

Unidade 4 | Seção 1

Roteiro Aula Prática



Sistemas para a Internet

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: **Sistemas para a Internet**

Unidade 4

Seção 4.1

OBJETIVOS

Definição dos objetivos da aula prática:

Aprender como instalar, configurar, criar um projeto, adicionar, salvar (realizar o commit), verificar o status, verificar log e restaurar as alterações de arquivos no sistema de versionamento Git.

INFRAESTRUTURA

Instalações:

1. Instalar o Git;
2. Será necessário realizar cadastro de uma conta(e-mail)

Materiais de consumo:

| Descrição | Quantid. de materiais por procedimento/atividade |
|------------|--|
| Computador | 1 por aluno |

Software: Git

Sim (X) Não ()

Em caso afirmativo, qual?

Pago () Não Pago (X)

Tipo de Licença: NSA.

Descrição do software:

O Git é um sistema de controle de versão distribuído, o que significa que um clone local do projeto é um repositório de controle de versão completo. Esses repositórios locais totalmente funcionais facilitam o trabalho offline ou remotamente.

O Git é um dos sistemas de controle de versão mais utilizados no mundo de desenvolvimento de software. Ele é um projeto de código aberto desenvolvido em 2005 por Linus Torvalds, o criador do kernel do Linux.

Equipamento de Proteção Individual (EPI):

PROCEDIMENTOS PRÁTICOS**Procedimento/Atividade Nº 1****Atividade proposta:**

- Entender o funcionamento do Git;
- Executar o Git, criar um projeto, adicionar, salvar (realizar o commit), verificar o status, verificar log e restaurar as alterações de arquivos no sistema de versionamento Git.

Procedimentos para a realização da atividade:

Git é o sistema de controle de versão open source mais usado no mundo atualmente! Ele é usado para controlar o histórico de alterações de arquivos e principalmente de projetos de desenvolvimento de software. Ele permite mais flexibilidade no fluxo de trabalho, segurança e desempenho.

Um sistema de controle de versão é uma ferramenta que ajuda equipes a gerenciar alterações em código fonte ao longo do tempo. Além disso, ele ajuda os times a trabalharem de forma mais rápida e inteligente. Isso acontece, pois, a ferramenta de controle de versão mantém um registro de todas as versões do código, ou seja, de todas as modificações. Isso ajuda a pessoa que desenvolve a conseguir voltar para qualquer versão anterior ou compará-las, ajudando a descobrir e corrigir erros muito mais rápido.

O Git pode ser usado em todo e qualquer projeto que tenha arquivos de diferentes tipos, podendo ser código, texto, imagens, vídeos, áudios, entre outros. O objetivo principal é permitir o controle de histórico e versão desses projetos, melhorar o trabalho em time e o fluxo de trabalho, proporcionar a segurança dos seus arquivos e outras tantas vantagens faladas nesse post.

Para utilizar o git, você precisará, criar uma conta no gitlab: <https://gitlab.com/users/> e o baixar o Agit, conforme os passos abaixo:

1. Visite o site oficial https://gitlab.com/users/sign_in e realize seu cadastro. Anote o seu usuário e e-mail, irá necessitar dessas informações para configurar o Git;
2. Visite o site oficial <https://git-scm.com/downloads> e realize o download;
3. Para a instalação do Git, clique em next até ao final da instalação;
4. Configurando o Git: após a instalação, clique no botão pesquisar do Windows e digite cmd, selecione a opção prompt de comando, como pode ser observado na Figura 1.

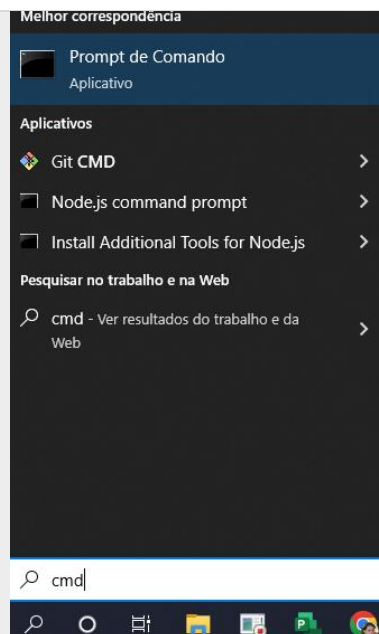


Figura 1 – Acessando o prompt de comando

5. Digite o comando git para verificar se a instalação foi executada de forma correta, caso sim, será disponibilizado a versão do Git, como pode ser observado na figura 2.

```
C:\Users\mauricio>git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone Clone a repository into a new directory
  init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add Add file contents to the index
  mv Move or rename a file, a directory, or a symlink
  restore Restore working tree files
  rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect Use binary search to find the commit that introduced a bug
  diff Show changes between commits, commit and working tree, etc
  grep Print lines matching a pattern
  log Show commit logs
  show Show various types of objects
  status Show the working tree status

grow, mark and tweak your common history
  branch List, create, or delete branches
  commit Record changes to the repository
  merge Join two or more development histories together
  rebase Reapply commits on top of another base tip
  reset Reset current HEAD to the specified state
  switch Switch branches
  tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch Download objects and refs from another repository
  pull Fetch from and integrate with another repository or a local branch
  push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

C:\Users\mauricio>
```

Figura 2 – Verificando a versão do Git.

6. Configurando o Git para que ao realizar alterações no projeto não haja problemas e as alterações sejam salvas. Ainda no prompt de comando digite: git config --global user.name "nome do usuário", necessita ser o que foi cadastrado no Git", como pode ser observado na figura 3.

```
C:\Users\eduardio>git config --global user.email "eduardio@hotmail.com"
```

Figura 3 – Configurando o Git - usuário.

7. Próximo passo é configurar o email, para tanto, basta digitar no prompt de comando a seguinte linha: `git config --global user.email "a conta de e-mail cadastrada no site do git"`, como pode ser observado na figura 4

```
C:\Users\eduardio>git config --global user.email "eduardio@hotmail.com"
C:\Users\eduardio>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=eduardio
user.email=eduardio@hotmail.com
C:\Users\eduardio>
```

Figura 4 – Configurando o Git – conta de e-mail.

Verifique a linha do `user.name` e a linha `user.email` para verificar se foi configurado com sucesso. Após esses procedimentos, podemos concluir que o Git já se encontra instalado e configurado.

Checklist:

Ao final da atividade o aluno deverá saber instalar e configurar o Git. Além de aprender a criar e manipular um determinado repositório(local), utilizando os comandos mais usuais.

RESULTADOS

Resultados da aula prática:

- 1- Instalar o Git;
- 2- Configurar o Git;
- 3- Criar uma pasta no computador do aluno;
- 4- Inserir alguns arquivos, pode ser formato txt, doc, xls e outros (necessário realizar alteração no conteúdo do arquivo, preferencialmente três;
- 5- Executar o comando para inicializar a pasta (repositório);
- 6- Verificar o status dos arquivos que estão na pasta repositório);
- 7- Incluir os arquivos no controle de versão, o Git;

8- Realize uma alteração em um dos arquivos;

9- Mostre o status dos arquivos na pasta, lembrando que temos 1 arquivo com alteração e 2 não alterado;

10- Realize ao menos mais duas alterações e mostre o status dos arquivos na pasta;

11- Inclua os arquivos alterados no repositório do Git (comitar);

12- Verifique o log dos arquivos alterados.