



UNIVERSIDADE
FEDERAL DE
MATO GROSSO DO SUL



Gerência de Configuração de Software

Aula 02 – Terminologia.

Papéis em gerência de configuração. Atividades de Gerência de Configuração (Princípios de gerência de configuração e relação com atividades de desenvolvimento de software)

Prof. Dr. Awdren de Lima Fontão
awdren.fontao@ufms.br



Atividades de Gerência de Configuração

- Identificação de itens de configuração e seus atributos
- Controle de versão e linhas de base ou de referências (baselines)
- Relatórios de status
- Auditoria de Gerência de Configuração

O que é Gerência de Configuração de Software (GCS)?

- É um conjunto de atividades de apoio que lida com as mudanças inerentes ao desenvolvimento de software, mantendo a integridade e a estabilidade durante a evolução do projeto

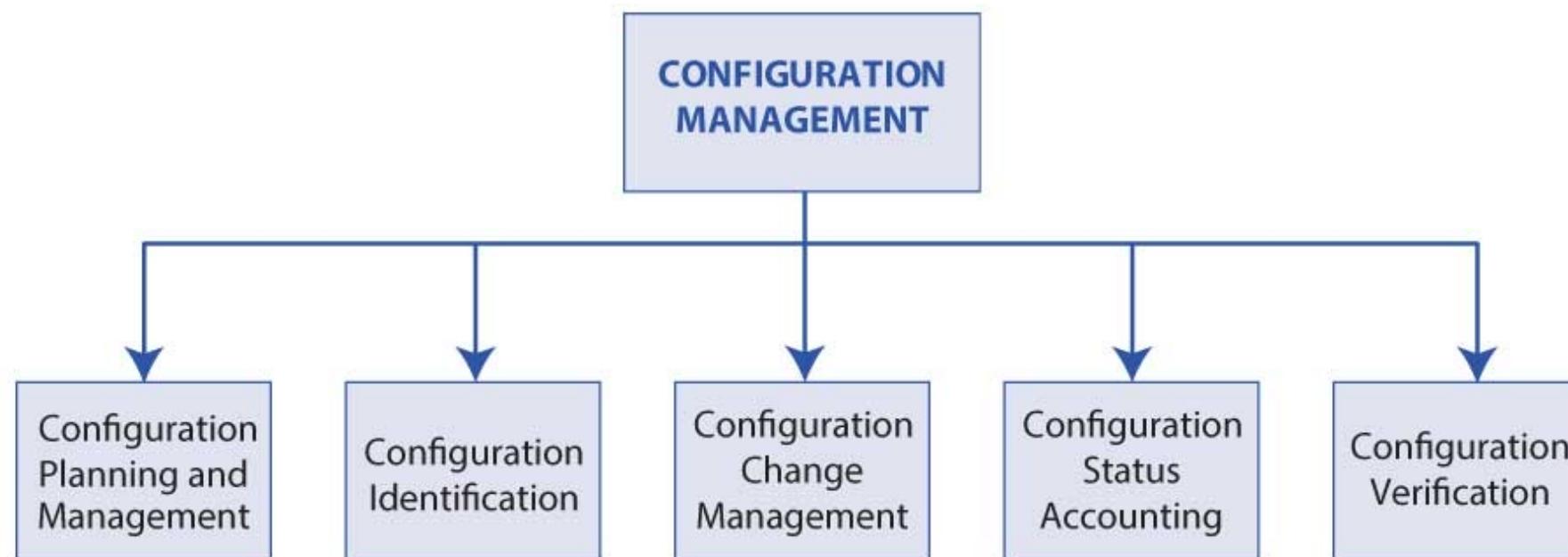


FIGURE 6.5-2 Five Elements of Configuration Management

O que é Gerência de Configuração de Software (GCS)?

- Todo projeto de desenvolvimento de software precisa de Gerência de Configuração de Software!

Resumo

1. GCS é essencial para produzir software de qualidade
2. Existem várias ferramentas open-source disponíveis
3. Requer esforço inicial
 - a) Entender conceitos
 - b) Definir processos
 - c) Integrar e aprender ferramentas
4. Baixo investimento pelo retorno que oferece

E quais são os desafios?

- Escolher as ferramentas certas que combinam com o perfil da sua equipe e do seu projeto
- Não basta só ter as ferramentas corretas, é preciso saber usá-las!
 - equipe acaba subutilizando as ferramentas de maneira burocráticas
 - usando de modo errado e perdendo a informação

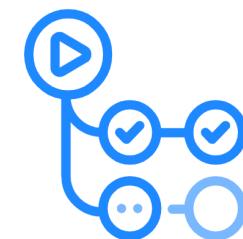
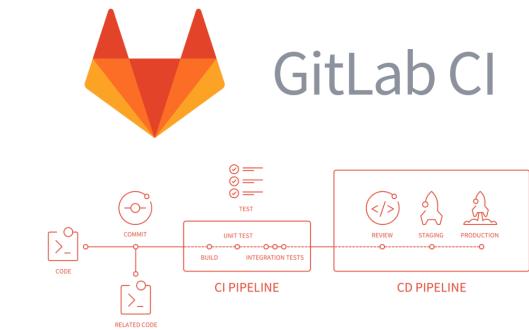
Gerenciamento de versões



Gerenciamento de mudanças



Integração Contínua



GitHub Actions

Principais tarefas de GCS

- Definição e implementação do processo
- Identificação da configuração
- Controle da configuração
- Relatos da situação da configuração
- Avaliação da configuração
- Controle de subcontratados e fornecedores

Processo de desenvolvimento de Software

- Em um processo de desenvolvimento de software, independentemente do paradigma de ciclo de vida adotado
 - inclui as fases:
 - engenharia de sistemas
 - análise de requisitos
 - projeto de software
 - codificação
 - testes e manutenção

Fases	Descrição
Engenharia de Sistemas	Coleta dos requisitos em nível do sistema, com uma pequena quantidade de projeto e análise de alto nível
Análise de Requisitos	Compreensão do domínio da informação através dos requisitos coletados na fase anterior
Projeto de Software	Desenvolvimento de quatro atributos distintos do software: estrutura de dados, arquitetura de software, detalhes procedimentais e caracterização de interfaces
Codificação	Tradução do projeto de software numa forma legível para a máquina
Testes	Realização de testes dos programas. Esses testes concentram-se nos aspectos lógicos internos do software e nos aspectos funcionais externos
Manutenção	Modificações após o software ser liberado. Essas mudanças ocorrem por erros, adaptações, novos ambientes e novas funcionalidades

Itens de Configuração de Software

- Durante o processo de desenvolvimento de um software, são produzidas grandes quantidades de itens de informações
- Alguns desses itens são selecionados de acordo com sua relevância e necessidade de controle tanto de versão quanto de mudança. Esses itens selecionados são chamados **itens de configuração de software** e o conjunto dos mesmos **compõe a configuração de software**

Itens de Configuração de Software

É um **artefato de software** que precisa ser gerido a fim de entregar um produto ou serviço de software.

É essencial para a entrega do serviço que gerenciado como um IC, para manter controle sob todos os registros de configuração, durante o ciclo de vida.

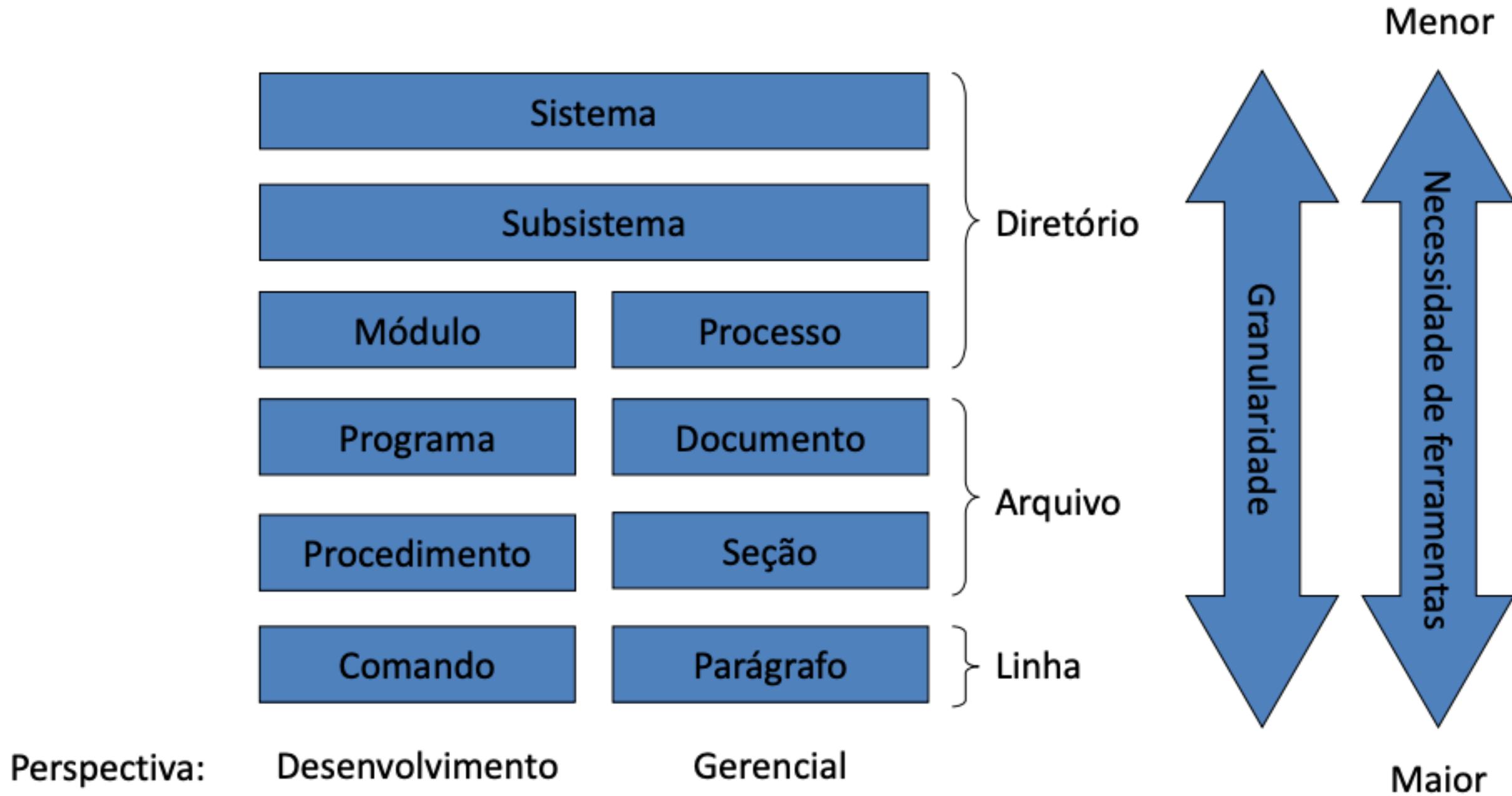
Itens de Configuração de Software

- Produto de Software
- ou
- Produto de desenvolvimento de software escolhido para fazer parte da configuração de software



Exemplos de Itens de Configuração (IC)

1. Especificação do Sistema
2. Plano de Projeto de Software
3. Especificação de Requisitos do Software
4. Manual Preliminar do Usuário
5. Especificação do Projeto
 - a. Descrição do Projeto de Dados
 - b. Descrição do Projeto Arquitetural
 - c. Descrições do Projeto Modular
 - d. Descrições do Projeto de Interface
 - e. Descrições de Objetos (se forem usadas técnicas orientadas a objetos)
6. Listagem do código-fonte
7. Planos, Procedimentos, Casos de Testes e Resultados Registrados
8. Manuais Operacionais e de Instalação
9. Programa Executável e Módulos Interligados
10. Descrição do Banco de Dados
 - a. Esquema e estrutura de arquivo
 - b. Conteúdo inicial
11. Manual do Usuário
12. Documentos de Manutenção
 - a. Relatórios de problemas de software
 - b. Solicitações de manutenção
 - c. Pedidos de mudança
13. Padrões e procedimentos para engenharia de software (templates, artefatos de processos)
14. Ferramentas de produção de software (editores, compiladores, CASE)



Construção *(building)*

Processo de compilação do sistema a partir dos itens fonte para uma configuração alvo;

Utiliza arquivo de comandos que descreve como deve ocorrer a construção;

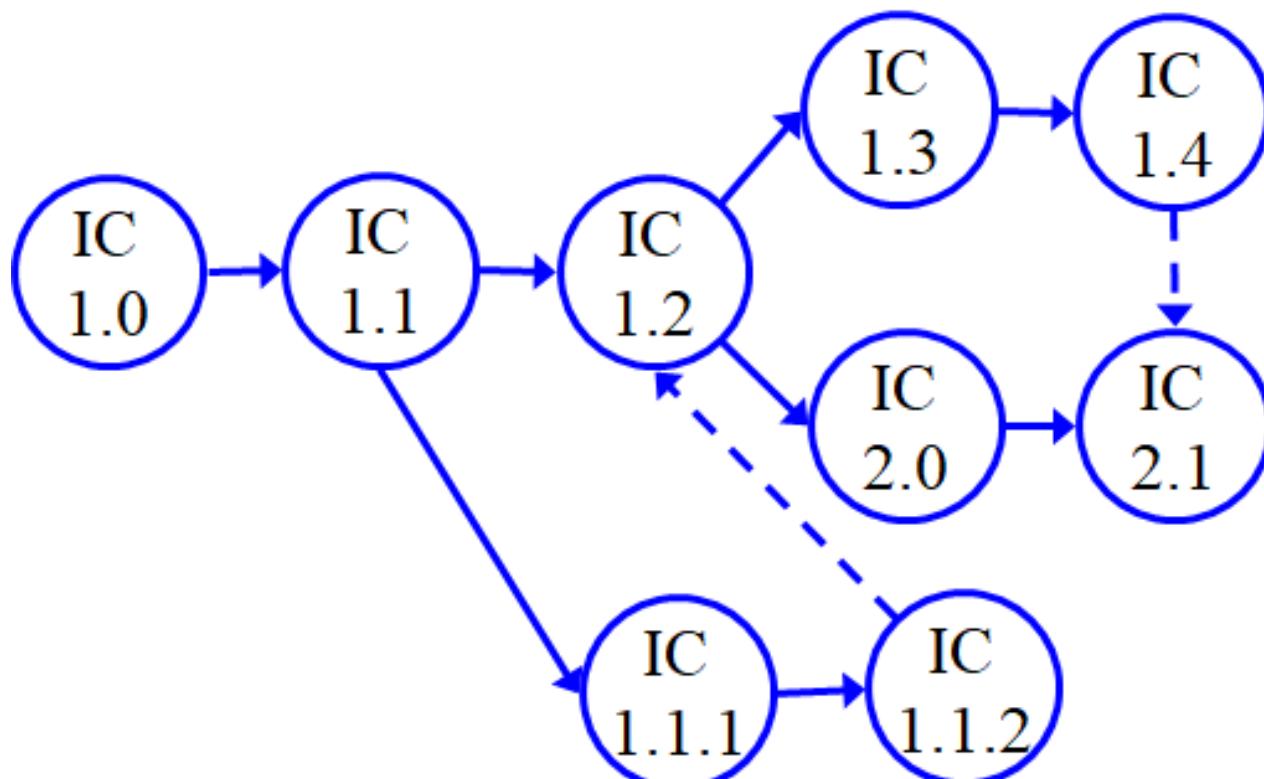
Exemplo: makefile, build.xml, pom.xml;

Os arquivos de comandos também devem ser considerados itens de configuração

Debate:

Para que serve Controle de Versão?

