

Software Libre

Trabalhos

- técnico - código
 - correção de bugs
 - adição de features
- técnico - fora do código
 - documentação
 - tradução
 - identificação de bugs
- político (ativismo)

Comunidades

- *sempre* observar antes de participar
 - onde está a comunicação?
 - quais os protocolos para contribuir?
- modelos de governança (possivelmente mistos)
 - ditaduras com hierarquia (Linux - kernel)
 - hierarquia empresarial com abertura democrática (SuSE)
 - fundações (LibreOffice)
 - democracia representativa (Debian)
 - adocracia (Parabola Linux)

Recomendações concretas

Ao procurar um projeto:

- Leia as discussões dos desenvolvedores, entenda a atitude
- Veja como `bugs` são reportados, como *pull requests* são feitos
- Veja se há itens "para iniciantes", ou classificados como "fáceis"
- Verifique se existe protocolo ou template padrão
- Nunca se ofenda com crítica técnica ou com a recusa de sua contribuição!
- *Pense em seus valores, e se são compatíveis com os que percebe no projeto*

"Forges" ("forjas")

- sistemas integrados para:
 - controle de versões
 - gerenciamento de bugs e tarefas
 - testes
 - distribuição
 - discussão

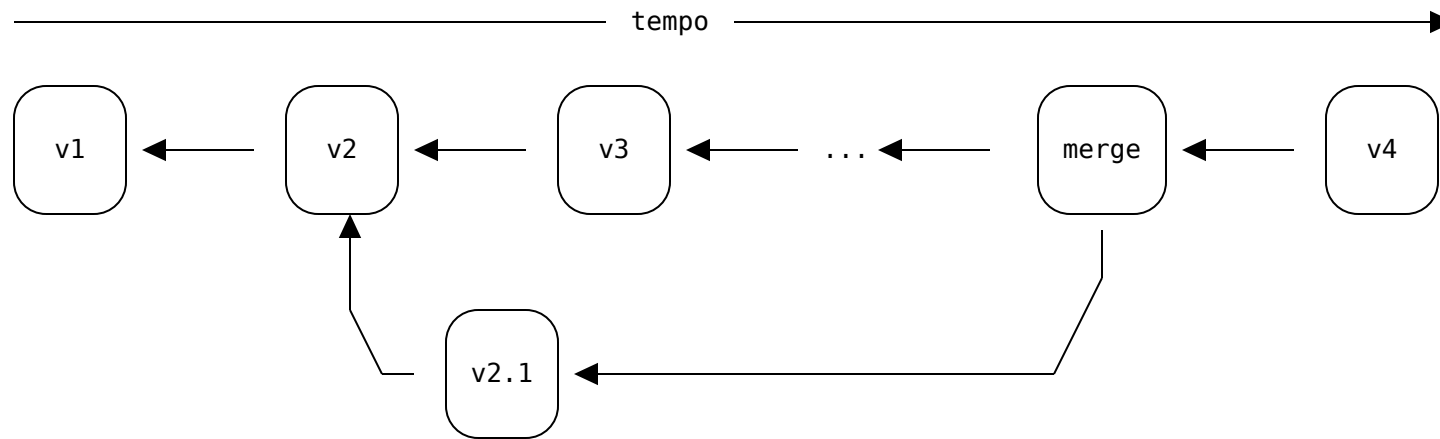
"Forges" ("forjas")

- GitHub (famoso, proprietário)
- GitLab (majoritariamente livre)
- SourceForge (antigo, proprietário)
- Savannah (projeto GNU)
- Codeberg (recente, livre)

Controle de versões

- `git`
 - código distribuído, uma cópia com cada desenvolvedor
 - na prática, usa-se um servidor central

git - fluxo de trabalho



- Setas indicam a estrutura interna
- v2.1 acontece em um branch
- o branch é re-incorporado mais adiante

Tutoriais de git

- <https://learngitbranching.js.org/>
- <https://www.w3schools.com/git>
- <https://www.atlassian.com/git/tutorials>
- <https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>
- Os *outros* slides deste curso