

Cookbook: Ferramentas de Versionamento

Allyson de Almeida Sirvano

23 de abril de 2025

Introdução

Este documento apresenta um guia prático sobre ferramentas de versionamento, com foco em **Git**, **GitHub** e **comandos de terminal**. O objetivo é destacar os **principais recursos** e **comandos**, fornecendo orientações para uso eficiente no dia a dia.

1 Git

1.1 Configuração Inicial

Antes de começar, configure o Git com o nome de usuário e e-mail:

```
git config —global user.name ‘‘Seu—Nome’’  
git config —global user.email ‘‘Seu—email’’
```

1.2 Principais Comandos do Git

Aqui estão os comandos mais utilizados no Git, organizados por funcionalidade:

1.2.1 Configuração e Inicialização

- **config**: Configura parâmetros globais ou locais do Git.

```
git config —global user.name ‘‘Seu—Nome’’
```

- **init**: Inicializa um novo repositório Git.

```
git init
```

1.2.2 Gerenciamento de Branches

- **branch:** Lista ou cria branches.

```
git branch  
git branch nova-branch
```

- **checkout:** Troca de branch.

```
git checkout nova-branch
```

1.2.3 Trabalho no Repositório

- **status:** Verifica o estado do repositório.

```
git status
```

- **add:** Adiciona arquivos ao *staging*.

```
git add arquivo.txt  
git add .
```

- **commit:** Confirma alterações no histórico.

```
git commit -m 'Mensagem-do-commit'
```

1.2.4 Sincronização com Repositórios Remotos

- **clone:** Clona um repositório remoto.

```
git clone <url-do-repositorio>
```

- **pull:** Baixa alterações do repositório remoto.

```
git pull origin main
```

- **push:** Envia alterações para o repositório remoto.

```
git push origin main
```

1.2.5 Resolução de Conflitos e Histórico

- **merge**: Mescla mudanças de outra branch.

```
git merge outra-branch
```

- **log**: Exibe o histórico de commits.

```
git log
```

- **diff**: Mostra diferenças entre arquivos.

```
git diff
```

2 GitHub

2.1 Principais Recursos do GitHub

O GitHub oferece ferramentas para colaboração e gerenciamento de projetos. Aqui estão os recursos mais importantes:

- **Repositórios**: Armazenamento de código na nuvem. Podem ser públicos ou privados.
- **Pull Requests**: Solicitações para revisar e mesclar alterações no código.
- **Issues**: Rastreamento de tarefas, bugs e melhorias.
- **GitHub Actions**: Automação de fluxos de trabalho como CI/CD.
- **Wiki**: Documentação colaborativa do projeto.
- **Projects**: Gerenciamento de tarefas com quadros Kanban.
- **Releases**: Criação de versões formalizadas do software.
- **GitHub Pages**: Hospedagem de páginas estáticas diretamente de um repositório.
- **Insights**: Métricas e análises sobre atividades no repositório.

2.2 Fluxo de Trabalho no GitHub

Um fluxo de trabalho comum no GitHub envolve as seguintes etapas:

1. Criar ou clonar um repositório.
2. Fazer alterações no código localmente.
3. Usar `git add`, `git commit` e `git push` para enviar alterações ao repositório remoto.
4. Abrir um *Pull Request* para revisão e mesclagem.
5. Gerenciar tarefas e bugs usando *Issues*.

3 Linhas de Comando

3.1 Comandos Essenciais de Terminal

Aqui estão os comandos mais usados para navegação e gerenciamento de arquivos no terminal:

- **Navegar entre Diretórios:**

```
cd <caminho-do-diretorio>
```

- **Listar Arquivos:**

```
ls  
ls -l  
ls -a  
ls -lha
```

- **Criar um Novo Diretório:**

```
mkdir <nome-do-diretorio>
```

- **Ver Conteúdo de Arquivos:**

```
cat <nome-do-arquivo>
```

3.2 Atalhos Úteis

- **Ctrl + C:** Interrompe o comando em execução.
- **Ctrl + L:** Limpa a tela do terminal.
- **Tab:** Autocompleta comandos e nomes de arquivos.