

Git e Deploy

Wagner Sabino

Explicando Git e Deploy

O que é?

Git

- Sistema de controle de versão
- Trabalho em equipe! Não perca suas coisas!
- Histórico (time machine)
- Linus Torvalds em 2005
- Criado para o desenvolvimento do kernel do Linux
- [Wiki para ler mais sobre...](#)
- [Documentação Git](#)



Configurando pela primeira vez

Será feito apenas uma vez por máquina.

Instalação Windows

Aconselho mudarem para Linux

1) Baixar GitBash (Terminal)

<https://git-scm.com/download/win>

1) Na primeira vez, será necessário configurar seu usuário e e-mail

Dentro do terminal do GitBash digite os comandos, um por vez:

```
git config --global user.name "Meu Nick do GitHub"
```

```
git config --global user.email "meu_email@example.com"
```

Instalação Linux

1) Atualizar repositórios e instalar o git

```
sudo apt update --yes && sudo apt install git --yes
```

1) Na primeira vez, será necessário configurar seu usuário e e-mail

Dentro do terminal digite os comandos, um por vez:

```
git config --global user.name "Meu Nick do GitHub"
```

```
git config --global user.email "meu_email@example.com"
```

MINGW64:/d/2500 Git Projects

Michael Gledhill@PA-ES-3XS MINGW64 /d/2500 Git Projects

```
$ git config --global user.name "practicalseries-lab"
```

Michael Gledhill@PA-ES-3XS MINGW64 /d/2500 Git Projects

```
$ git config --global user.email "lab@practicalseries.com"
```

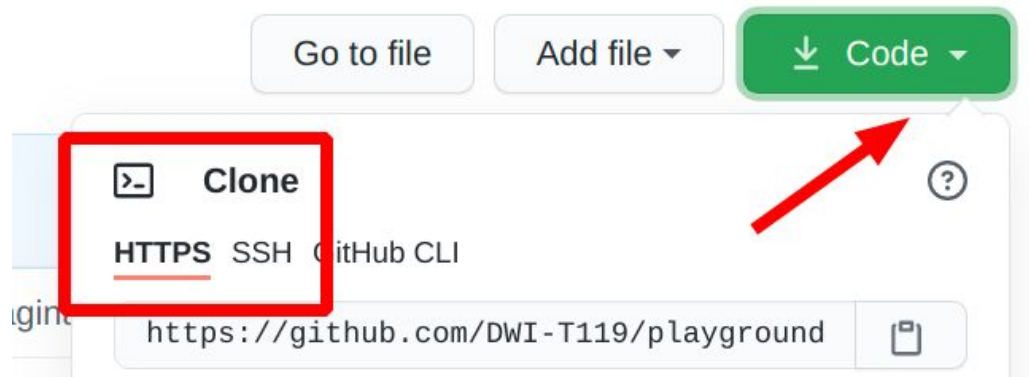
Michael Gledhill@PA-ES-3XS MINGW64 /d/2500 Git Projects

```
$ |
```

Ambos Sistemas (Windows e Linux)

Existem duas formas para transferir dados do seu computador para o repositório online (**e vice-versa**)

- **HTTPS** - Hyper Text Transfer Protocol Secure; **Credencial por usuário e senha**
- **SSH** - Secure Socket Shell; **Necessita configurar a chave SSH**



HTTP

- Se o repositório for público, você conseguirá fazer o clone sem problemas;
- Ao enviar alguma alteração o terminal irá pedir para que você **insira seu usuário e senha** (pode ser requisitado via Browser);
- **Vantagem:**
 - É fácil de utilizar e acessível em qualquer computador
- **Desvantagem:**
 - Incomoda pedindo autorização mais de uma vez em determinados momentos

SSH

- **Vantagem:**
 - Nunca mais incomoda depois de configurado corretamente
- **Desvantagem:**
 - É mais complicado de configurar

Configurando SSH (Windows e Linux)

Parte 1

- **No terminal digite o seguinte comando, para gerar uma chave SSH:**
`ssh-keygen`
- Serão requisitados vários inputs, tecle **enter** sem digitar nada até que termine a configuração (deixe tudo vazio, futuramente veja uma configuração melhor)
- **No terminal digite o seguinte comando, para mostrar o conteúdo da chave:**
`cat ~/.ssh/id_rsa.pub`
- Essa é sua chave pública, **copie ela do terminal**, ela deverá ser inserida no GitHub para que a comunicação seja feita entre seu computador e o serviço
Nunca exponha sua chave privada .

