



GERÊNCIA DE CONFIGURAÇÃO

PARTE II

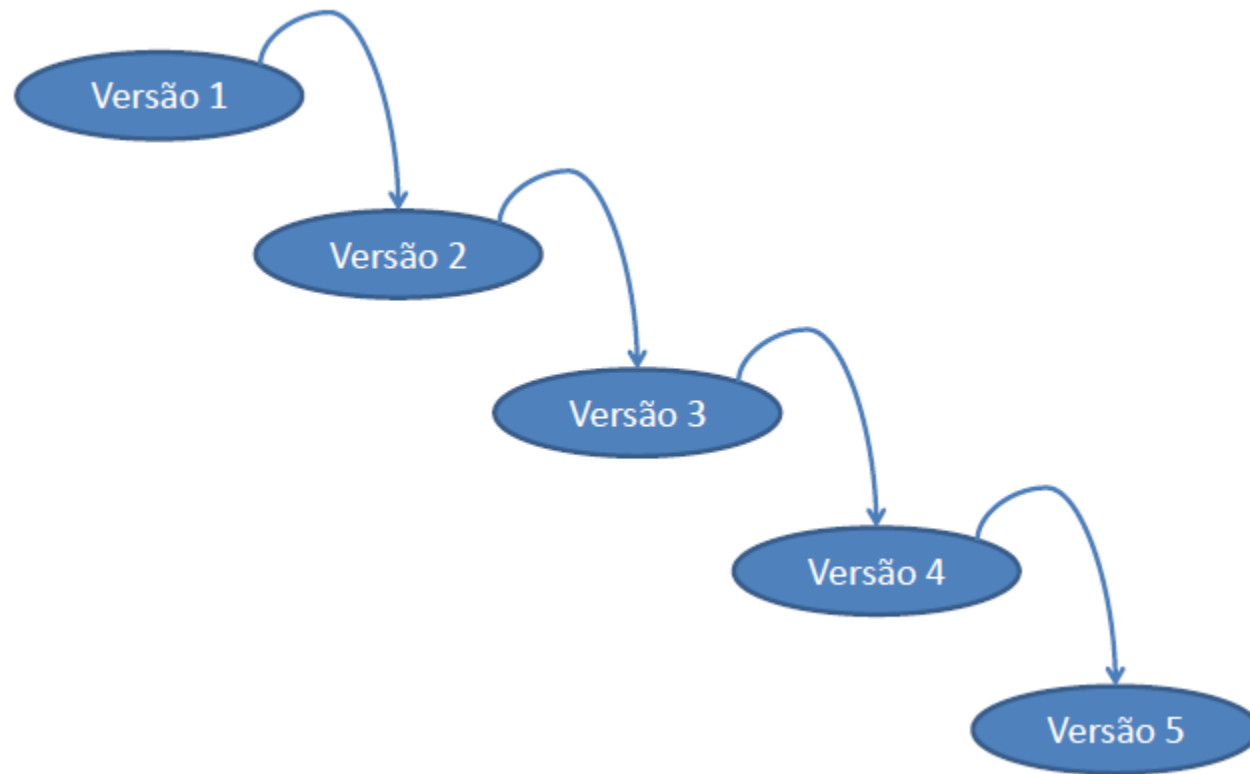
Daricélio Moreira Soares

VERSÕES

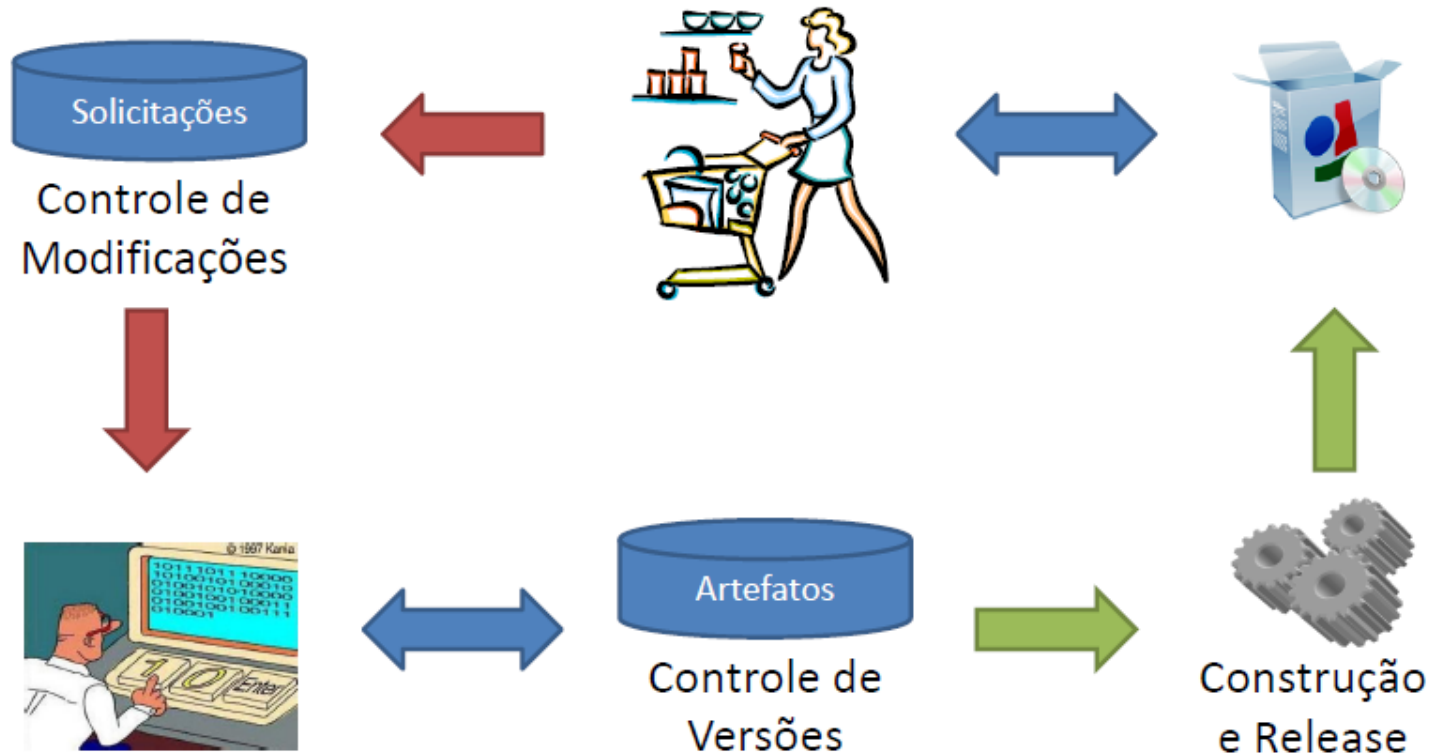
- Instâncias de um mesmo item de configuração que diferem entre si em algo
- **Revisões:** versões criadas para substituir versões anteriores seguindo uma linha temporal (e.g., em resposta a correção ou evolução)
- **Variantes:** versões coexistentes, projetadas para propósitos distintos (e.g., em resposta a diferentes arquiteturas de hardware ou sistemas operacionais)



