

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

Daniel Casanova

O conteúdo do material aborda profundamente o controle de versão utilizando Git, um sistema que permite o gerenciamento eficiente de diferentes versões de arquivos em um projeto, e o GitHub, uma plataforma de hospedagem de código-fonte que facilita a colaboração e o compartilhamento de projetos.

### 1. Conceitos Fundamentais:

- **Git:** É um sistema de controle de versão distribuído criado por Linus Torvalds, que gerencia o histórico de versões de projetos, possibilitando a reversão a estados anteriores e a colaboração entre desenvolvedores por meio de branches (ramificações).
- **GitHub:** Plataforma web que utiliza o Git para versionamento de código, permitindo o compartilhamento e a revisão de projetos de forma colaborativa.

### 2. Comandos Essenciais:

- **git log:** Exibe o histórico de commits, mostrando informações como o autor, data, e mensagens de cada commit.

```
git log [<options>] [<revision-range>] [--] [<path>...]
```

- **git diff:** Compara as mudanças entre o estado atual dos arquivos e o que está salvo no repositório.

```
git diff [<options>] [<commit>] [--] [<path>...]
```

- **git checkout:** Reverte um arquivo para a última alteração registrada no Git.

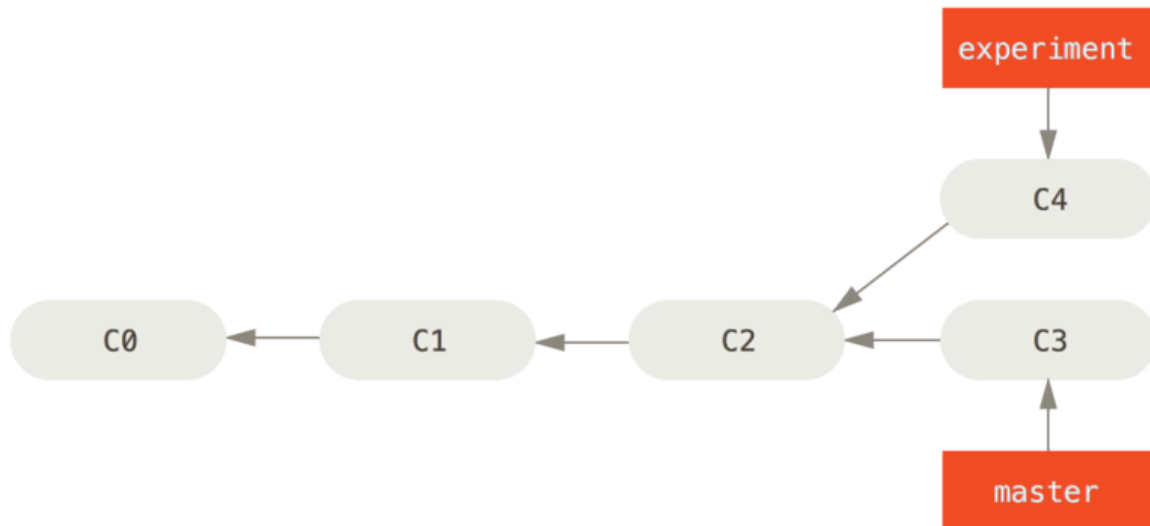
```
git checkout [-q] [-f] [-m] [<branch>]
```

- **git reset:** Desfaz commits, podendo retornar ao estado de commits anteriores com diferentes níveis de impacto (soft, mixed, hard).

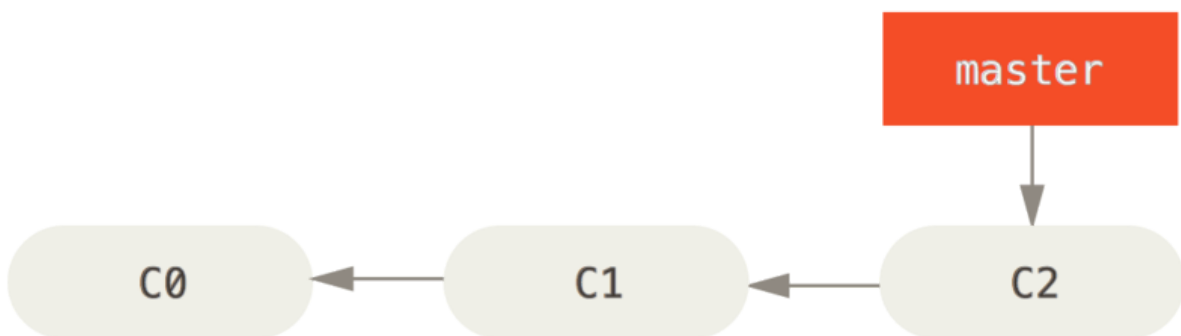
```
git reset [-q] [<tree-ish>] [--] <paths>...
```

### 3. Branching, Merge e Rebase:

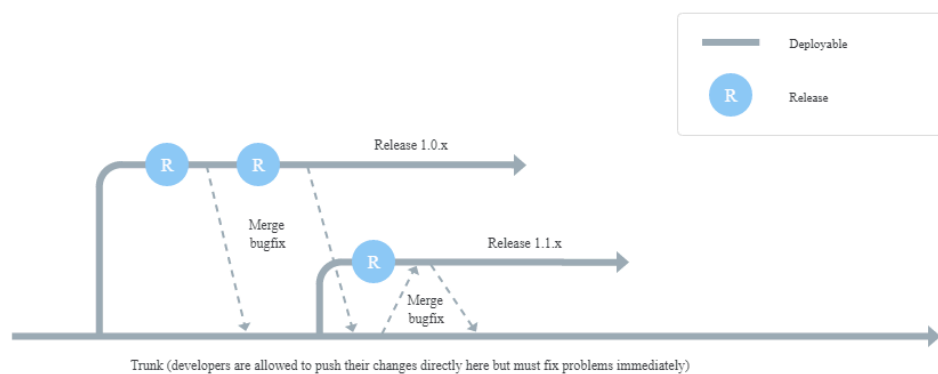
- **Branching:** Permite a criação de linhas de desenvolvimento independentes, o que é útil para trabalhar em novas funcionalidades sem afetar o código principal.



- **Merge:** Combina mudanças de diferentes branches, mantendo o histórico de commits.



- **Rebase:** Reorganiza commits de uma branch, movendo-os para o topo de outra branch, e não mantém o histórico de commits como o merge.



#### 4. Git Desktop:

- Uma interface gráfica para o Git, facilitando o uso dos comandos sem necessidade de interagir diretamente com a linha de comando. Permite realizar todas as operações comuns do Git através de uma interface mais amigável.

### **Conclusões:**

O Git é essencial para o desenvolvimento moderno de software, fornecendo ferramentas robustas para controle de versões e colaboração. O GitHub complementa essa funcionalidade, servindo como um repositório centralizado onde desenvolvedores podem compartilhar, revisar e contribuir em projetos. Comandos como `git log`, `git diff`, e `git reset` são fundamentais para gerenciar mudanças no código, enquanto técnicas de branching, merge, e rebase permitem a organização eficiente do fluxo de trabalho. A introdução de interfaces gráficas como o Git Desktop torna o uso do Git mais acessível, especialmente para iniciantes.

### **Referencias**

<https://desktop.github.com/download/>

<https://stackoverflow.com/questions/73109008/release-branch-process-how-to-merge-release-branches>

<https://www.udemy.com/course/git-e-github-para-iniciantes/>

<https://www.youtube.com/watch?v=Fj3gtbaF8WA>