Hello World")`
```

## Código

```
'''
```

```
num = int(input('Digite um valor: '))
```

```
if num % 2 == 0:
```

```
    print(f'O valor {num} é PAR.')
```

```
else:
```

```
    print(f'O valor {num} é ÍMPAR.')
```

```
'''
```

## Emoji



: + nome do emoji

:wolf