Versão

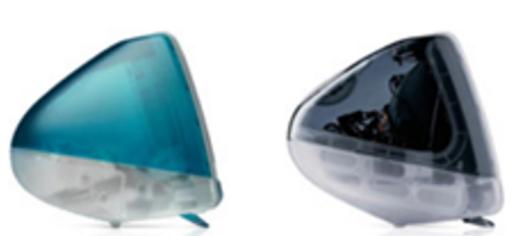


- Instâncias de um mesmo item de configuração que diferem entre si em algo (tipos: **revisões e variantes**)

Versões: Revisões ou Variantes

- **Revisões:** versões criadas para substituir versões anteriores seguindo uma linha temporal (e.g., em resposta a correção ou evolução)
- **Variantes:** versões coexistentes, projetadas para propósitos distintos (e.g., em resposta a diferentes arquiteturas de hardware ou sistemas operacionais)

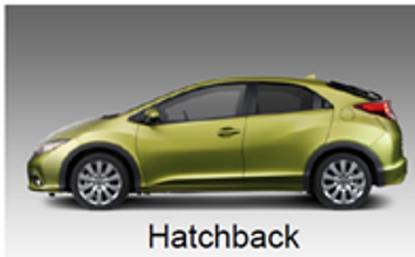
Revisões e Variantes



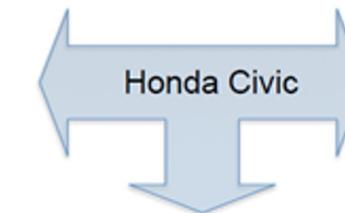
Revisões

Gerações do iMac (1998 – 2013)

Variantes



Hatchback



Sedan



Coupe

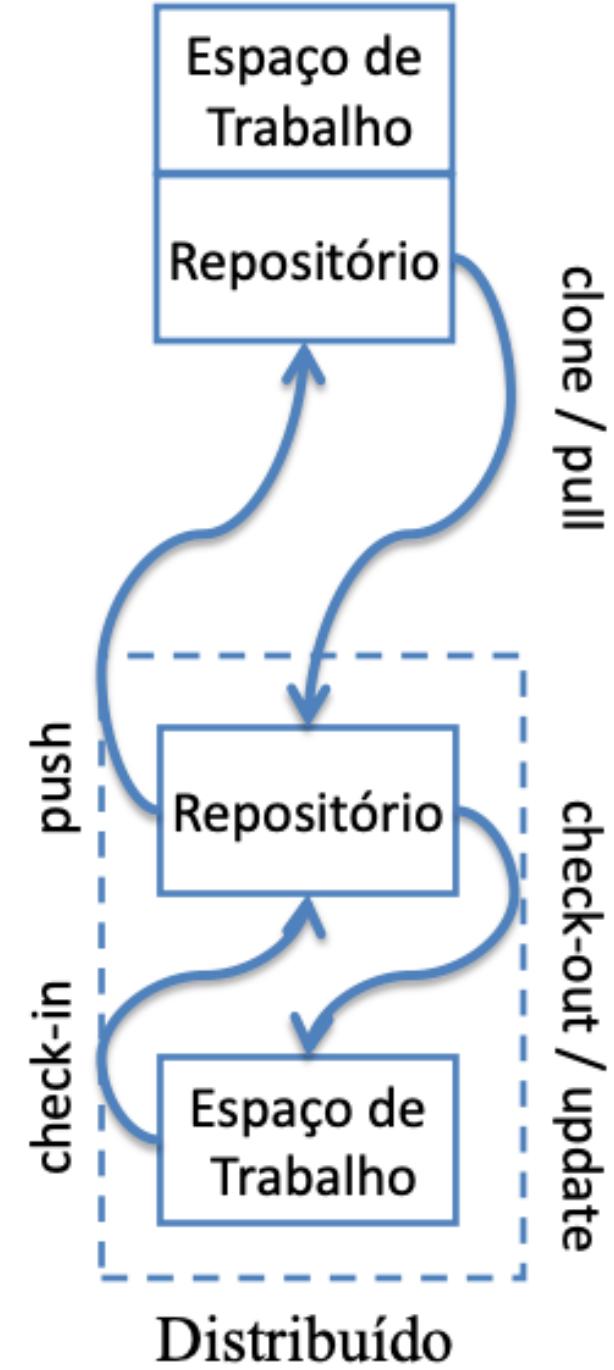
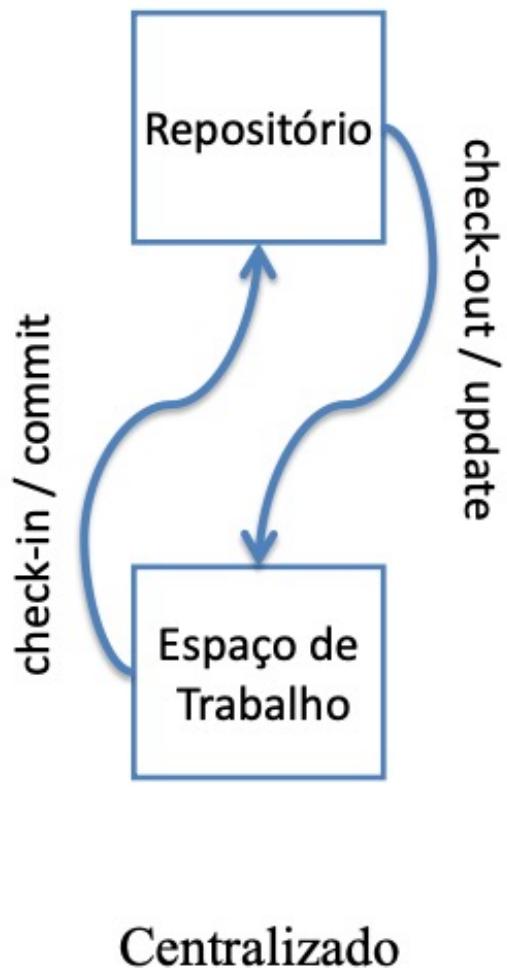
Configuração

- Um conjunto de versão de Itens de Configuração (IC), onde existe somente uma versão selecionada para cada IC do conjunto.
- Uma configuração pode ser vista como a versão de um IC composta de outros IC
- Exemplos:
 - Configuração do Sistema
 - Configuração do processo
 - Configuração do módulo X
 - Configuração dos requisitos do sistema
 - Configuração do código fonte

Repositório

- Local onde os ICs são armazenados
 - Armazena o histórico do projeto;
 - Controle na entrada e saída de ICs
 - Poucos por projeto (normalmente, somente um)
- Utiliza diferentes mecanismos de armazenamento
 - Versionamento completo;
 - Versionamento de diferenças (delta)
- Utiliza diferentes mecanismos de controle de concorrência;
 - Pessimista;
 - Otimista;
 - Misto
- Permite a geração de diferentes relatórios
 - Por item de configuração
 - Por modificação

Topologia de um repositório

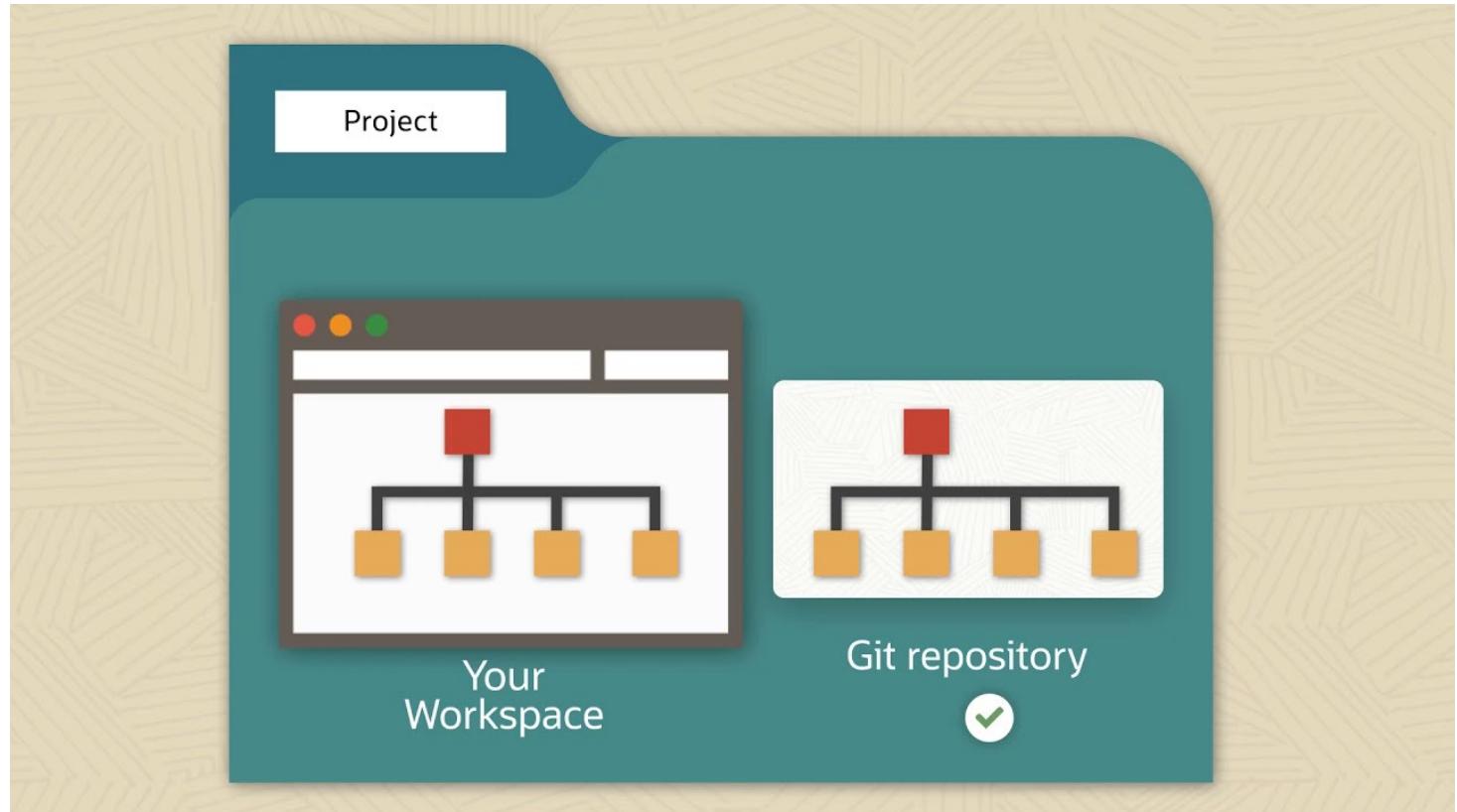


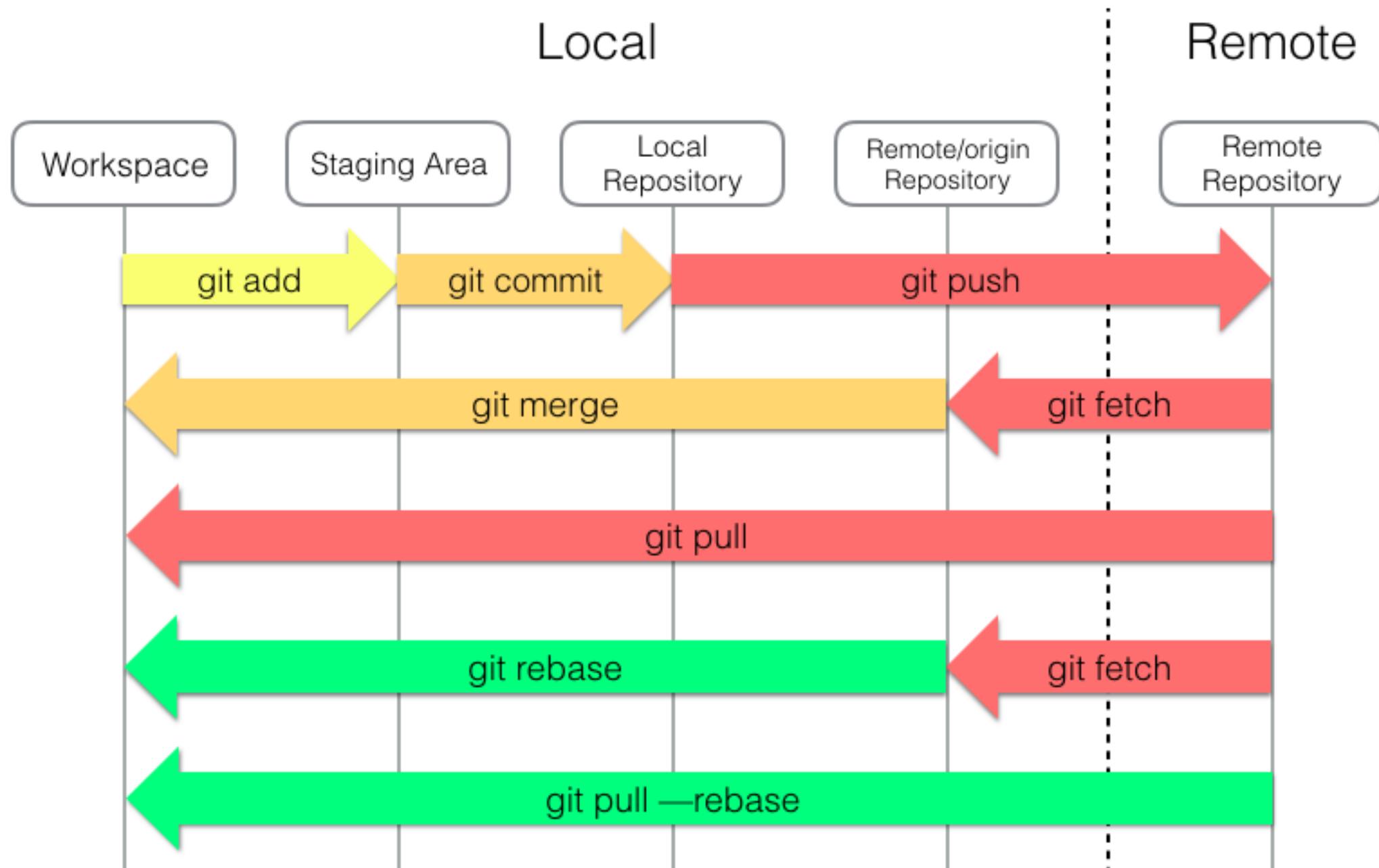
Colaboração

- **Controle de concorrência pessimista**
 - Somente um desenvolvedor pode modificar o IC em um dado momento
 - Custo zero de junção de trabalho
 - Ausência de paralelismo no desenvolvimento
- **Controle de concorrência otimista**
 - Vários desenvolvedores pode modificar um mesmo IC ao mesmo tempo
 - Alto custo de junção de trabalho no caso de ICs complexos (e.g.: IC binários)
 - Permite paralelismo no desenvolvimento
- **Controle de concorrência otimista com notificação**
 - Permite que qualquer desenvolvedor saiba quem está modificando o IC
 - Bom custo/benefício entre controle otimista e pessimista