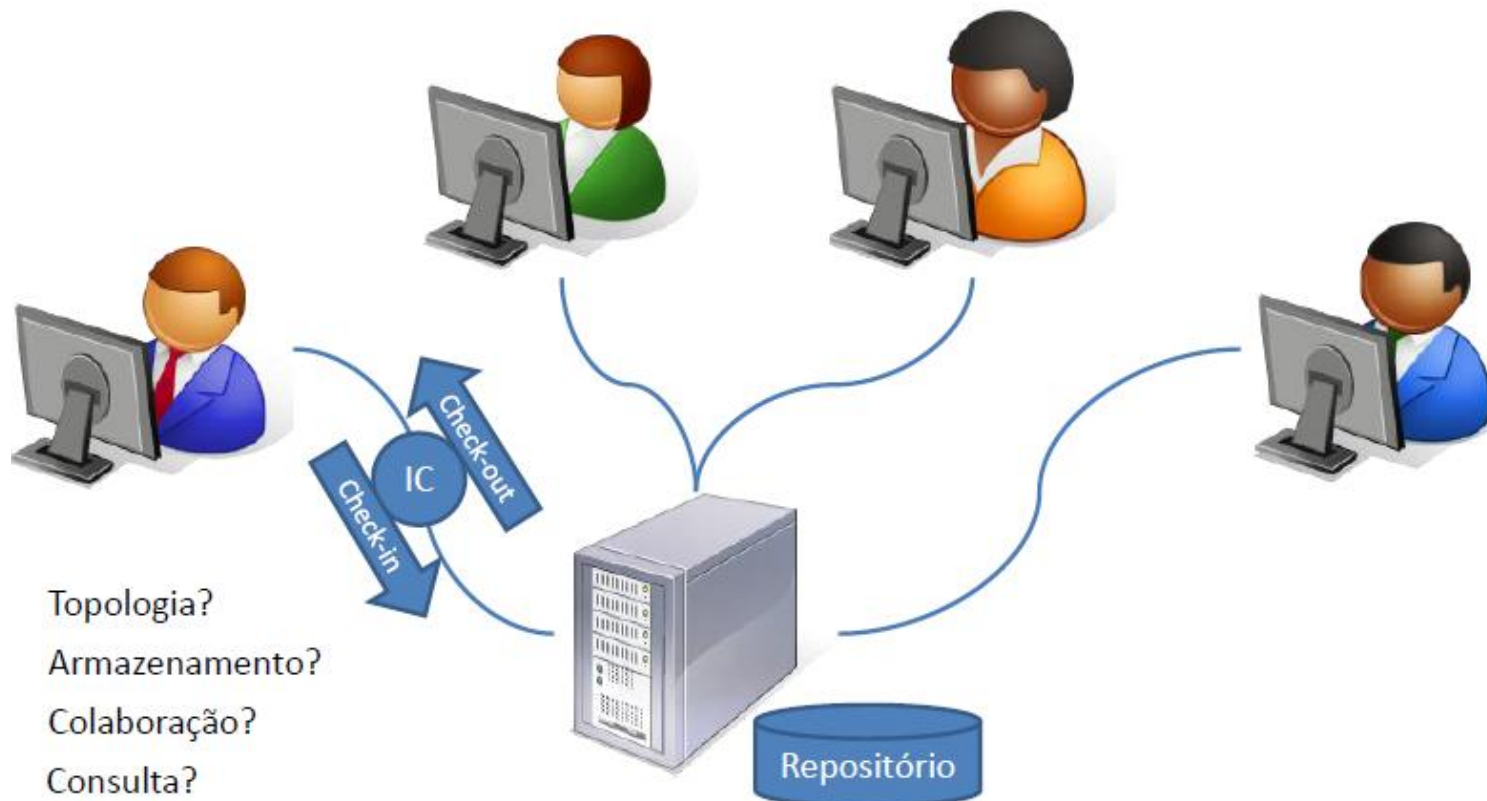
SISTEMA DE GERÊNCIA DE VERSÃO



SISTEMA DE GERÊNCIA DE VERSÃO



CONTROLE DE VERSÕES

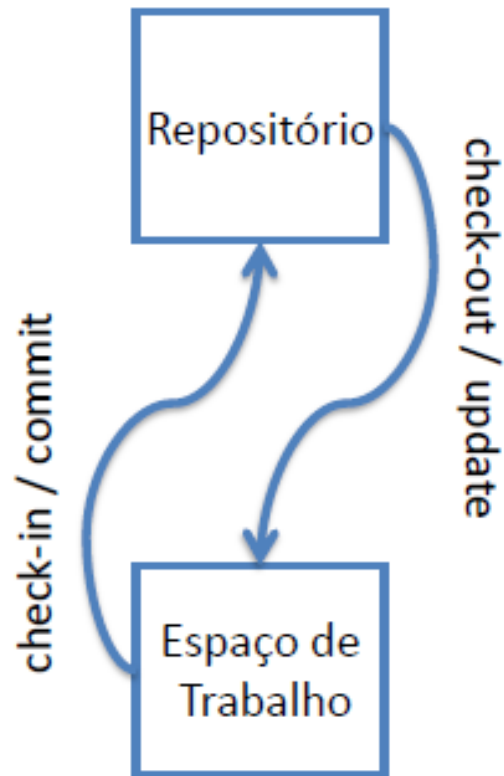


REPOSITÓRIO

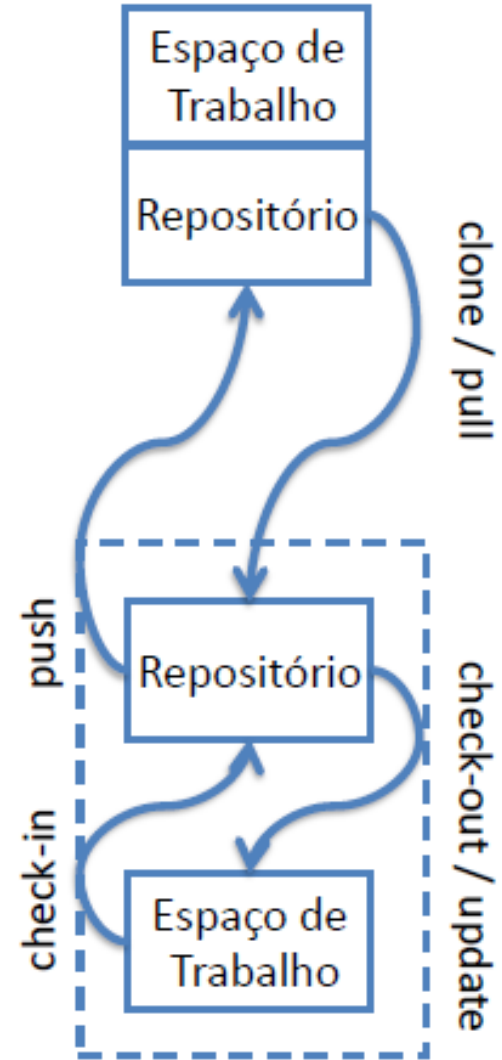
- Local onde os ICs são armazenados
 - Armazena o histórico do projeto
 - Controle na entrada e saída de ICs
 - Poucos por projeto (normalmente, somente um)
- Utiliza diferentes mecanismos de armazenamento
 - Versionamento completo
 - Versionamento de diferenças (delta)
- Utiliza diferentes mecanismos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



TOPOLOGIA



Centralizado



Distribuído



ARMAZENAMENTO

- Versionamento completo
 - Demanda grande espaço em disco
 - Permite rápida recuperação dos ICs
- Versionamento de diferenças
 - Reduz o espaço requerido em disco
 - Qualquer versão pode ser derivada a partir da aplicação dos *deltas* sobre a versão base
 - Pode demandar grande carga de processamento para recuperar ICs
 - Tipos existentes: *forward*, *reverse*



