












# Controle de versão e Git

# O que é e para que serve?



## O que é e para que serve?

-  TCC corrigido esse aqui
-  TCC corrigido
-  tcc escrevendo
-  TCC final
-  tcc pdf
-  tcc-2\_Correção-Cristina-21jun
-  TCC-corrigido-esse-aqui 2
-  TCC-corrigido-esse-aqui-2 LEVAR QUAR...
-  TCC-corrigido-esse-aqui-2



# O que é e para que serve?

O Controle (ou gerenciamento) de versão permite a gestão de arquivos, com o intuito de **organizar** as diferentes **versões geradas** naturalmente pelo processo de desenvolvimento de Software.



## O que é e **para que serve?**

Serve principalmente para ajudar os desenvolvedores a **trabalhar em equipe** nos seus projetos.

Dessa forma, fica muito mais fácil manter várias pessoas trabalhando num mesmo projeto, sem que uma afete o trabalho da outra, permitindo uma **maior agilidade** no desenvolvimento com a divisão de tarefas, por “features”.

EM OUTRAS PALAVRAS: ao invés de cada um trabalhar em um arquivo “diferente” salvo na máquina pessoal, com o uso de uma ferramenta de versionamento, **todos trabalharão no “mesmo” arquivo** localmente e sincronizarão seus trabalhos.



# Git

O Git é a principal plataforma de versionamento utilizada atualmente pelo fato de ser disponibilizada gratuitamente (*open source*).

Com ele, os arquivos ficam armazenados num REPOSITÓRIO e suas versões são salvas nele.

Os desenvolvedores acessam os arquivos, trabalham na última versão disponível e após alterar os arquivos, enviam a versão atualizada de volta para o REPOSITÓRIO.



# Git - Instalação

A instalação pode ser feita pelo site <https://git-scm.com/downloads>

Digitem no Google GIT DOWNLOAD. Será o primeiro link (ou não! 🤔 )

## About

## Documentation

## Downloads

[GUI Clients](#)

[Logos](#)

## Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

# Downloads



**Mac OS X**



**Windows**



**Linux/Unix**

Older releases are available and the [Git source repository](#) is on GitHub.



## GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

## Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)





# Git - Principais comandos

- `git --version` - para saber se o GIT está instalado na máquina.
- `git init` - Inicializa um novo repositório localmente na pasta atual.
- `git add` - Adiciona arquivos para commit.
- `git commit` - Notifica o git de que ele deve atualizar (ou commitar) as alterações adicionadas.
- `git branch <novoBranch>` - cria um <novoBranch>. Uma nova ramificação (ou caminho).
- `git checkout <branchFeature>` - troca para <branchFeature>. A partir desta troca, todos os commit's serão armazenados na branch <branchFeature>.
- `git push` - envia as alterações comitadas de volta ao servidor, atualizando as versões de todos os arquivos alterados ou adicionados.
- `git pull` - pega a última versão disponível no servidor.
- `git clone` - clona um repositório do servidor, criando uma cópia dele no computador.

# Soluções de software

[GitCola](#) - um dos primeiros aplicativos para facilitar a gestão de versões em um projeto.

[Github Desktop](#) - A versão que é disponibilizada pela comunidade GitHub.

[GitKraken](#) - Uma versão mais completa e profissional.

**TODOS SÃO FREE! UTILIZE A OPÇÃO QUE MAIS SE ADEQUAR A VOCÊ/SUA EQUIPE.**