Espaço de Trabalho

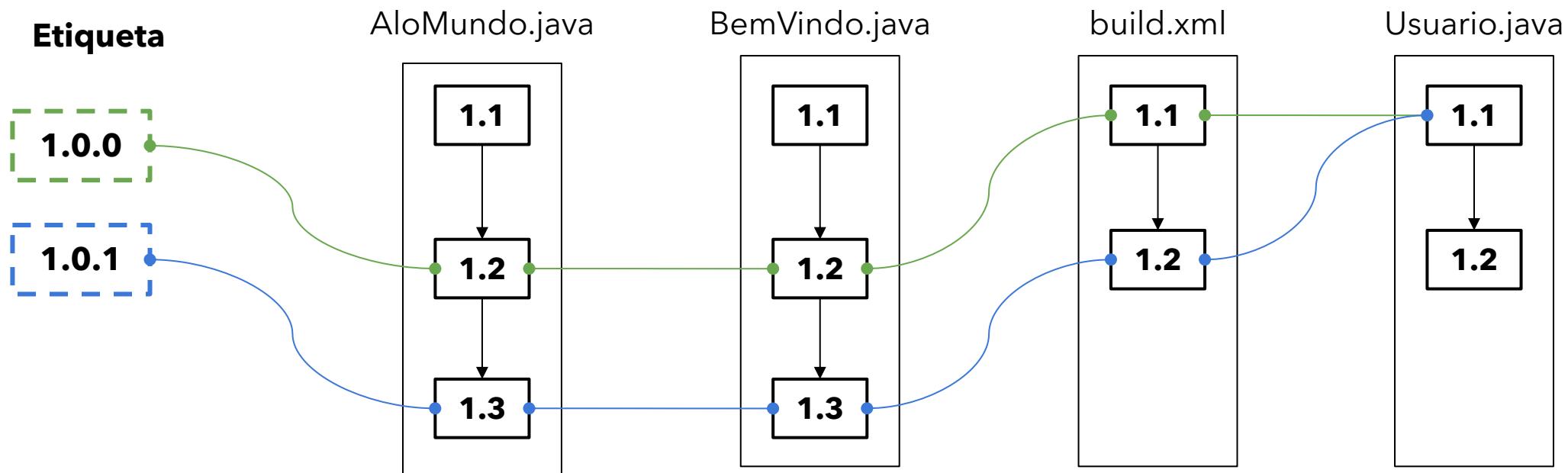
- Local onde o usuário pode fazer seu trabalho de forma isolada
 - Armazena um momento específico do projeto
 - Controle sobre quando sincronizar com o repositório
 - Muitos por projeto (normalmente, um ou mais por desenvolvedor)
 - Sinônimo: caixa de areia (sandbox)



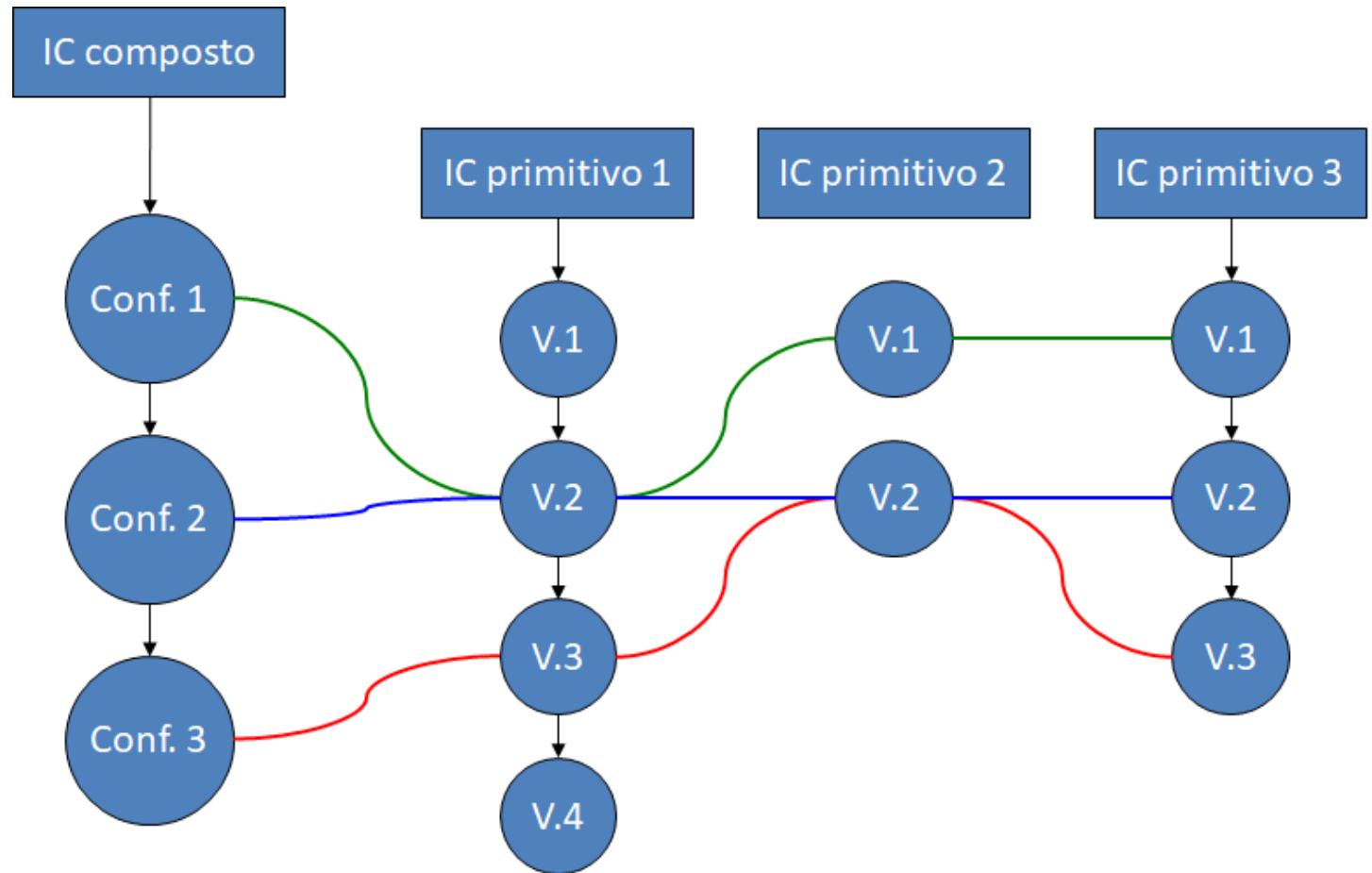


Etiquetas (tag)

- Mecanismo usado para identificar uma configuração
 - As diversas versões de ICs marcadas com um rótulo constituem uma configuração do sistema
- Permite identificar níveis de qualidade dos ICs
- Sinônimo: rótulo (label)

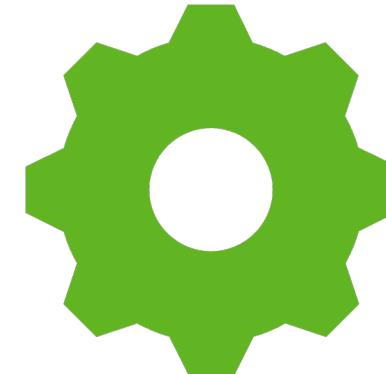


Configuração x Versão



Baseline

- Configuração revisada e aprovada que serve como base para uma próxima etapa de desenvolvimento e que somente pode ser modificada via processo formal de GCS
- São estabelecidas ao final de cada fase de desenvolvimento: análise (*functional*), projeto (*allocated*) e implementação (*product*)
- Momento de criar: balanceamento entre controle e burocracia



Teremos aula específica para baselines.
Incluindo os níveis de controle

Liberação (*release*)

- Substantivo: Versão disponibilizada para um propósito específico
- Verbo: Notificação formal e distribuição de uma versão aprovada
- Importante
 - Toda liberação é uma versão
 - Nem toda versão é uma liberação
- Em alguns casos liberações podem ser desenvolvidas em paralelo (time to market)
- Exemplos
 - Liberação para testes de sistema
 - Liberação para homologação
 - Liberação para entrega ao cliente



