

# Relatório CARD 25 - Prática: Git e Github para Iniciantes

## (III)

Willian Augusto Soder de Souza

O objetivo deste relatório é apresentar os principais conhecimentos adquiridos ao assistir aos dois cursos indicados. O primeiro curso, 'Git e Github para Iniciantes', disponível na Udemy, dá ênfase ao funcionamento do Git via terminal, apresentando diversos comandos e explicando os principais conceitos. Já o segundo curso, 'Como usar o Github Desktop', aborda o uso da interface gráfica do Git. A seguir, apresento um resumo dos principais conceitos e a aplicação prática dos comandos do Git via terminal.

### O que é o Git?

O Git é um sistema de controle de versão distribuído criado por Linus Torvalds em 2005. Ele foi desenvolvido para facilitar o gerenciamento de projetos de software, permitindo que várias pessoas trabalhem em um código simultaneamente sem conflitos. O principal objetivo do Git é rastrear mudanças no código-fonte ao longo do tempo, possibilitando a colaboração eficiente, o controle de versões e a reversão para estados anteriores do projeto quando necessário. Ele garante segurança, flexibilidade e um histórico completo das alterações feitas no projeto.

Abaixo segue a explicação de cada comando básico do git além de sua prática via terminal.

- **Git config:** usado para configurar opções e preferências do Git, tanto em nível global quanto para repositórios específicos. Ele permite definir informações como o nome de usuário, o endereço de e-mail, configurações de editores de texto, entre outros parâmetros importantes.

```
williansoder@pop-os:~$ git --version
git version 2.34.1
williansoder@pop-os:~$ git config --global user.name "Willian Soder"
williansoder@pop-os:~$ git config --global user.email "williansousa5097@gmail.com"
```

- **Repositório:** é o local onde todo o histórico de um projeto é armazenado, funciona como um "banco de dados" que guarda o código e todas as versões anteriores, permitindo que os desenvolvedores colaborem e rastreiem alterações ao longo do tempo. Abaixo segue a criação de um repositório por meio do git.

```
williansoder@pop-os:~$ mkdir git
williansoder@pop-os:~$ cd git
williansoder@pop-os:~/git$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Repositório vazio Git inicializado em /home/williansoder/git/.git/
williansoder@pop-os:~/git$ cd .git
williansoder@pop-os:~/git/.git$ ls
branches  config  description  HEAD  hooks  info  objects  refs
williansoder@pop-os:~/git/.git$
```

É possível editar arquivos diretamente no terminal usando editores como o 'vim' e adicioná-los ao repositório. Abaixo, segue a criação de um arquivo 'Readme.md' e sua inserção no repositório.

```
# Teste de escrita em arquivo
```

```
williansoder@pop-os:~/git$ # adicionando o arquivo
williansoder@pop-os:~/git$ git add Readme.md
williansoder@pop-os:~/git$ git status
O comando 'dit' não foi encontrado, mas existem 22 semelhantes.
williansoder@pop-os:~/git$ git status
No ramo master

No commits yet

Mudanças a serem submetidas:
  (utilize "git rm --cached <arquivo>..." para não apresentar)
    new file:   Readme.md
```

- **Commit:** uma "foto" ou "instantâneo" do estado atual do projeto, representando uma mudança ou um conjunto de mudanças feitas nos arquivos do repositório. Cada commit salva o histórico do projeto naquele momento específico, permitindo que você registre o progresso e rastreie alterações ao longo do tempo.

```
williansoder@pop-os:~/git$ git commit -m "Add Readme.md"
[master (root-commit) ae337af] Add Readme.md
1 file changed, 1 insertion(+)
create mode 100644 Readme.md
```

- **Log:** é usado para exibir o histórico de commits de um repositório, mostrando uma lista dos commits realizados, junto com informações detalhadas.

```
williansoder@pop-os:~/git$ git log
commit ae337af3799f6e077010bca0b56f2bbd57324dbb (HEAD -> master)
Author: Willian Soder <williansousa5097@gmail.com>
Date: Thu Sep 5 15:27:00 2024 -0300

    Add Readme.md
williansoder@pop-os:~/git$ git shortlog
Willian Soder (1):
    Add Readme.md
```

