**Colégio Cotemig**

**Concepção de Produtos de Software**

**Prof.** [ALEXANDRE SAUER PAIS LEMES](mailto:alexandresauer@cotemig.com.br)

GIT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Objetivo do exercício:** Esse exercício tem como objetivo praticar s/ controle de versão

**Data de entrega:** Verificar no classroom;

**Descrição:**

1. Crie um novo repositório local com um projeto à sua escolha e realize pelo menos dois commits neste repositório
2. Escolha um repositório disponível no GITHUB .com e faça a clonagem do mesmo, identificando qual foi o autor do último commit realizado no projeto e a(s) linguagem(s) utilizadas.  
   git clone (Clonei o meu repositorio que fiz para o item 1)
3. Identifique a finalidade dos seguintes comandos:

a. git init

Inicializa um novo repositório Git local em um diretório. Isso cria um novo repositório vazio.

b. git config --global user.name "turing"

Define o nome do usuário global para o Git. O nome "turing" será usado para identificar o autor dos commits.

c. git add EXERCICIO.txt

Adiciona o arquivo EXERCICIO.txt ao stage (área de preparação), tornando-o pronto para o próximo commit.

d. git add -A

Adiciona todas as alterações na pasta de trabalho (working tree) ao stage. Isso inclui novos arquivos, arquivos modificados e arquivos removidos.

e. git commit -m "Adicionado nova interface"

Cria um novo commit com a mensagem "Adicionado nova interface". Um commit é uma captura do estado atual do repositório.

f. git commit

Abre um editor de texto para você inserir uma mensagem de commit. Isso é útil quando você deseja fornecer uma mensagem mais detalhada sobre as alterações.

g. git reset --hard HEAD

Desfaz as alterações no diretório de trabalho e no stage até o commit mais recente (HEAD).

h. cd Downloads

Navega para o diretório "Downloads".

i. pwd

Mostra o diretório atual (caminho de trabalho).

j. cd ..

Navega um nível acima no sistema de arquivos.

k. ls

Lista os arquivos e diretórios no diretório atual.

l. git pull

Puxa as alterações do repositório remoto e as mescla com o repositório local.

m. git push

Envia commits locais para o repositório remoto no GitHub.

n. git clone <<url>>

Clona um repositório remoto do GitHub para o seu computador local.

o. git diff

Mostra as diferenças entre o diretório de trabalho e o stage.

p. git show

Exibe informações detalhadas sobre um commit específico.

1. Descreva a função dos seguintes componentes do Git

a. Working Tree, Working Area

É a área onde você trabalha nos arquivos do seu projeto. Os arquivos na working tree podem ser modificados, adicionados ou removidos.

b. Stage Area, Commit

A stage area é onde você prepara as alterações que deseja incluir no próximo commit. O commit é uma captura do estado do repositório no momento em que as alterações são confirmadas.

c. Local Repository

O repositório local é onde o histórico completo do projeto é armazenado no seu computador. Ele inclui todos os commits e ramos criados.

d. Remote Repository

O repositório remoto é uma cópia do repositório em um servidor remoto, como o GitHub. Ele permite a colaboração e a sincronização das alterações entre diferentes computadores.

1. Crie seu próprio repositório no GitHub denominado AlunoQS. Crie uma pasta chamada exercicios e inclua a resolução do exercícios
2. Envie o commit para o repositório remoto. Em seguida, execute as seguintes alterações em cada um dos algoritmos:

* Exercício 01.txt;
* Exercício 02.txt;
* Exercício 03.txt;

1. Adicione um arquivo denominado README.md ao projeto e inclua o seguinte conteúdo

Repositório de exercícios das aulas de Linguagem de Programação