Rótulos

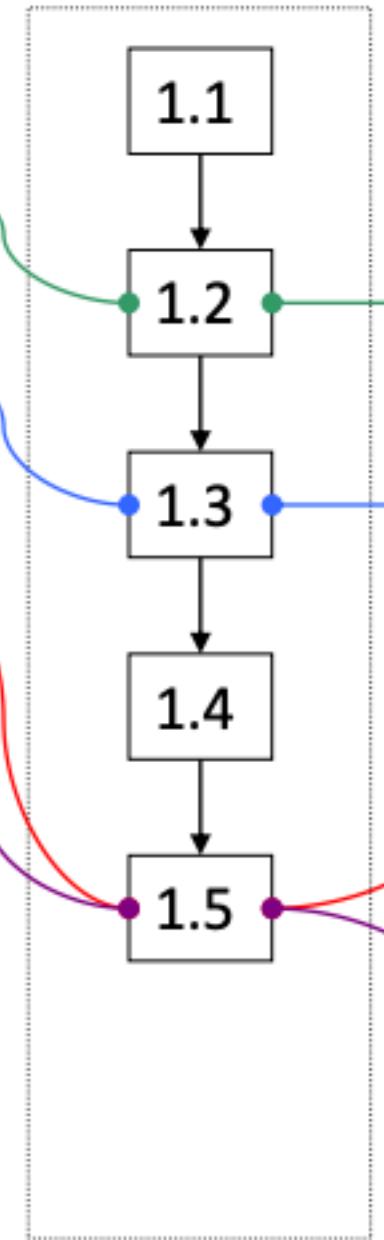
rel_1-0-0

rel_1-0-1

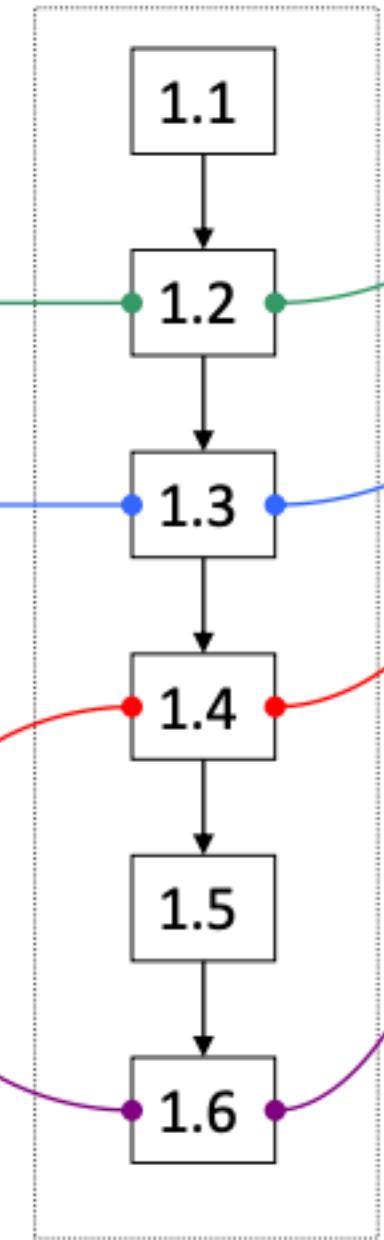
rel_1-1-0

HEAD

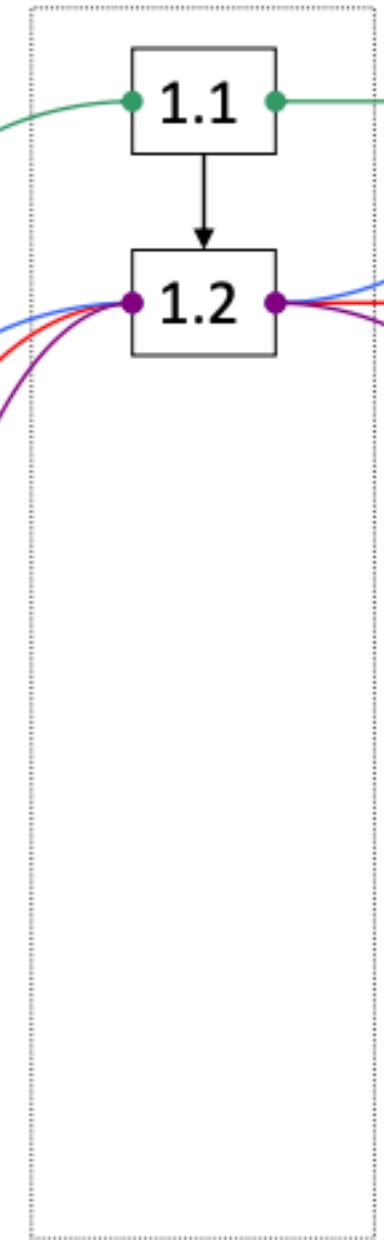
AloMundo.java



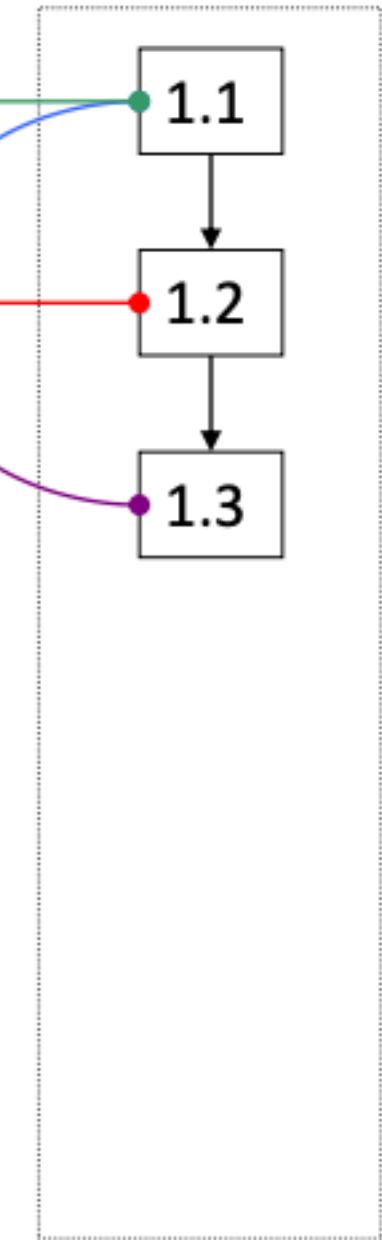
BemVindo.java



build.xml



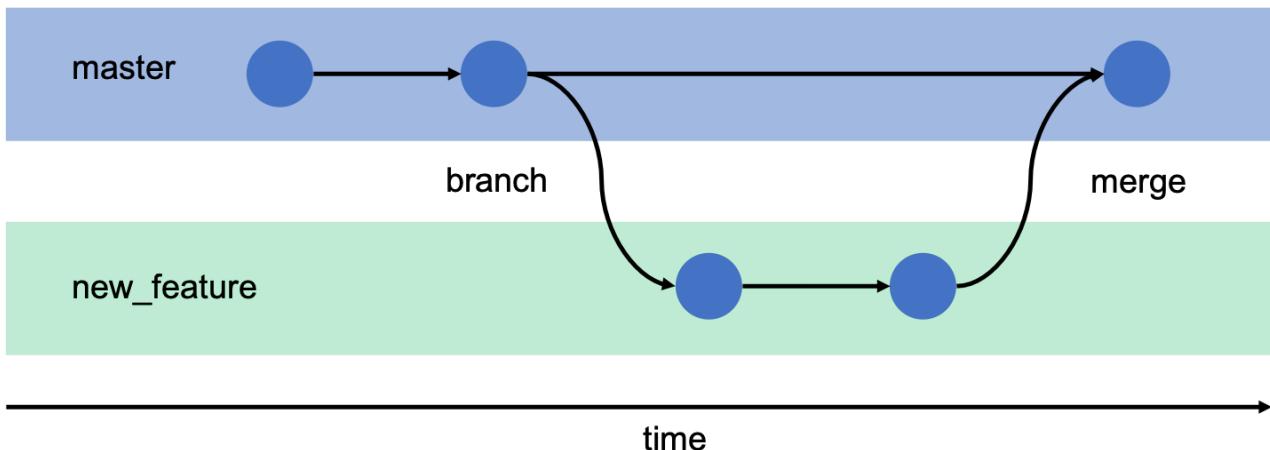
Usuario.java



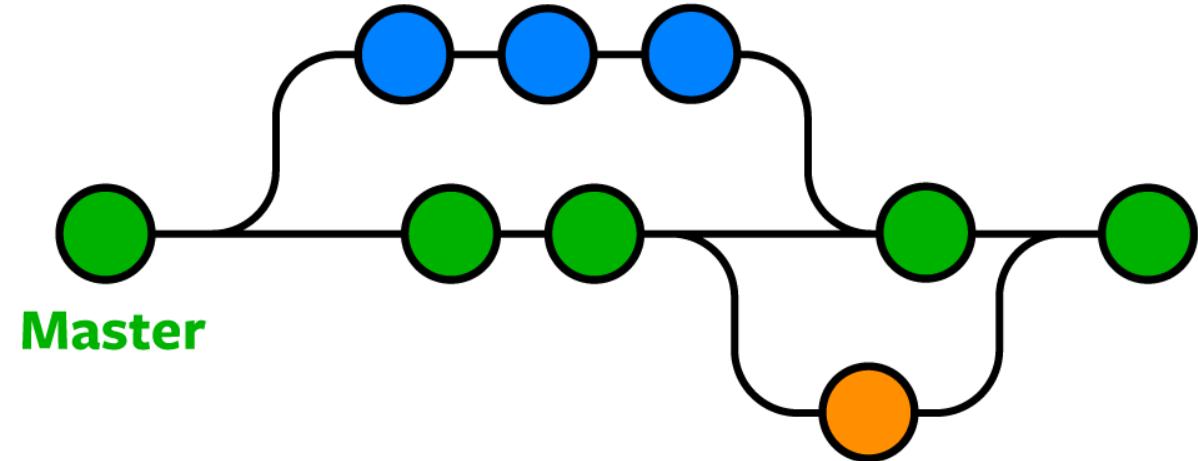
A complex network graph with many colored nodes and connections.

Ramos (branches)

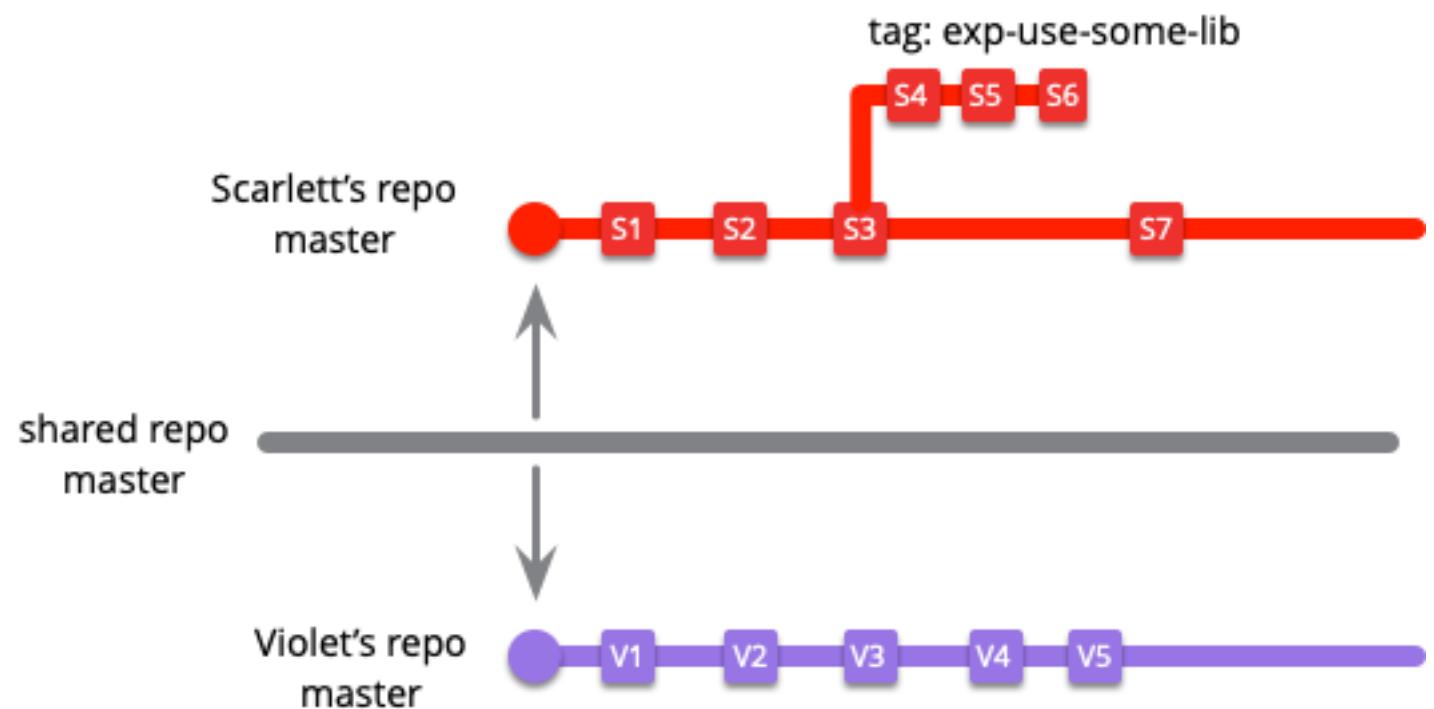
- **Versões que não seguem a linha principal de desenvolvimento**
- **Fornecem isolamento** para o processo de desenvolvimento
 - Ramos usualmente são migrados à linha principal de desenvolvimento
 - A migração pode ser complicada no caso de isolamento longo
- **O espaço de trabalho de um desenvolvedor pode ser visto como um ramo**
 - Extremamente isolado (ramos são compartilhados por outras pessoas);
 - Que reside no cliente (ramos residem no servidor);
 - Que são momentâneos (ramos são históricos);
 - Que são temporários (ramos são permanentes)

