
Versionamento do Software e Identificação de Responsáveis

Breno Campos Vieira - 2211068

Universidade Tecnológica Federal do Paraná - Câmpus Cornélio Procopio

Este documento apresenta a análise do uso de sistemas de versionamento (Git) e sua influência na cultura de desenvolvimento de software. São discutidas boas práticas de colaboração, a adoção do SEMVER e a definição de fluxos de trabalho para a equipe Alpha da empresa Tech Solutions.

1 de setembro de 2025

Sumário

1	Avaliação da escolha do Git	3
2	Medidas adicionais além do controle de versão	3
3	Impacto cultural do versionamento	3
4	Impactos da ausência de controle de versão	3
5	Uso do SEMVER e processos de branch/merge	3
5.1	Fluxo geral de versionamento Git	4
6	Endereço do repositório	4

1 Avaliação da escolha do Git

A decisão de utilizar o Git e um provedor de hospedagem de código (como GitHub, GitLab ou Bitbucket) é adequada para a maioria dos cenários, pois:

- É amplamente utilizado e possui extensa documentação.
- Permite colaboração assíncrona e controle de branches.
- Facilita integração com pipelines de CI/CD.

Alternativas incluem o **SVN**, ainda útil em projetos legados, e ferramentas de controle distribuído específicas (Mercurial, Perforce), que podem ser mais adequadas em cenários corporativos.

2 Medidas adicionais além do controle de versão

Como líder, além do Git, seriam implementadas:

- **Políticas de code review.**
- **Integração Contínua (CI)** para testes automáticos.
- **Documentação atualizada** em README e wiki interna.
- **Política de branches** clara (main, develop, feature, hotfix).
- **Backups automáticos** do repositório remoto.

3 Impacto cultural do versionamento

A adoção do Git influencia diretamente a cultura da equipe:

- Estimula a **colaboração**.
- Promove **responsabilidade individual**.
- Cria uma cultura de **melhoria contínua**.
- Facilita auditorias e promove confiança.

4 Impactos da ausência de controle de versão

A falta de versionamento causou:

- **Perda de eficiência.**
- **Conflitos de código.**
- **Baixa segurança.**
- **Risco de perda de dados.**

5 Uso do SEMVER e processos de branch/merge

O uso de **Semantic Versioning (SEMVER)** auxilia na padronização:

- **MAJOR** – mudanças incompatíveis.
- **MINOR** – novas funcionalidades compatíveis.
- **PATCH** – correções de bugs.

5.1 Fluxo geral de versionamento Git

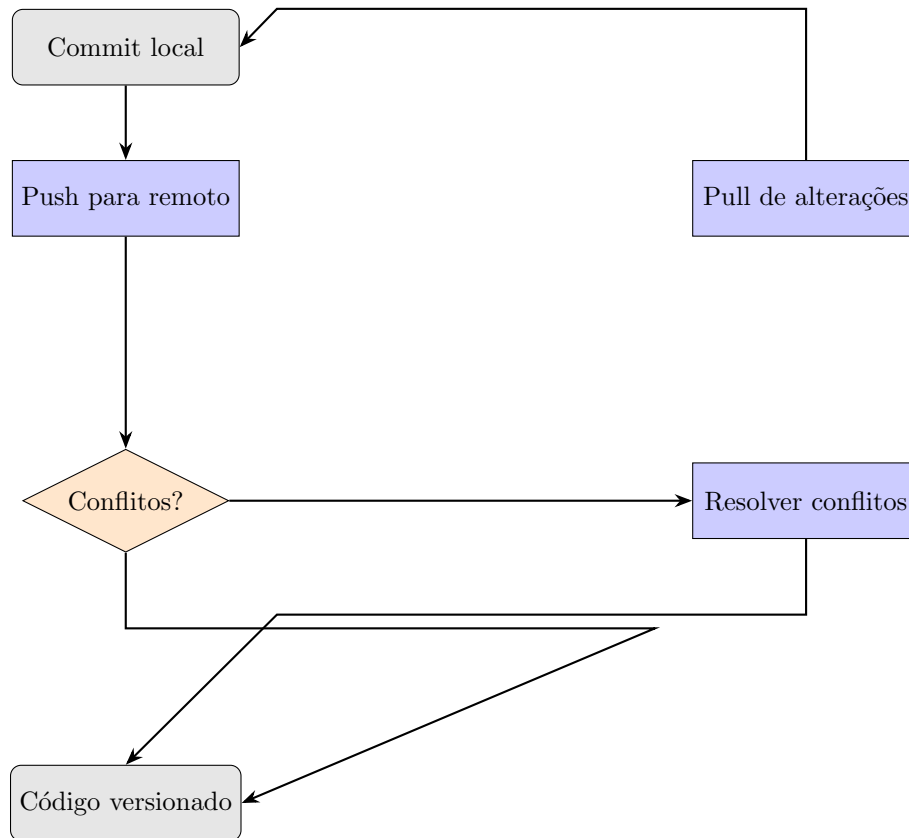


Figura 1: *Fluxo geral de versionamento Git*

6 Endereço do repositório

O projeto encontra-se disponível no seguinte repositório remoto:

`https://github.com/usuario/projeto-exemplo`

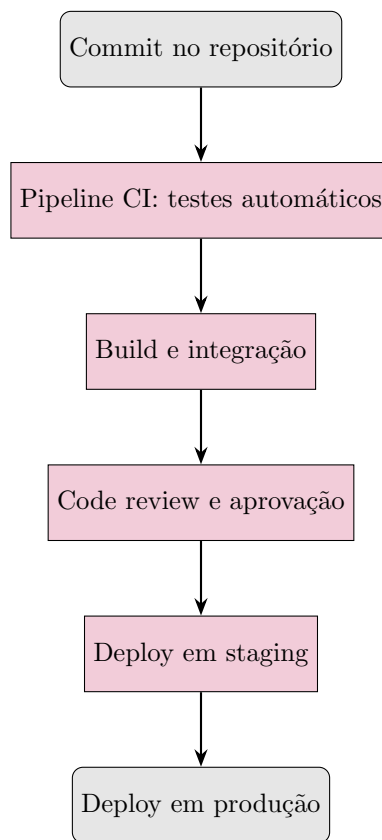


Figura 2: *Pipeline de integração contínua com Git*