

Versionamento – Utilização do GIT e GITHUB para versionamento de software

Objetivo Geral: Instalar e configurar software gratuito para praticar o monitoramento e o gerenciamento de alterações no código.

Conteúdo:

1. Utilizar ferramentas de versionamento em projeto de TI.
2. Reconhecer o sequencial de versões de códigos.

Metodologia e Estratégia:

1. Aula expositiva dialogada com apoio de tutorial;
2. Exercícios de aplicação.

Tutorial Versionamento com GIT e GITHUB

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



Instale a ferramenta GIT.

Faça seu cadastro no GitHub: <https://github.com>

O sistema Git possibilita:

- o registro de todo o histórico de alterações realizadas no código de um projeto e de quem realizou a mudança;
- o retorno para versões anteriores e acesso ao código como estava antes de ser alterado;
- o trabalho de vários programadores simultaneamente, no mesmo arquivo;
- a restauração de um código removido ou modificado.

GitHub

Desenvolvedor do
software



GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo. [Wikipédia](#)

Configurações Iniciais

A primeira configuração é identificar o nome de usuário e um endereço de e-mail. Assim, todas as atualizações (ou *commits*) feitas poderão ser identificadas.

Essa configuração pode ser global (válida para todo o sistema) ou local (para um projeto específico).

Fazendo a configuração global, não é necessário configurar cada projeto. Se a configuração local precisa ser alterada para um projeto específico, você pode rodar o comando sem a configuração global neste projeto.



- Identificação do Nome do usuário e endereço de e-mail

```
MINGW64:/c/Users/Administrador/Documents/_git/senai-versoes-colabora...
Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracao
$ git config --global user.name "fulano de tal"
Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracao
$ git config --global user.email "fulano@email.com"
Administrador@LAPTOP-OT70BICR MINGW64 ~/Documents/_git/senai-versoes-colaboracao
$ |
```

TERMOS E COMANDOS

Agora vamos abordar outros termos típicos da área para que você comece a se familiarizar.

Prompt

É a janelinha na qual digitamos os comandos, também chamada de terminal.

Repositório local

Pasta ou diretório na máquina do programador, na qual são gravadas todas as versões do código.

Repositório remoto

Pasta ou diretório na nuvem, na qual é gravado o projeto completo, com todas as versões do código.

Staging

Local intermediário entre a máquina do desenvolvedor e o repositório remoto, para revisão de alterações antes de salvar no histórico.

Commit

Ato de registrar ou salvar as alterações.

Ramo principal

Sequência principal do código-fonte.

Branch

É uma ramificação do ramo principal, para testes em geral.

README

Arquivo que é uma mescla de cartão de visitas e documentação do projeto.

.gitignore

Arquivo de texto que guarda os arquivos ou pastas que devem ser ignorados, ou seja, os arquivos nomeados no .gitignore não serão monitorados pelo Git.

Acesse uma pasta de referência em seu computador

Inicie o Git Bash – Clicar com o botão direito do mouse e –
Selecionar “Git Bash Here”

Comando Iniciais – \$”código digitado no Prompt”

\$git init – inicia o Repositório

\$git branch –M main

Criar um arquivo de texto dentro do Repositório

Rastrear alterações com o - \$git Status

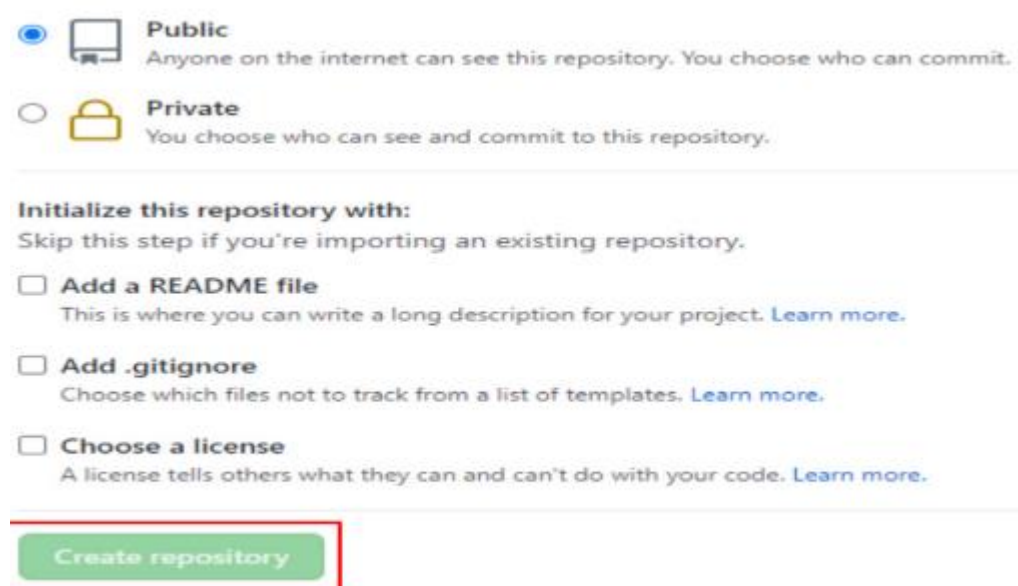
Adicionar versões no repositório – \$git add .