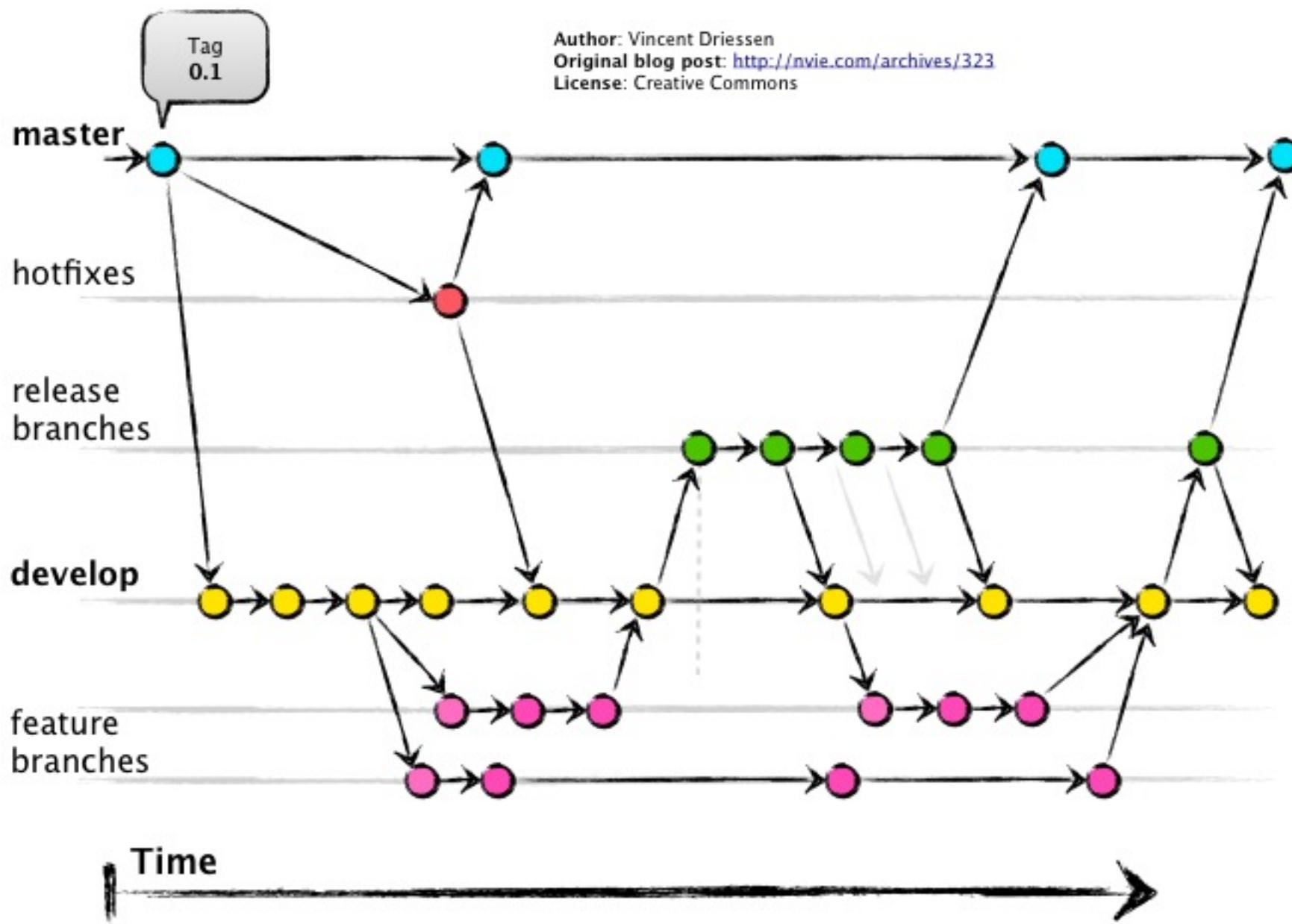


Your Work

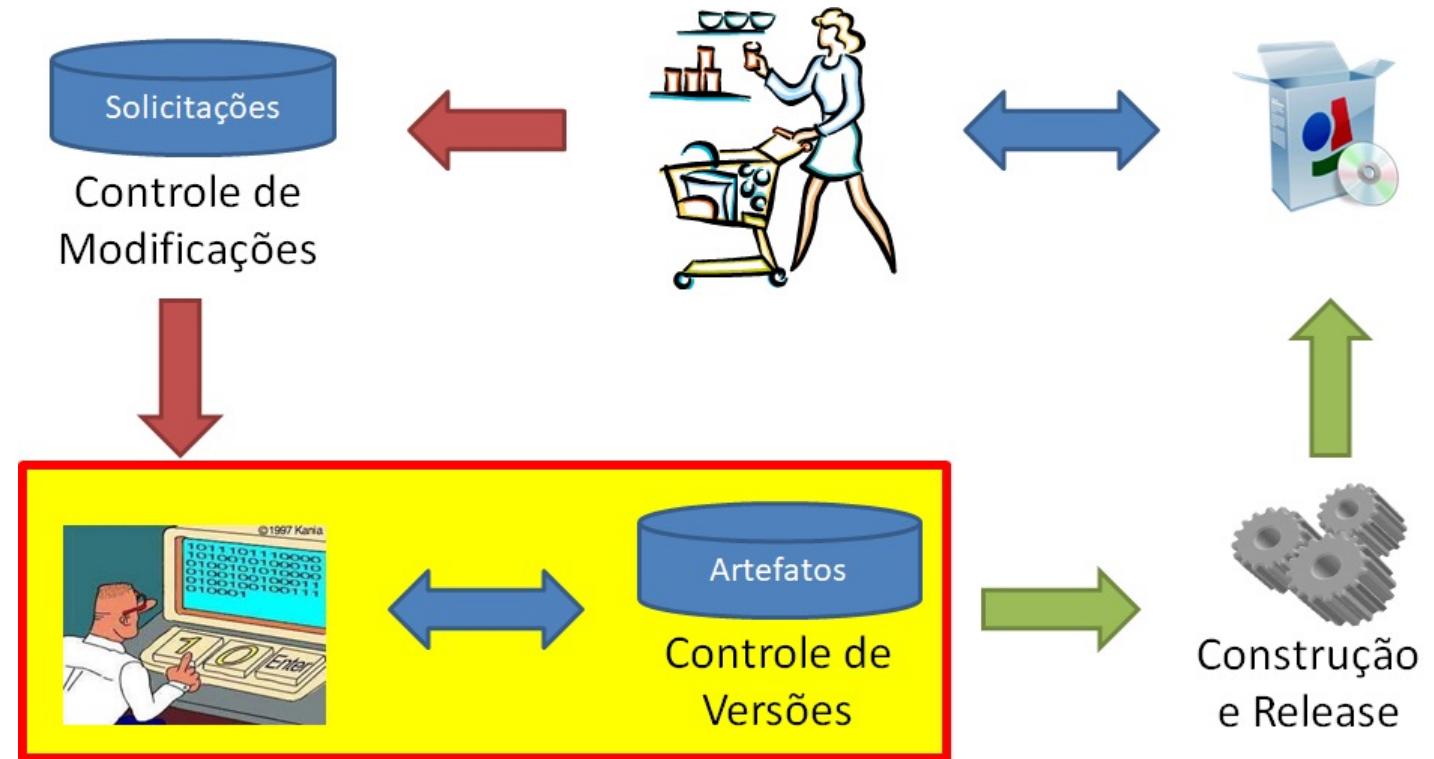


Someone Else's Work

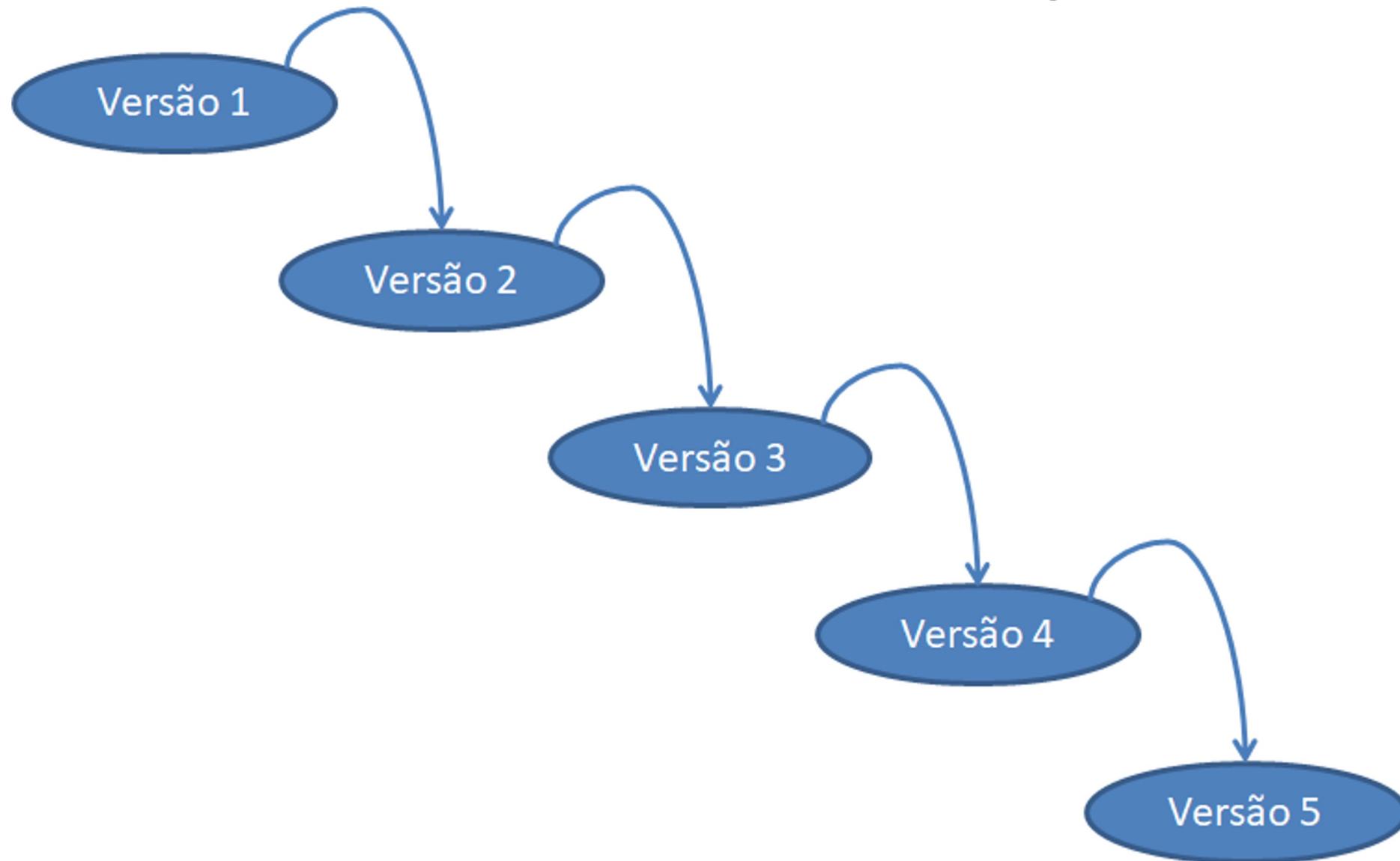




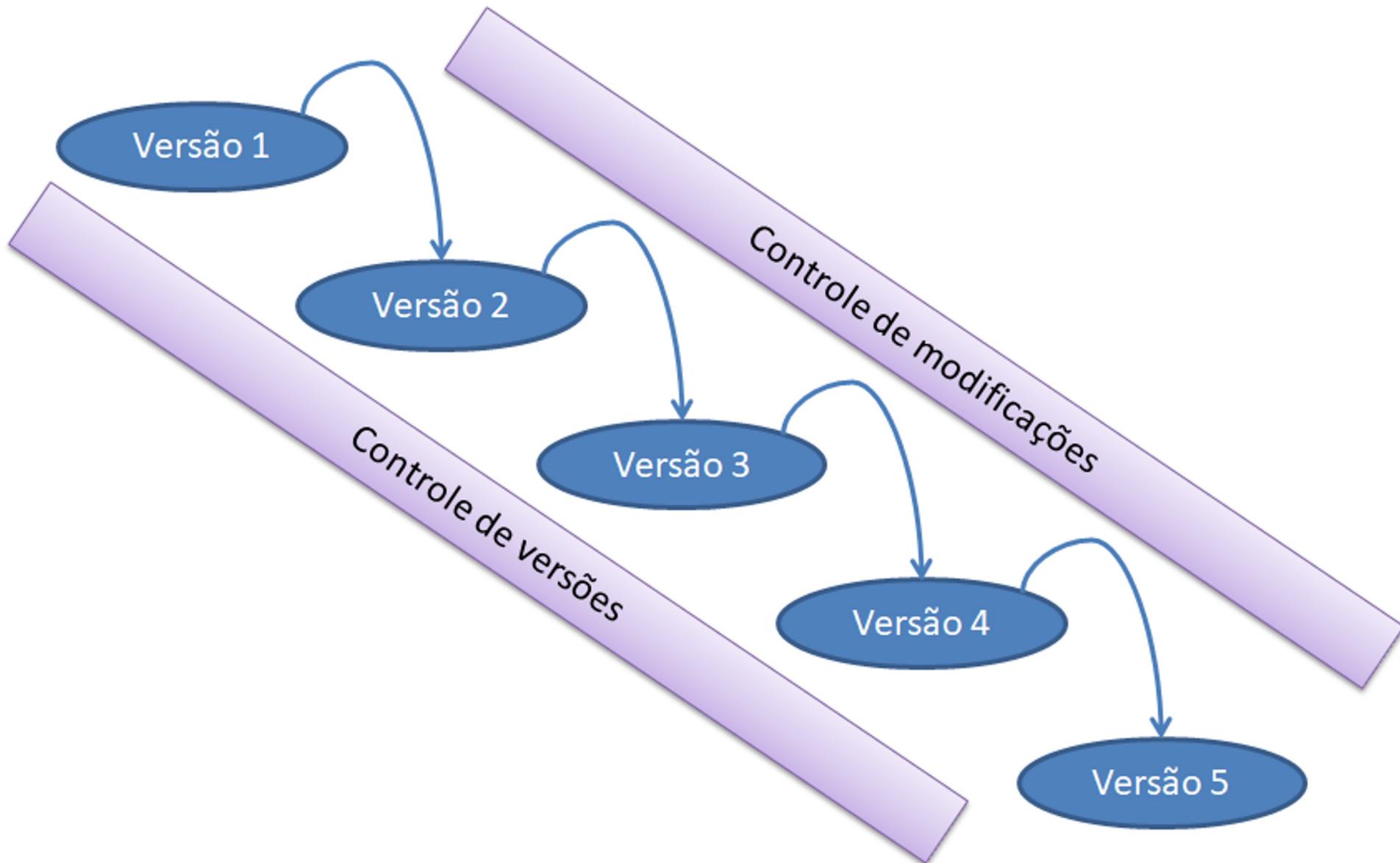
Sistema de Gerência de Configuração



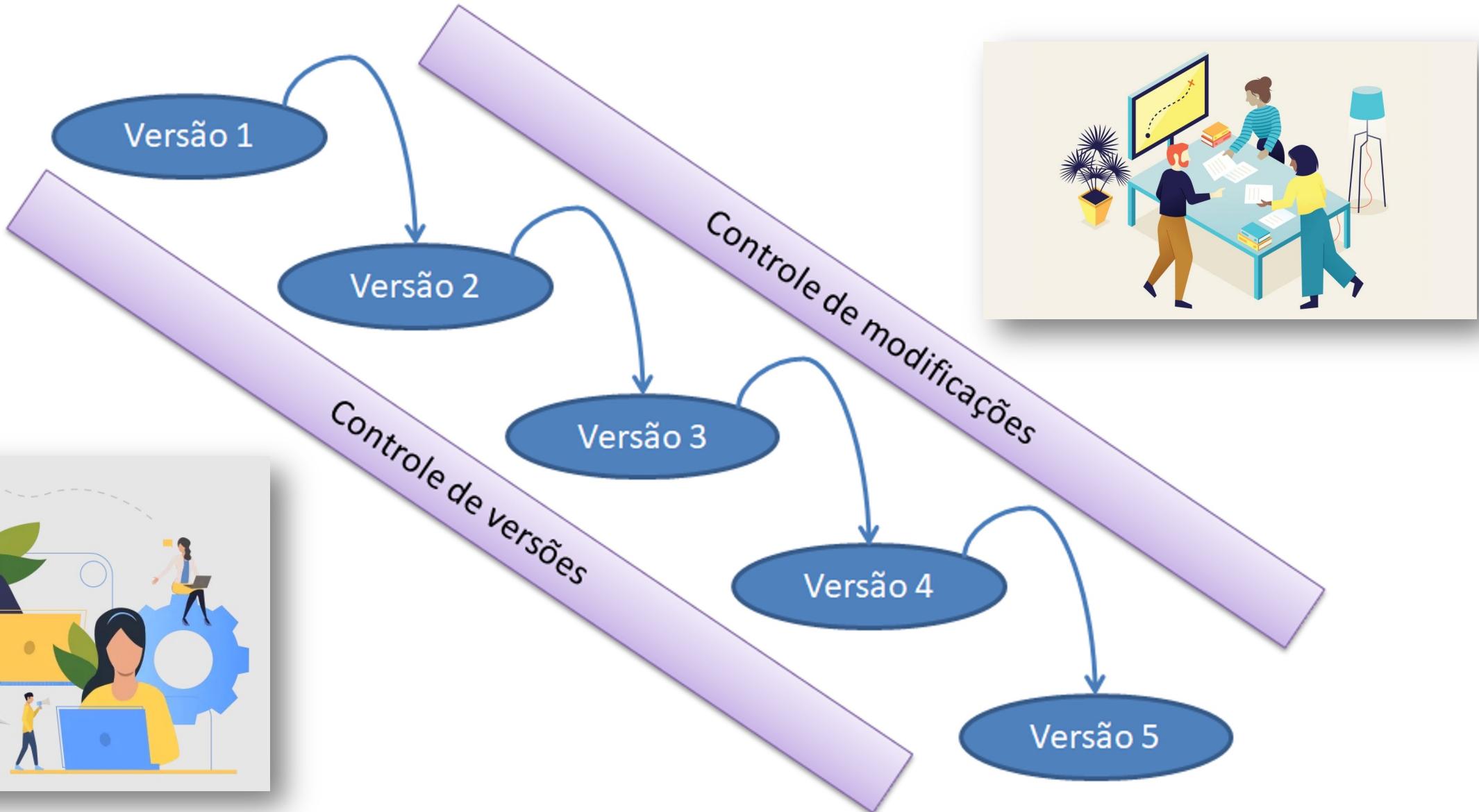
Sistema de Gerência de Configuração



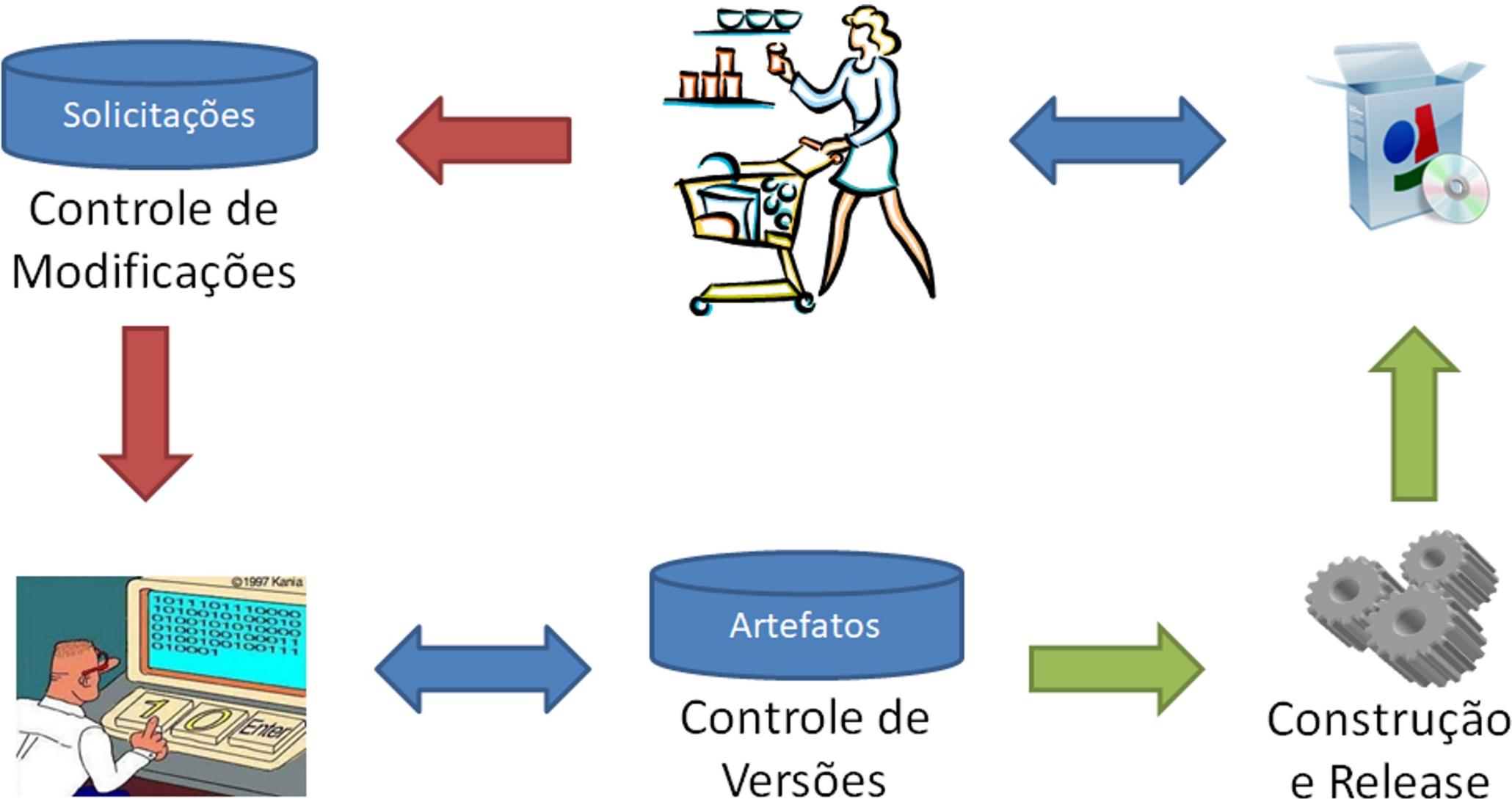