ARMAZENAMENTO

v.3

v.2

v.1

Completo

delta 2→3

delta 1→2

v.1

Forward

v.3

delta 3→2

delta 2→1

Reverse

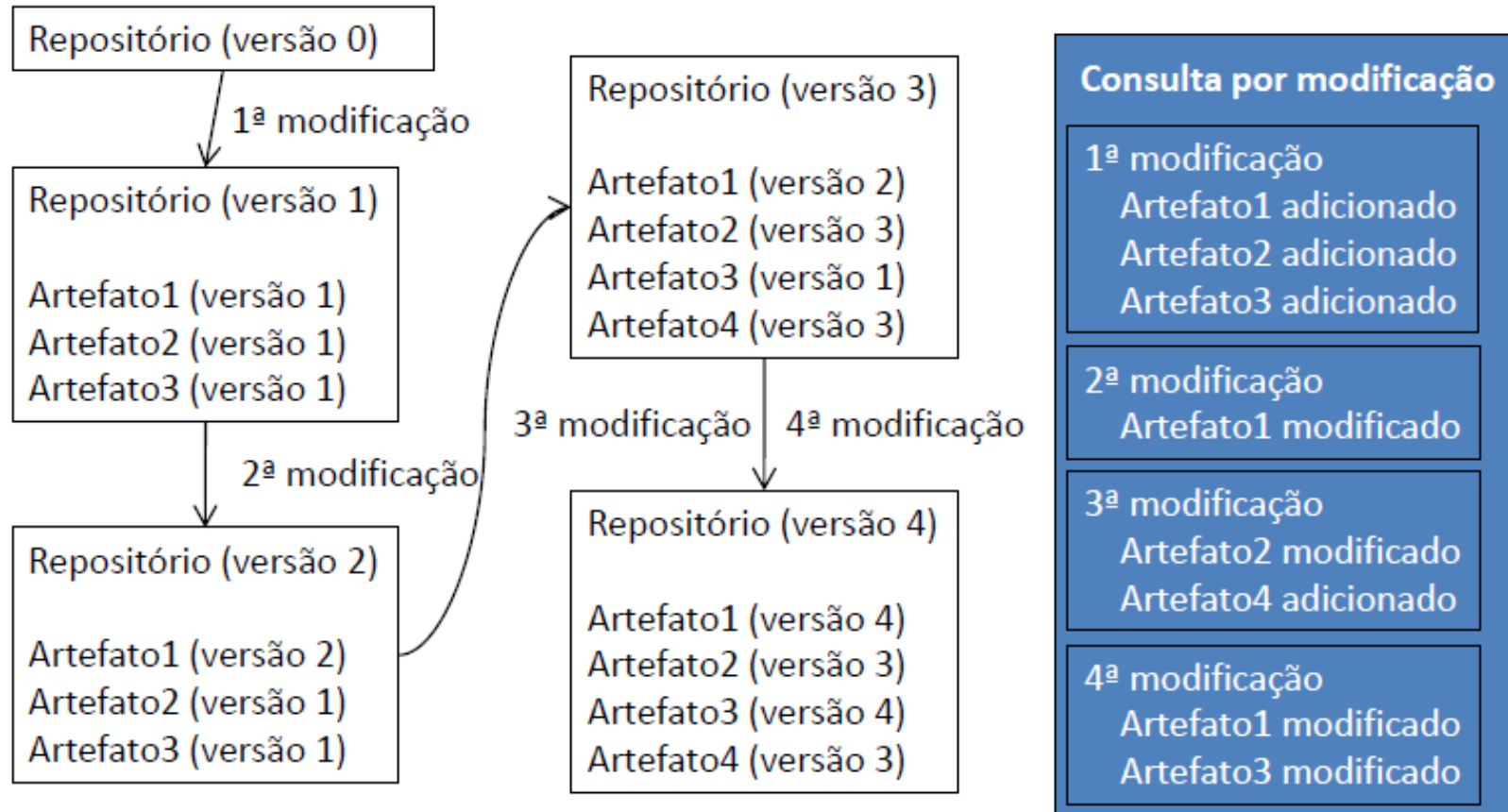


COLABORAÇÃO

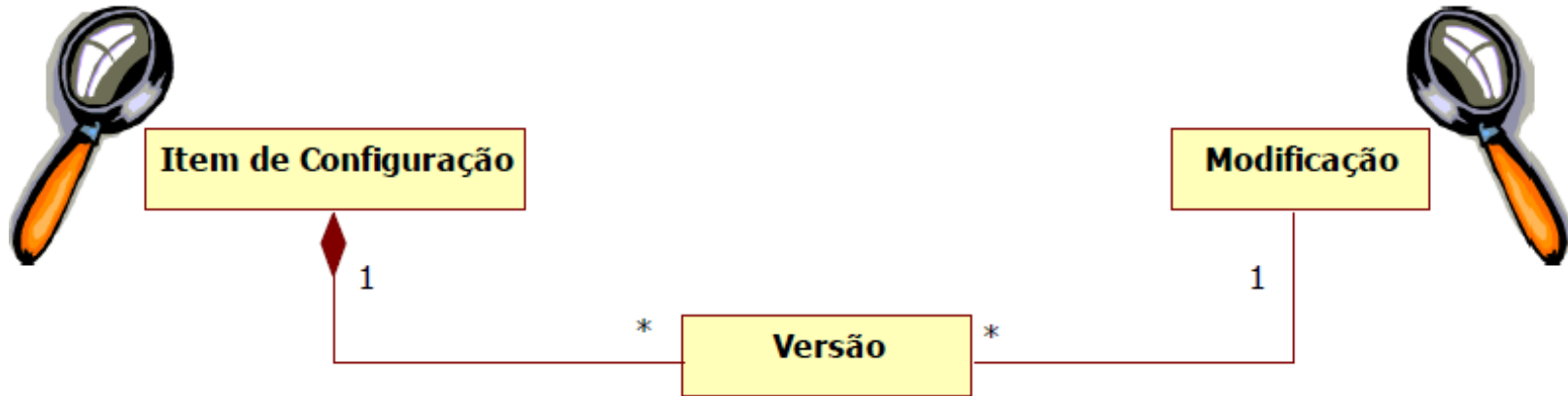
- Controle de concorrência pessimista
 - Somente um desenvolvedor pode modificar o IC em um dado momento
 - Custo zero de junção de trabalho
 - Ausência de paralelismo no desenvolvimento
- Controle de concorrência otimista
 - Vários desenvolvedores pode modificar um mesmo IC ao mesmo tempo
 - Alto custo 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 mais está modificando o IC
 - Bom custo/benefício entre controle otimista e pessimista



CONSULTA



CONSULTA



Arquivo 5
Versão 1
Versão 2
Versão 4

Modificação 4
Arquivo 2
Arquivo 5
Arquivo 7



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)