Salvar alterações – \$git commit –m “Mensagem”


Visualizar alterações - \$git log

Fazendo a ligação com o Repositório Online – GITHUB

“Create repository”



☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

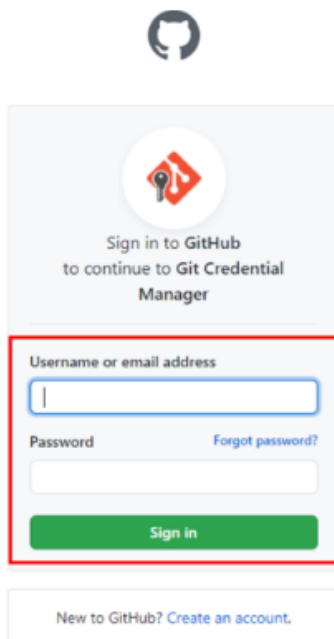
Create repository

Comando a serem digitados no GIT Bash

...or create a new repository on the command line

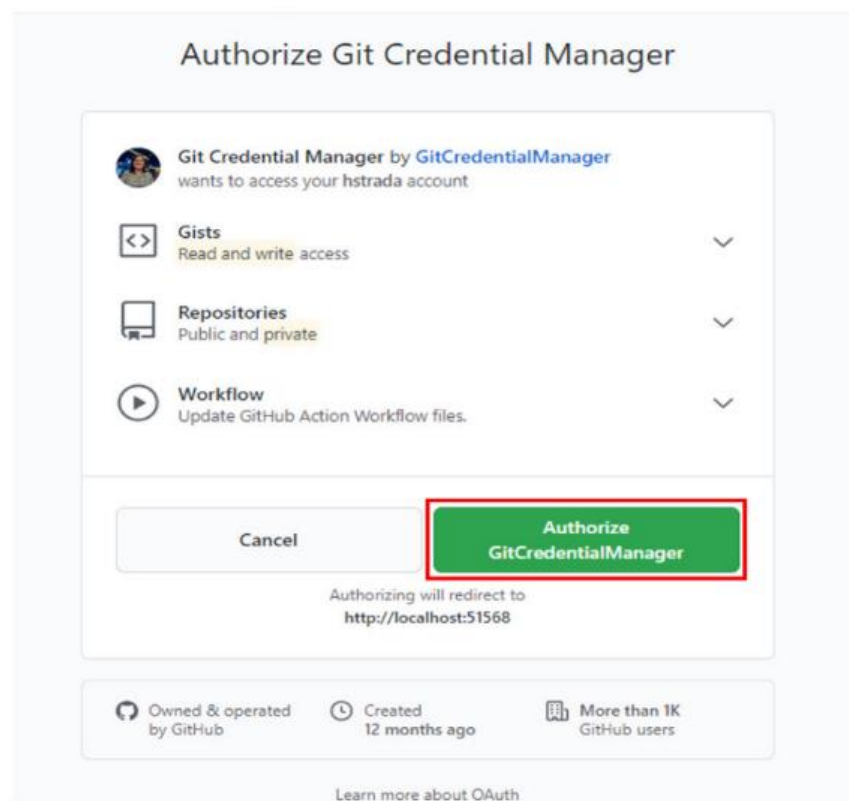
```
echo "# teste1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Diogotb/teste1.git
git push -u origin main
```

Esses comandos serão utilizados em seu computador, na pasta que deseja subir para o seu repositório após digitar esses comando será necessário fazer a conexão da máquina com o repositório

The image shows the GitHub login page. At the top is the GitHub logo. Below it is a circular icon with a red key. The text reads "Sign in to GitHub to continue to Git Credential Manager". Below this is a red-bordered box containing the login form. The form has two input fields: "Username or email address" and "Password". There is a "Forgot password?" link next to the password field. Below the fields is a green "Sign in" button. At the bottom of the page, there is a link for "New to GitHub? Create an account."

- Digite seus dados e clique em "**sign in**".

- Clique em "**Authorize GitCredentialManager**" para autorizar o seu usuári



- A autenticação será feita com sucesso. Feche o navegador e retorne ao terminal local para visualizar.