Sistema de Gerência de Configuração



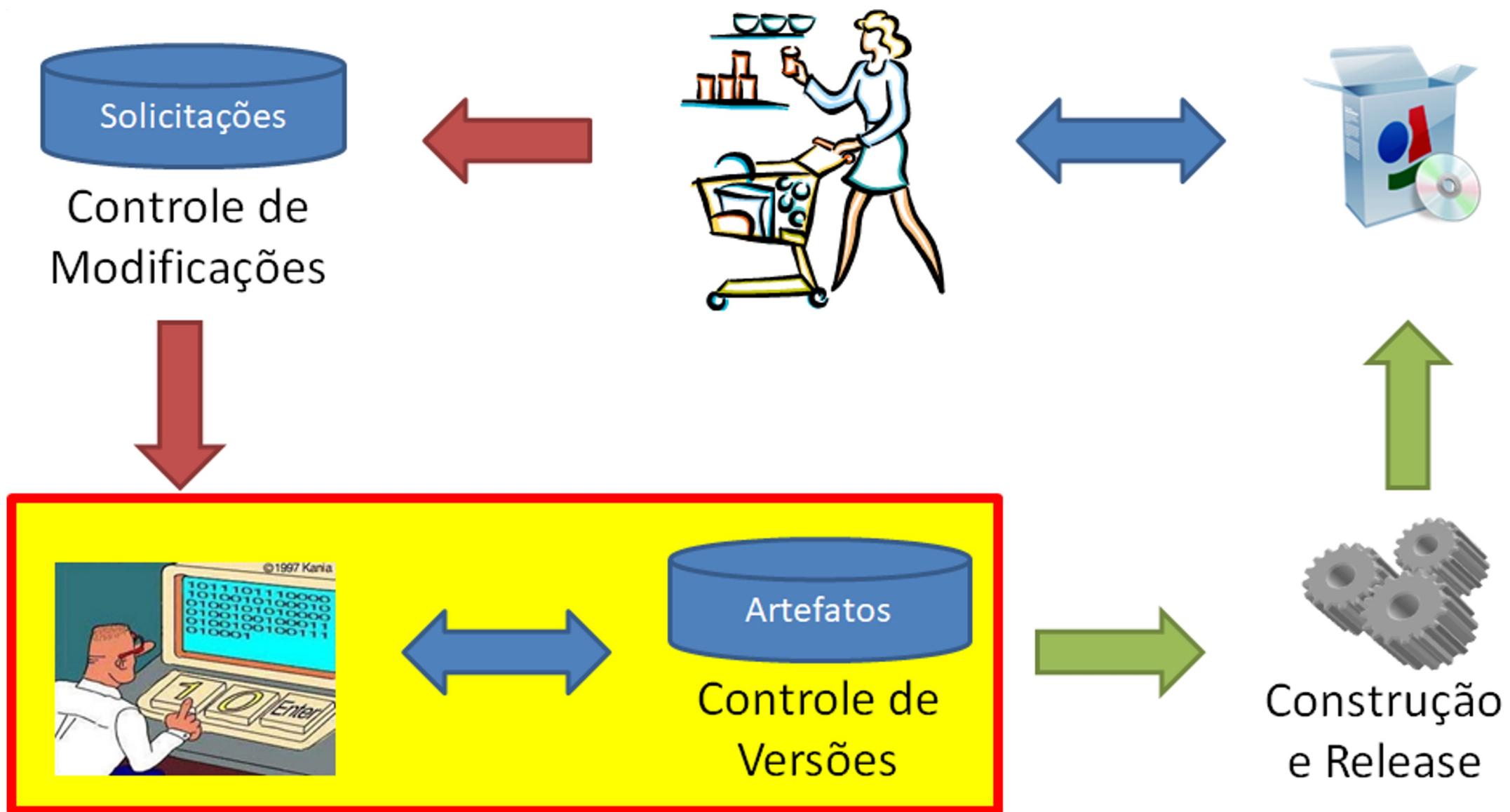
Sistema de Gerência de Configuração



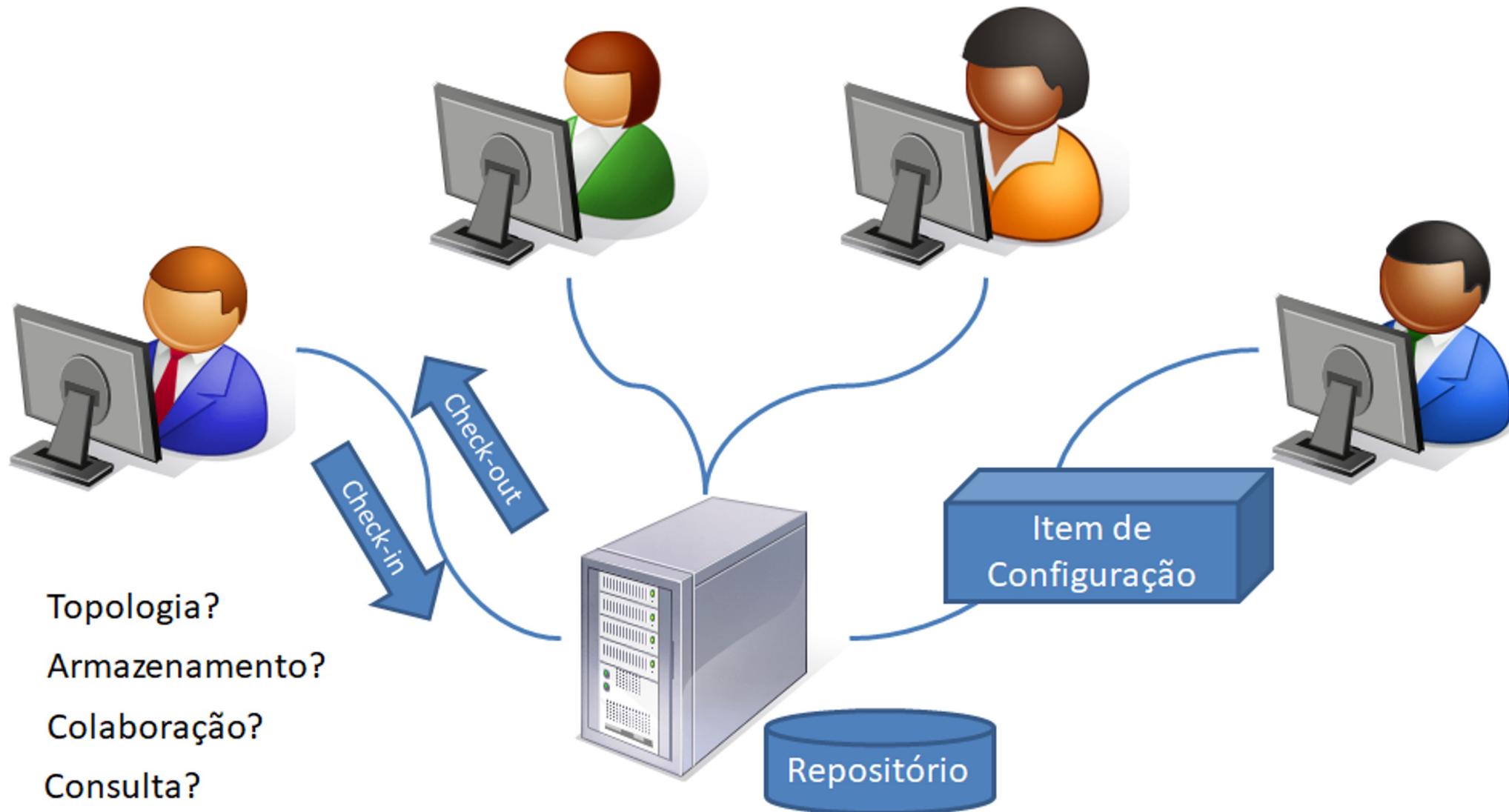
Sistema de Gerência de Configuração



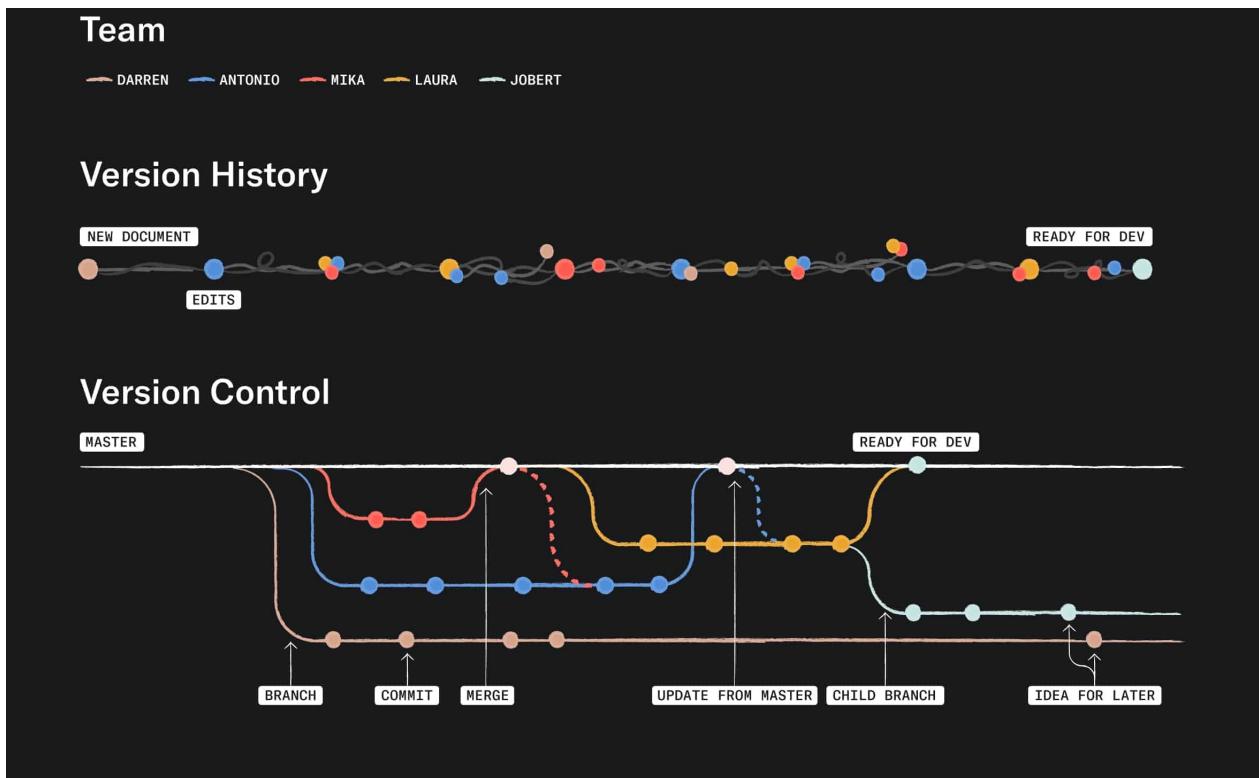
Sistema de Gerência de Configuração



Controle de versão



Para que serve o Controle de Versão?