JUNÇÃO

Processo de migração de

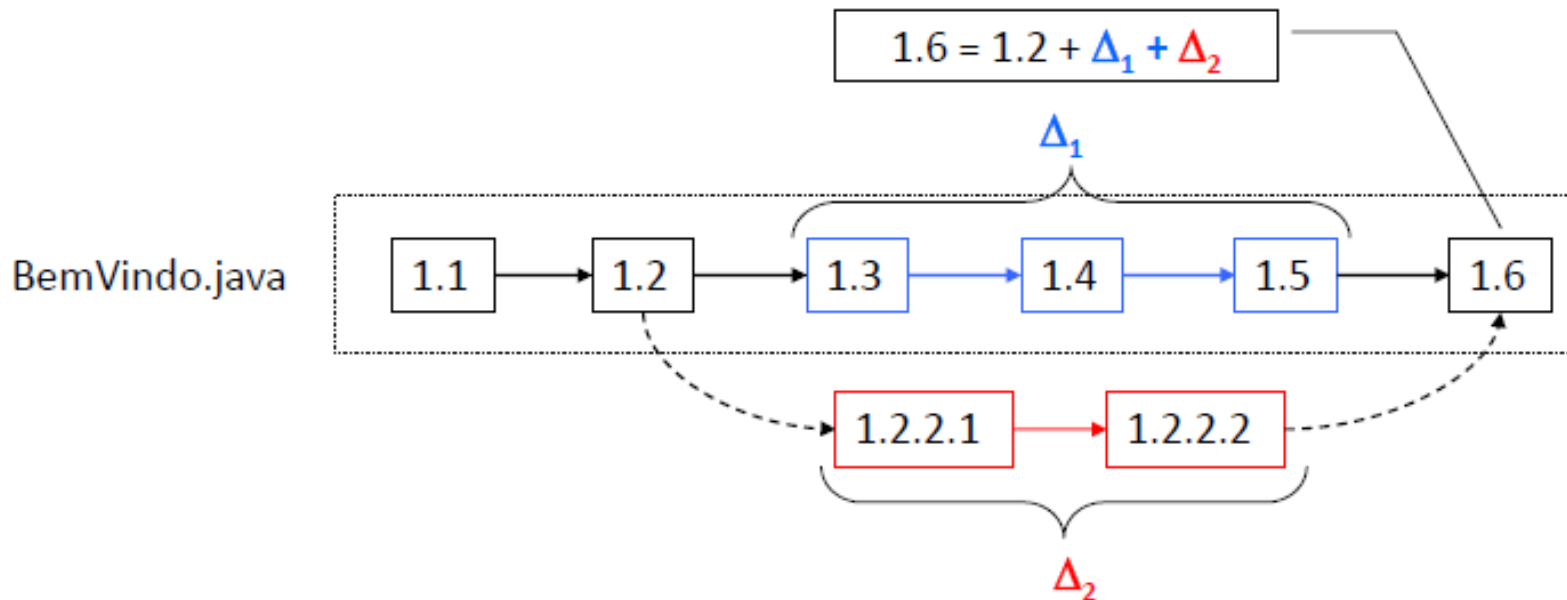
- Espaços de trabalho
- Ramos

É necessário inclusive em desenvolvimento seqüencial (*check-out reservado*) *quando são utilizados ramos*



EXEMPLO DE JUNÇÃO

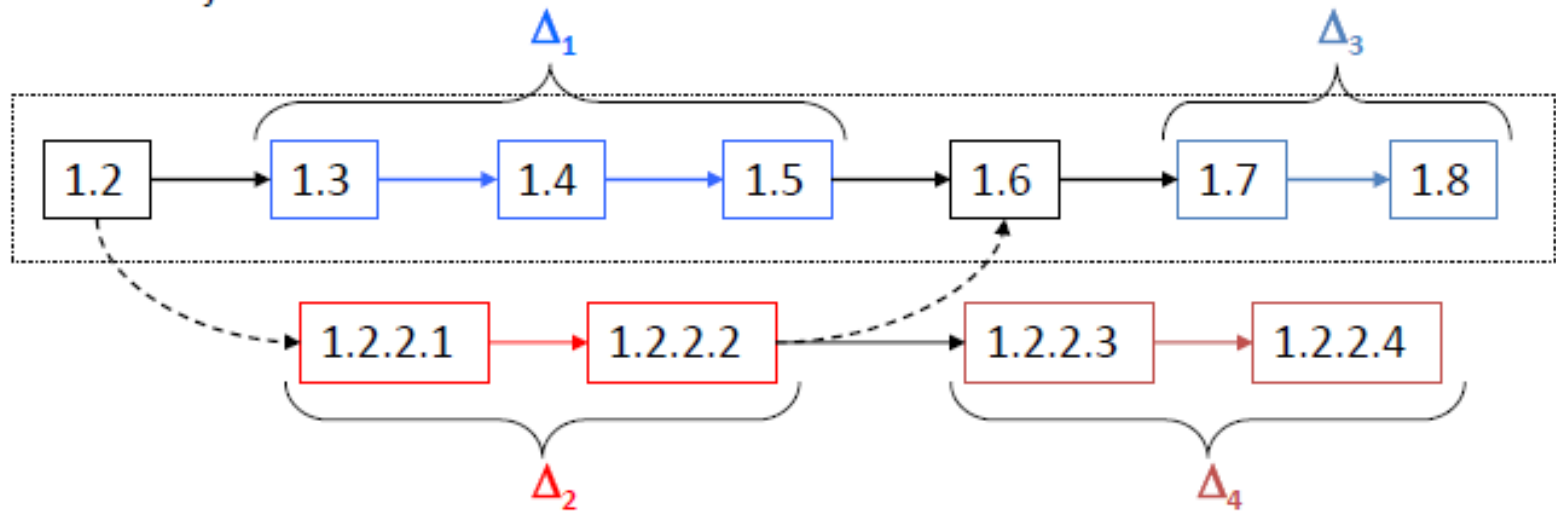
- A junção é efetuada para cada artefato do ramo
- São levadas em consideração todas as modificações desde o ancestral em comum



EXEMPLO DE JUNÇÃO (INCREMENTAL)

- O que fazer quando o ramo continua evoluindo mesmo depois da junção?

AloMundo.java



AVALIAÇÃO N2.1

- Dia 18/10 – ÀS 09:20HS
 - Evolução
 - Gerência de Configuração – Partes I e II

