

# Doc.manual

## Conceitos Básicos:

### 1. O que é controle de versão:

- Definição de controle de versão.

É como uma linha do tempo que registra as mudanças no código, facilitando a colaboração e o rastreamento de problemas.

- Exemplos de sistemas de controle de versão.

### 2. Por que fazer controle de versão:

- Benefícios do controle de versão.
- Importância na colaboração e rastreamento de alterações.

### 3. O que é um repositório no Git:

- Definição de repositório.
- Como o Git funciona.

### 4. Diferença entre repositórios do Git e GitHub:

- Explicação das diferenças entre Git (sistema) e GitHub (plataforma web).

## Comandos Simples:

### 1. Como criar um repositório usando comandos do Git:

- git init.

### 2. Comandos Básicos:

- Caso de case sensitive.
- Control + L.
- cd ..
- Mkdir.
- Pwd.

### 3. CLI (Command Line Interface):

- Definição e uso.
- Exemplos de comandos.

### 4. GUI (Graphical User Interface):

- Definição e uso.
- Comparação com CLI.

### 5. Diferença entre git init e git branch:

- Descrição das finalidades de cada comando.

### 6. Por que usar branches:

- Importância das branches no desenvolvimento.

## Configurações Iniciais:

### 1. Configuração de email e usuário:

- Configuração local, global e system.

### 2. git init.defaultBranch:

- Significado e uso.

## Comandos Funcionais:

### 1. Git status:

- Verificação do estado do repositório.
- 2. *Git add:*
  - Adição de arquivos ao staging.
- 3. *Git commit -m:*
  - Confirmação de alterações com mensagem.
- 4. *git reset --mixed, hard, soft:*
  - Explicação das opções de reset.
- 5. *git commit --amend -m "mensagem":*
  - Alteração da mensagem de commit.
- 6. *Git remote rm origin:*
  - Remoção de um repositório remoto.
- 7. *Git commit -a -m "mensagem":*
  - Adição e confirmação em um único comando.

#### **Funcionalidades dentro do GitHub (github.com):**

1. *O que é README FILE:*
  - Explicação do arquivo README no GitHub.
2. *PUBLIC VS PRIVATE:*
  - Diferença entre repositórios públicos e privados.
3. *Add .gitignore:*
  - Significado e uso do arquivo .gitignore.
4. *Choose a license:*
  - Escolha de licença para o repositório.
5. *Git fork ou apenas fork:*
  - Como criar um fork de um repositório.

#### **Processo de Conexão com o Servidor:**

1. *git remote add origin master URL:*
  - Conexão com um repositório remoto.
2. *git remote --v:*
  - Listagem de repositórios remotos.
3. *Push and Pull:*
  - Enviando e recebendo alterações de um servidor remoto.

#### **Comandos de Apoio:**

1. *Git help -h:*
  - Obtendo ajuda sobre comandos do Git.
2. *Git help log:*
  - Ajuda específica para o comando "log".
3. *--log-size:*
  - Uso do parâmetro para definir o tamanho do log.
4. *git log --author "inicial":*
  - Filtrando log por autor.
5. *git log --grep "inicial":*
  - Filtrando log por mensagem.
6. *Git remote rename "Nome do repositório" e "o novo nome":*
  - Renomeando um repositório remoto.

#### **Sessão Erros:**

1. *O que significa esse erro? – Erro de merge:*

- Explicação de um erro de merge.
- 2. E esse? – Fatal: This operation must be run in a work tree:
  - Significado do erro “This operation must be run in a work tree.”

### **Teoria da Computação:**

1. Caminho absoluto e relativo:
  - Diferença entre caminhos absolutos e relativos.

### **Empreendedorismo Aplicado no Conteúdo Aprendido:**

1. Necessidade e oportunidade:
  - Identificação de necessidades e oportunidades no contexto tecnológico.
2. Sistemas capitalistas no meio tecnológico com a criação de aplicação ou aplicativos:
  - Exploração de sistemas capitalistas na indústria de tecnologia.
3. Relação interseccional de características para aplicação em projetos:
  - Abordagem das características necessárias para o sucesso de projetos.
4. Como funciona o processo de construção de ideias baseado em alguns critérios:
  - Passos para a concepção de ideias com base em critérios específicos.

### **Parte da Programação:**

1. Semântica e sintaxe:
  - Diferença entre semântica e sintaxe na programação.
2. Backend e Frontend:
  - Distinção entre desenvolvimento backend e frontend.
3. O que é uma extensão:
  - Significado de extensões em programação.
4. Regras de escrita de nomes no HTML:
  - Convenções para nomes de elementos no HTML.
5. O que são plug-ins dentro do VS Code:
  - Explicação de plug-ins no Visual Studio Code.
6. Estrutura básica de um código HTML:
  - Estrutura fundamental de um documento HTML.

### **Comandos Específicos da Sessão Programação:**

1. ul e ol:
  - Uso de listas não ordenadas (ul) e ordenadas (ol) em HTML.
2. Atalhos – ! (criação de estrutura sozinho):
  - Atalho para criar estrutura HTML básica.
3. Shift+Alt+F:
  - Atalho para formatação de código no VS Code.
4. Tabelas:
  - Criação de tabelas em HTML.
5. Nomeação de páginas:
  - Como nomear páginas HTML.
6. Importação de imagens:
  - Inclusão de imagens em um documento HTML.
7. Como posso importar arquivos CSS em um arquivo HTML usando o comando LINK/REL/HREF:

### **Outras Perguntas sobre Programação:**

1. O que são formulários?
  - Explicação de formulários em programação.

## 2. Como podemos criar um formulário?

- Processo de criação de um formulário HTML.