- O controle de versão apoia o desenvolvimento de diversas maneiras:
 - **Registro do Histórico.** Registra toda a evolução do projeto, cada alteração sobre cada arquivo. Com essas informações sabe-se quem fez o quê, quando e onde. Além disso, permite reconstruir uma revisão específica do arquivo sempre que desejado

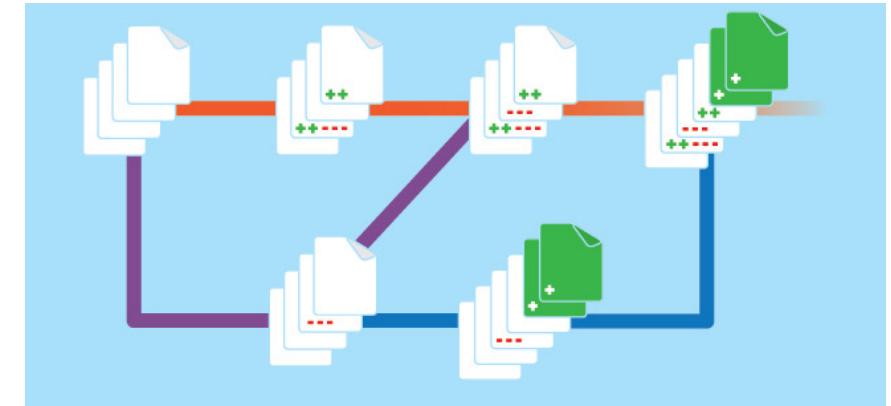
Para que serve o Controle de Versão?

O controle de versão apoia o desenvolvimento de diversas maneiras:

- **Colaboração Concorrente.** O controle de versão possibilita que vários desenvolvedores trabalhem em paralelo sobre os mesmos arquivos sem que um sobrescreva o código de outro, o que traria reaparecimento de defeitos e perda de funcionalidades

Para que serve o Controle de Versão?

- O controle de versão apoia o desenvolvimento de diversas maneiras:
 - **Variações no Projeto.** Mantém linhas diferentes de evolução do mesmo projeto. Por exemplo, mantendo uma versão 1.0 enquanto a equipe prepara uma versão 2.0



Controle de Versão

É a espinha dorsal de toda a gerência de configuração, apoiando as atividades de controle de mudança e integração contínua.

- Fornece os seguintes serviços:
 1. Identificação, armazenamento e gerenciamento dos itens de configuração e de suas versões durante todo o ciclo de vida do software;
 2. Histórico de todas as alterações efetuadas nos itens de configuração;
 3. Criação de rótulos e ramificações no projeto;
 4. Recuperação de uma configuração em um determinado momento desejado do tempo.

Situações comuns durante o Desenvolvimento

"Essa mudança não deu certo, vamos voltar como estava antes."

"O que mudou nessa rotina? Quando foi essa mudança?
Quem é o responsável?"

"Esse defeito já tinha sido corrigido, como reapareceu?"

"A nova função que implementei ontem era para estar aqui, mas sumiu!"

"Preciso daquela versão que estava no cliente. Cadê os arquivos?"

"Como fazer os testes sem parar o desenvolvimento?"

Situações comuns durante o Desenvolvimento

"Essa mudança não deu certo, vamos voltar como estava antes."

"O que mudou nessa rotina? Quando foi essa mudança?
Quem é o responsável?"

"A nova função que implementei ontem era para estar aqui, mas sumiu!"

"Preciso daquela versão que estava no cliente. Cadê os arquivos?"

Registro da evolução do projeto

"... se detecto ja tinha sido corrigido, como reapareceu?"

"Como fazer os testes sem parar o desenvolvimento?"

Situações comuns durante o Desenvolvimento

"Essa mudança não deu certo, vamos voltar como estava antes."

Controle de Concorrência

O que mudou essa rotina? Quando foi essa mudança?
Quem é o responsável?"

"Esse defeito já tinha sido corrigido, como reapareceu?"

"A nova função que implementei ontem era para estar aqui, mas sumiu!"

"Preciso daquela versão que estava no cliente. Cadê os arquivos?"

"Como fazer os testes sem parar o desenvolvimento?"

Situações comuns durante o Desenvolvimento

"Essa mudança não deu certo, vamos voltar como estava antes."

Variações de projeto

"A implementação que fizemos ontem era para estar aqui, mas sumiu!"

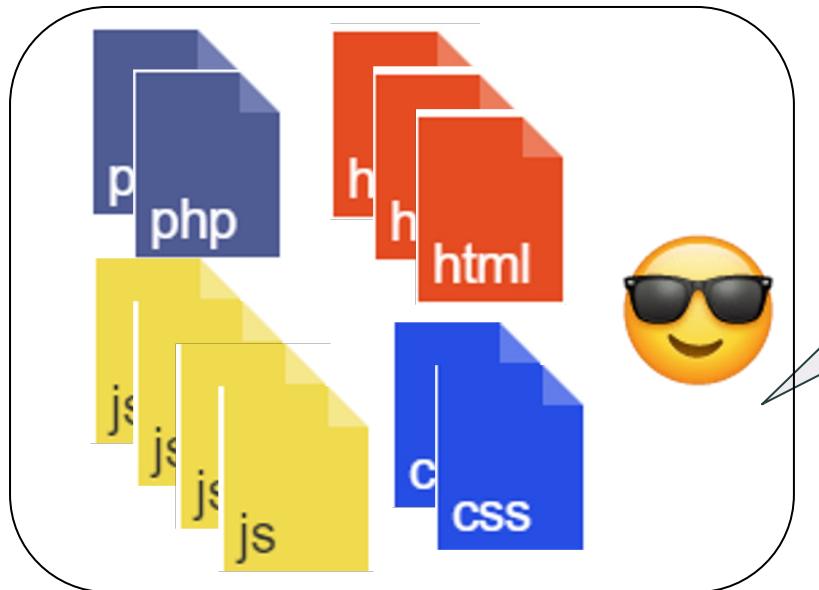
"O que mudou nessa rotina? Quando foi essa mudança?
Quem é o responsável?"

"Preciso daquela versão que estava no cliente. Cadê os arquivos?"

"Esse defeito já tinha sido corrigido, como reapareceu?"

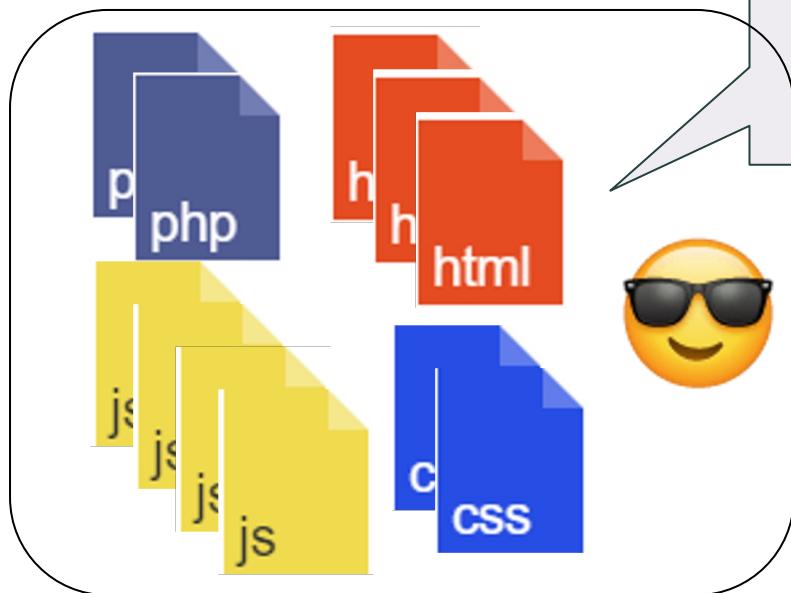
"Como fazer os testes sem parar o desenvolvimento?"

Registro da Evolução do Projeto



- Compilar
- Passa nos testes
- Ajustes
- entre outros

Registro da Evolução do Projeto



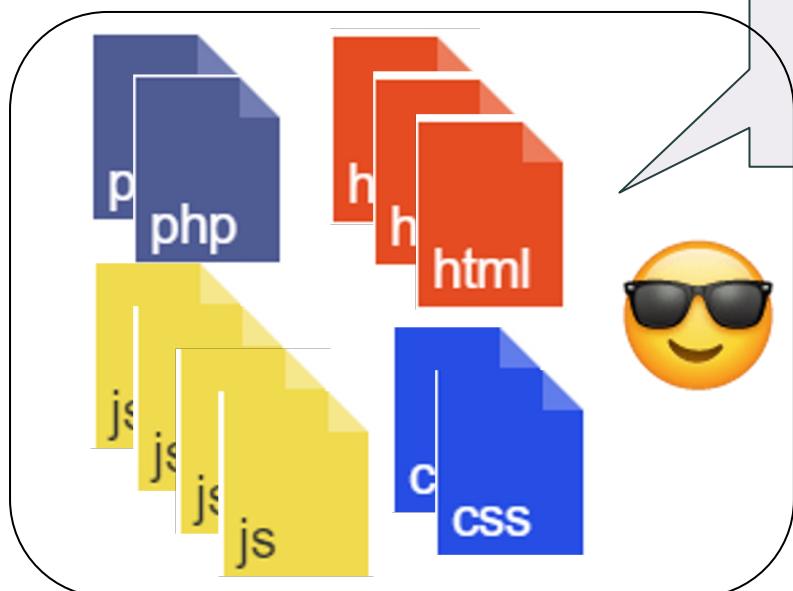
Registrar
estado dos
arquivos

- Compilar
- Passa nos testes
- Ajustes
- entre outros

Registrar
manualmente ou
compactar em
arquivos

projeto.r1.2020-02-
19.zip

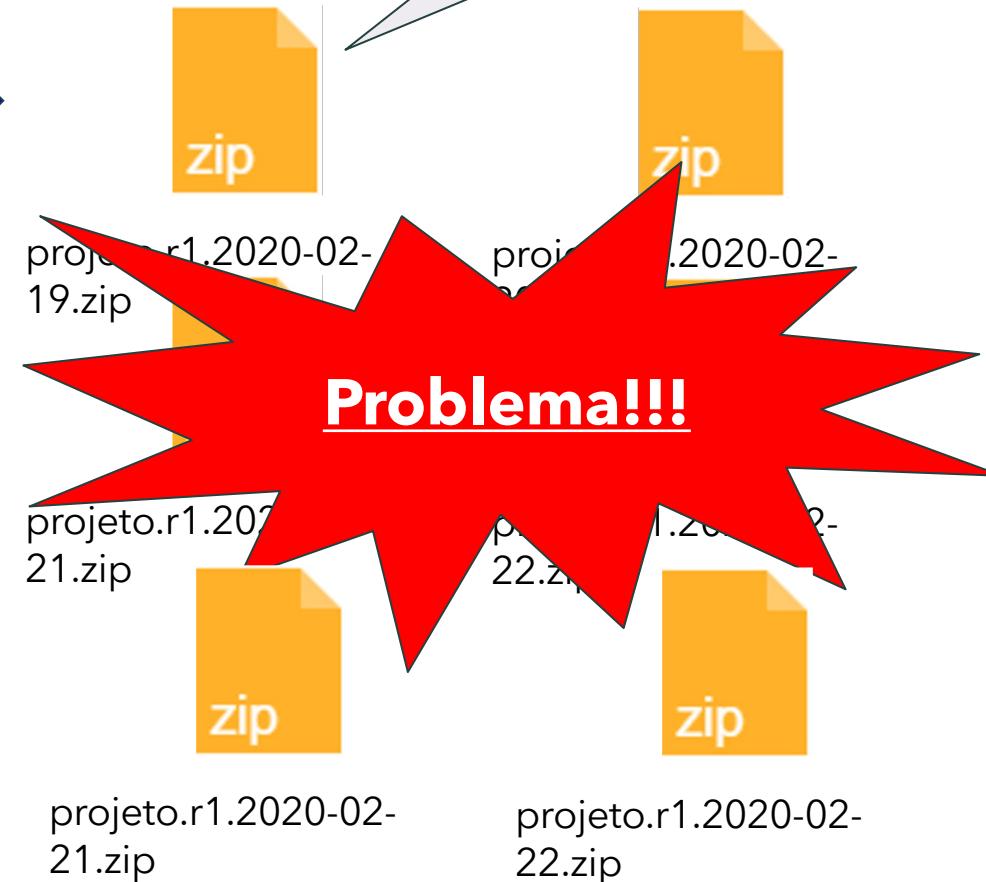
Registro da Evolução do Projeto



- Compilar
- Passa nos testes
- Ajustes
- entre outros

Registrar
estado dos
arquivos

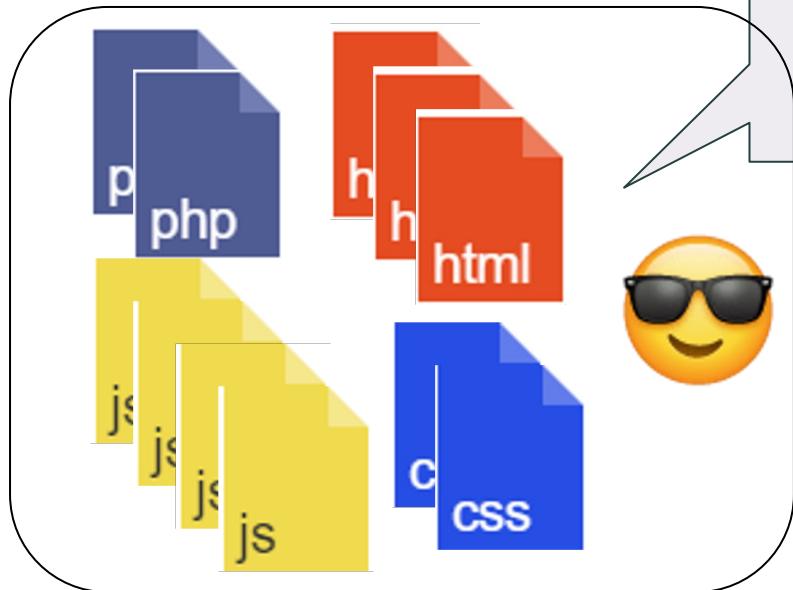
Registro manualmente
ou compactar em
arquivos



projeto.r1.2020-02-
21.zip

projeto.r1.2020-02-
22.zip

Registro da Evolução do Projeto



- Compilar
- Passa nos testes
- Ajustes
- entre outros

Registrar
estado dos
arquivos

Registro manualmente
ou compactar em
arquivos



- Acúmulo de material de backup
- Dificuldade em gerenciar os arquivos
- Dificuldade em extrair informações úteis para o projeto

Só Backups não bastam



Controle de versão
precisa responder:



O que mudou?