- **Diff:** utilizado para mostrar as diferenças entre versões de arquivos em um repositório Git. Ele compara o conteúdo atual do projeto com uma versão anterior ou entre dois commits específicos, exibindo as mudanças feitas linha por linha. Isso permite que os desenvolvedores

visualizem o que foi adicionado, modificado ou removido antes de fazer um commit ou em relação a commits anteriores.

```
williansoder@pop-os:~/git$ git status
No ramo master
nothing to commit, working tree clean
williansoder@pop-os:~/git$ vim Readme.md
williansoder@pop-os:~/git$ git diff
diff --git a/Readme.md b/Readme.md
index 6fe9c98..84b71e9 100644
--- a/Readme.md
+++ b/Readme.md
@@ -1,2 @@
-# Teste de escrita em arquivo
+# Teste de escrita em arquivo
+teste 2
williansoder@pop-os:~/git$
```

- **Checkout:** usado para navegar entre diferentes branches (ramificações) ou commits em um repositório Git. Ele permite que você altere o estado atual do repositório, seja para trabalhar em uma nova branch, revisar um commit anterior ou até restaurar um arquivo para uma versão anterior.

```
williansoder@pop-os:~/git$ vim Readme.md
williansoder@pop-os:~/git$ git status
No ramo master
Changes not staged for commit:
  (utilize "git add <arquivo>..." para atualizar o que será submetido)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Readme.md

nenhuma modificação adicionada à submissão (utilize "git add" e/ou "git commit -a")
williansoder@pop-os:~/git$ git diff
diff --git a/Readme.md b/Readme.md
index 84b71e9..ad737e8 100644
--- a/Readme.md
+++ b/Readme.md
@@ -1,2, +1,3 @@
 # Teste de escrita em arquivo
 teste 2
+linha errada
williansoder@pop-os:~/git$ # RESETANDO ARQUIVO
williansoder@pop-os:~/git$ git checkout Readme.md
Updated 1 path from the index
williansoder@pop-os:~/git$ git diff
williansoder@pop-os:~/git$
```

Outro comando usado para desfazer alterações em um repositório é o **reset**. Ele permite ajustar o ponto de referência da branch atual, ou seja, mover o ponteiro para um commit anterior. Dependendo da opção usada, as alterações podem ser mantidas ou descartadas. O reset possui 3 modos:

- **Soft:** move o ponteiro da branch atual para o commit especificado, mas mantém as alterações feitas nos arquivos tanto na área de stage (index) quanto na working directory.
- **Mixed:** move o ponteiro da branch atual para o commit especificado, removendo as alterações da área de stage (index), mas mantendo-as na working directory.

- **Hard:** move o ponteiro da branch atual para o commit especificado e descarta todas as alterações na área de stage e na working directory. Ou seja, o estado do repositório volta exatamente ao estado daquele commit.

```
williansoder@pop-os:~$ cd git
williansoder@pop-os:~/git$ vim Readme.md
williansoder@pop-os:~/git$ git status
No ramo master
Changes not staged for commit:
  (utilize "git add <arquivo>..." para atualizar o que será submetido)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Readme.md

nenhuma modificação adicionada à submissão (utilize "git add" e/ou "git commit -a")
williansoder@pop-os:~/git$ git commit -am "commit reset"
[master e919f07] commit reset
 1 file changed, 1 insertion(+)
williansoder@pop-os:~/git$ git log
commit e919f0783c85257e92a7547aa95ee88f8a4da0ea (HEAD -> master)
Author: Willian Soder <williansousa5097@gmail.com>
Date:   Thu Sep 5 22:17:44 2024 -0300

    commit reset

commit dbea4984c540f7a3b31ca04c0507d49182ebcd5e
Author: Willian Soder <williansousa5097@gmail.com>
Date:   Thu Sep 5 15:44:56 2024 -0300

    Edit Readme

commit ae337af3799f6e077010bca0b56f2bbd57324dbb
Author: Willian Soder <williansousa5097@gmail.com>
Date:   Thu Sep 5 15:27:00 2024 -0300

    Add Readme.md
```

```
williansoder@pop-os:~/git$ git reset --hard dbea4984c540f7a3b31ca04c0507d49182ebcd5e
HEAD is now at dbea498 Edit Readme
williansoder@pop-os:~/git$ git log
commit dbea4984c540f7a3b31ca04c0507d49182ebcd5e (HEAD -> master)
Author: Willian Soder <williansousa5097@gmail.com>
Date:   Thu Sep 5 15:44:56 2024 -0300

    Edit Readme

commit ae337af3799f6e077010bca0b56f2bbd57324dbb
Author: Willian Soder <williansousa5097@gmail.com>
Date:   Thu Sep 5 15:27:00 2024 -0300

    Add Readme.md
williansoder@pop-os:~/git$
```