Configurando SSH (Windows e Linux)

Parte 2

- Com a chave copiada, vá até a página principal do GitHub:
 - > Clique no seu avatar no canto superior direito
 - > Settings
 - > SSH and GPG keys
 - > New SSH Key
- Adicione um título qualquer
- Cole a chave onde está “Key”
 - > Add SSH Key
- Pronto! Agora poderá utilizar todos os comandos “sem fazer autenticação”

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Baixando um repositório

Começando a trabalhar em um projeto já existente.
Será feito apenas uma vez por projeto.

Baixando um repositório (Clonando)

- Se irá trabalhar em um repositório (projeto) já existente, será necessário cloná-lo para começar a trabalhar junto com a equipe.
- Será utilizado o comando **git clone** apenas uma vez para o projeto.
- Este comando criará uma pasta com o nome do repositório.

Execute no terminal:

```
git clone git@github.com:<IDENTIFICADOR_DO_PROJETO>
```

- Entre na pasta criada pelo comando e você poderá trabalhar no projeto!

Utilizando
no
dia a dia

Utilizando

- **Tudo se resume aos seguintes passos:**
 - **Trazer** o código atualizado do repositório online (Se atualizar)
 - **Alterar** localmente os arquivos, com coisas novas
 - **Registrar** alterações, dizendo para o Git local quais arquivos
 - **Identificar com uma mensagem** estas novas alterações (**novo pacote**)
 - **Trazer** o código atualizado do repositório online (Podem ter atualizações)
 - **Enviar** o pacote criado para o repositório online



- **Trazer** o código atualizado do repositório online
 - `git pull --rebase`
- **Alterar** localmente os arquivos, com coisas novas
 - Adicionando código novo nos arquivos
- **Registrar** alterações, dizendo para o Git local quais arquivos
 - `git add .` **OU** `git add <nome do arquivo>`
- **Identificar com uma mensagem** este novo pacote. Altere a mensagem.
 - `git commit -m "Fechada versão v1.0.1"`
- **Trazer** o código atualizado do repositório online (Podem ter atualizações)
 - `git pull --rebase`
- **Enviar** o pacote criado para o repositório online
 - `git push`

Utilizando

- `git pull --rebase`
- Alterações em arquivos
- `git add .` **OU** `git add <nome do arquivo>`
- `git commit -m "Fechada versão v1.0.1"`
- `git pull --rebase`
- `git push`



Comandos básicos

Comandos básicos

- **Copiar projeto já existente**
 - `git clone git@github.com:DWI-T119/playground.git`
- **Visualizar status atual local**
 - `git status`
- **Visualizar commits já enviados e se estão remoto**
 - `git log`



Comandos básicos

- **Atualizar repositório com o remoto**
 - `git pull --rebase`
- **Adicionar arquivos para o stage**
 - `git add .`
 - `git add <arquivo>`



Comandos básicos

- **Comitar arquivos localmente**
 - `git commit -m "Mensagem do commit aqui"`
- **Enviar atualizações locais para o remoto**
 - `git push`



Estudando

Treinando Git

- [Git Flow - Artigo](#)
- [Learning Git Branching](#)
- [Configurando Git - Youtube](#)



GitHub

GitHub

- “Rede social”; Plataforma de hospedagem de código-fonte
- Utiliza a tecnologia Git para versionamento
- <https://github.com/>



Deploy

Deploy

- <https://dashboard.heroku.com/apps>
- Integração contínua
- Automação
- Testes
- Pipeline
- AWS (Amazon), Google Cloud, Azure (Microsoft)



Deploy – CI e CDs

Continuous Integration



Continuous Delivery



Continuous Deployment