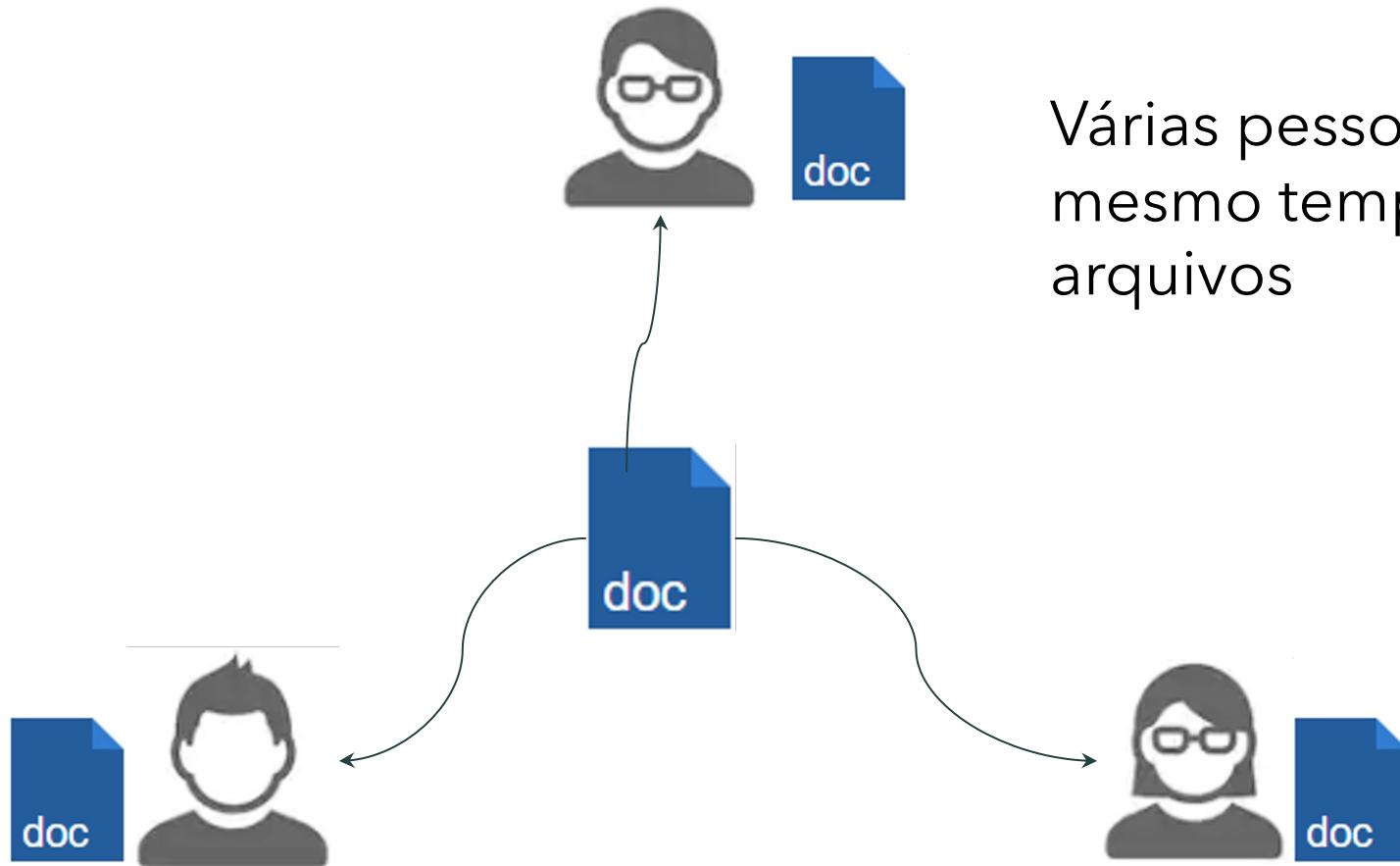


Quem mudou?



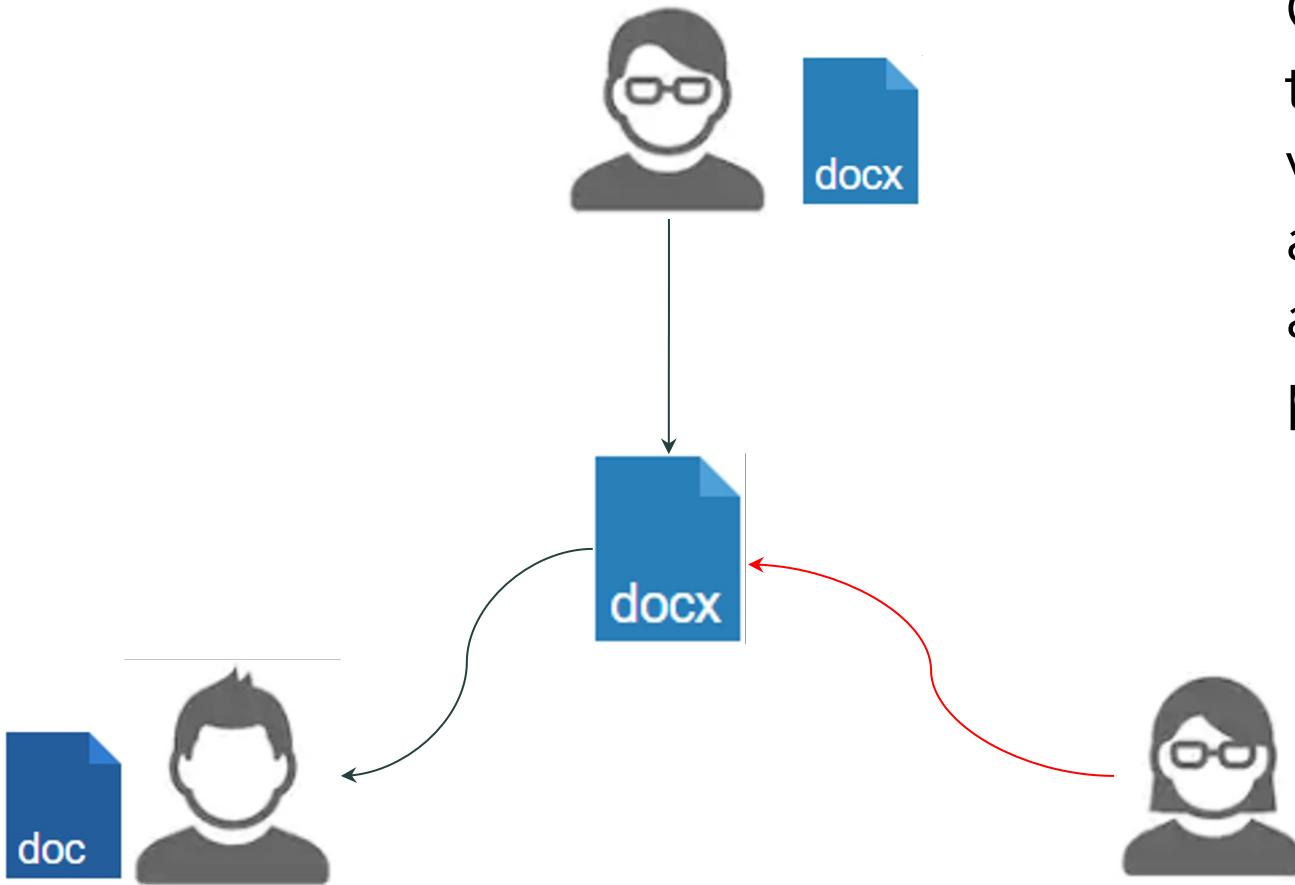
Quando mudou?

Trabalho em Equipe



Várias pessoas vão editar ao mesmo tempo os mesmos arquivos

Trabalho em Equipe



Caso não exista algum tipo de controle de versão, uma pessoa acaba sobreescrivendo as alterações realizadas pelos outros usuários.

- perda de código
- desaparecimento de novas implementações
- reaparecimentos de bugs

Mecanismos de Controle de Concorrência

Travamento

Mesclagem de arquivos

Ramos individuais

Estudaremos em outras aulas....

Variações de Projeto



Fornecem outro nível de paralelismo

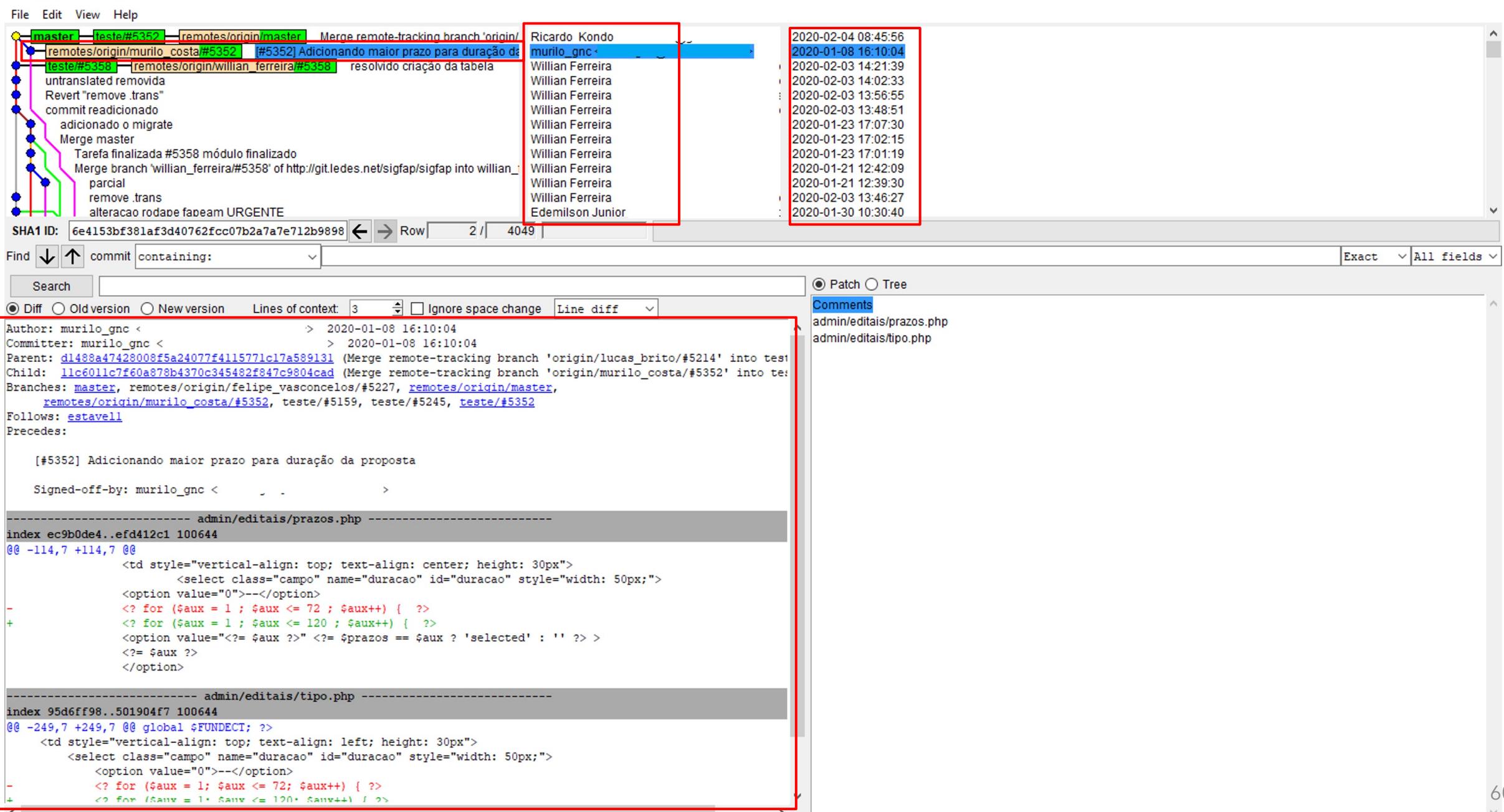


Lidar com atividades distintas de desenvolvimento



Exemplos:

Manter diferentes versões em produção
Desenvolver novas funcionalidade da próxima versão



Papeis em GCS (1)

Papel	Responsabilidade
Gerente de Configuração	<ul style="list-style-type: none">• Controlar as <i>baselines</i> dos produtos;• Gerar as versões do produto que contemplem demandas de projetos e de solicitações de mudança;• Realizar auditorias para garantir o cumprimento das atividades de CM pela equipe de trabalho.
Gerente de Produto	<ul style="list-style-type: none">• Solicitar estabelecimento de <i>baselines</i>;• Participar do CCB, tendo poder de aprovação nas solicitações de mudança que não impactam custo, prazo e outros compromissos externos.
Gerente de Projeto	<ul style="list-style-type: none">• Participar do CCB nas decisões de solicitações de mudanças de requisitos que impactam custos, prazos e outros compromissos externos (com o cliente final).
Relator	<ul style="list-style-type: none">• Reportar solicitações de mudança (erros ou melhorias) para a fábrica analisar e executar as atividades relacionadas à solicitação, quando aprovada.

Papeis em GCS (2)

Papel	Responsabilidade
Gerência Sênior	<ul style="list-style-type: none">• Analisar os relatórios de status da Gerência de Configuração e tomar ações para resolução de possíveis conflitos encontrados.
Analista Líder	<ul style="list-style-type: none">• Liderar equipe de desenvolvimento vinculada a projetos, respondendo ao gerente de produto e ao gerente do projeto vinculado à equipe quanto ao andamento das atividades da equipe dentro da fábrica.
Equipe de desenvolvimento	<ul style="list-style-type: none">• Realizar suas atividades que envolvem gerência de configuração, cumprindo as definições do processo de CM.