## O que é o GitHub?

O GitHub é uma plataforma de hospedagem de código-fonte que utiliza o Git, um sistema de controle de versão. Ele permite que desenvolvedores colaborem em projetos, gerenciem versões de software, compartilhem código e rastreiem alterações. Além disso, oferece recursos como *issues* (para controle de problemas), *pull requests* (para propostas de mudanças no código), e integração

com ferramentas de automação e CI/CD. O GitHub é amplamente usado para projetos de código aberto e privados.

- **Repositórios remotos:** são versões de um repositório Git que estão hospedadas em servidores externos, geralmente acessíveis via internet ou rede interna. Eles permitem que vários colaboradores trabalhem em um projeto simultaneamente, compartilhando e sincronizando suas alterações. O repositório remoto serve como uma referência centralizada onde as pessoas podem enviar (push) e obter (pull) atualizações.
- **Branch:** é uma linha independente de desenvolvimento que permite que você trabalhe em diferentes versões de um projeto ao mesmo tempo. Cada branch tem seu próprio histórico de commits e permite que você desenvolva novas funcionalidades, corrija bugs ou experimente novas ideias sem afetar a versão principal do projeto.
- **Merge:** é o processo de combinar duas ou mais branches em uma única. Isso permite integrar as alterações feitas em diferentes ramificações, consolidando o trabalho realizado separadamente. O merge geralmente é feito quando você conclui o trabalho em uma branch e deseja trazer essas mudanças para outra branch, como a principal.
- **Rebase:** é uma operação usada para reescrever o histórico de commits, aplicando as alterações de uma branch sobre outra de forma linear. Diferente do merge, que combina o histórico de duas branches criando um commit de merge, o rebase move os commits de uma branch para o topo de outra, reorganizando o histórico para parecer que o trabalho foi feito de forma sequencial.
- **GitHub Desktop:** é uma aplicação gráfica que facilita o uso do Git e a interação com repositórios no GitHub, sem a necessidade de usar o terminal ou linha de comando. Ele foi criado para tornar o Git mais acessível, permitindo que desenvolvedores gerenciem seus projetos de forma visual e intuitiva, ideal para iniciantes e para quem prefere uma interface gráfica.

## CONCLUSÃO:

Compreender e dominar o Git e o GitHub é essencial para quem pretende atuar nas áreas de visão computacional e machine learning. Essas ferramentas não apenas facilitam o controle de versões e a colaboração em projetos, mas também garantem a integridade e o rastreamento do progresso no desenvolvimento de modelos e algoritmos. Em projetos de visão computacional e machine learning, onde os experimentos são constantes e as mudanças no código são frequentes, o uso eficiente do Git permite gerenciar diferentes abordagens, realizar testes de forma segura e manter um histórico claro das evoluções. Além disso, o GitHub oferece um ambiente colaborativo robusto para compartilhar código, acompanhar feedback e integrar novas funcionalidades de forma organizada, potencializando o trabalho em equipe e a qualidade dos projetos. Portanto, dominar essas ferramentas é um diferencial crucial para qualquer profissional que busca sucesso nessas áreas tecnológicas em rápida expansão.