Instituto de Computação

Gerência de Configuração: Terminologia

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br

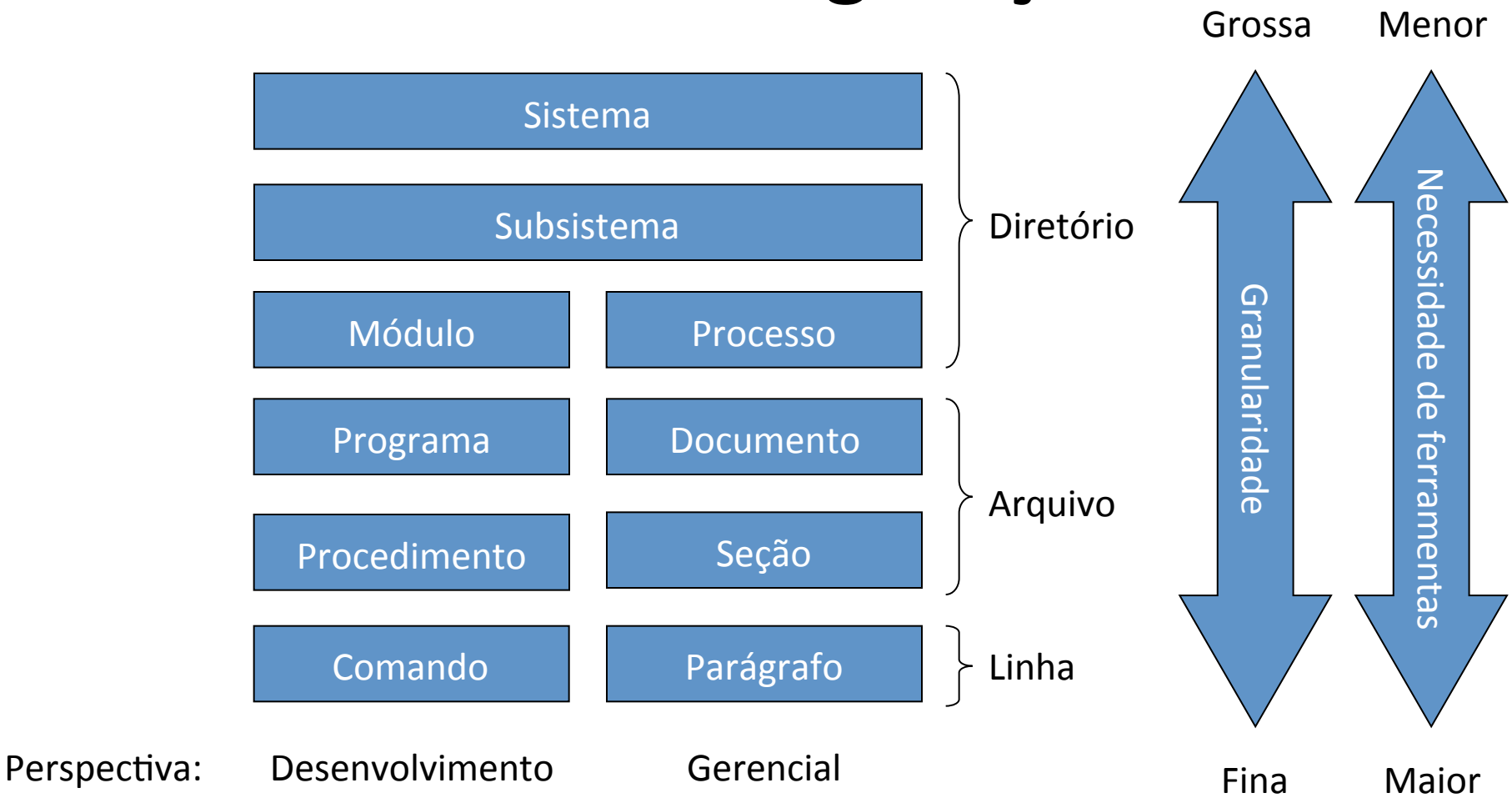
Item de configuração

- Agregação de hardware e/ou software que será passível de gerência de configuração e tratado como um elemento único
- Tipos de ICs
  - Produtos de trabalho do projeto
  - Produtos de trabalho de processos
- Exemplos: plano de GC, requisitos, modelos, código-fonte, etc.

Item de configuração

- A seleção de ICs deve levar em conta princípios como acoplamento e coesão
- ICs com alto acoplamento tornam complexo o processo de construção
  - Muitas dependências para outros ICs
- ICs com baixa coesão tornam o processo de desenvolvimento complexo
  - Vários desenvolvedores concorrendo para modificar o IC
- GCS é altamente beneficiada por sistemas com arquitetura corretamente definida

Item de configuração



Item derivado

- Item de configuração que pode ser obtido a partir de outro item de configuração (item fonte)
- Exemplo
  - Os itens de configuração que compõem o código-fonte são itens fonte para o programa executável, que é item derivado
- Estratégias
  - Versionamento do item derivado
  - Documentação do processo de derivação (roteiro, ferramentas, ambiente, etc.)

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

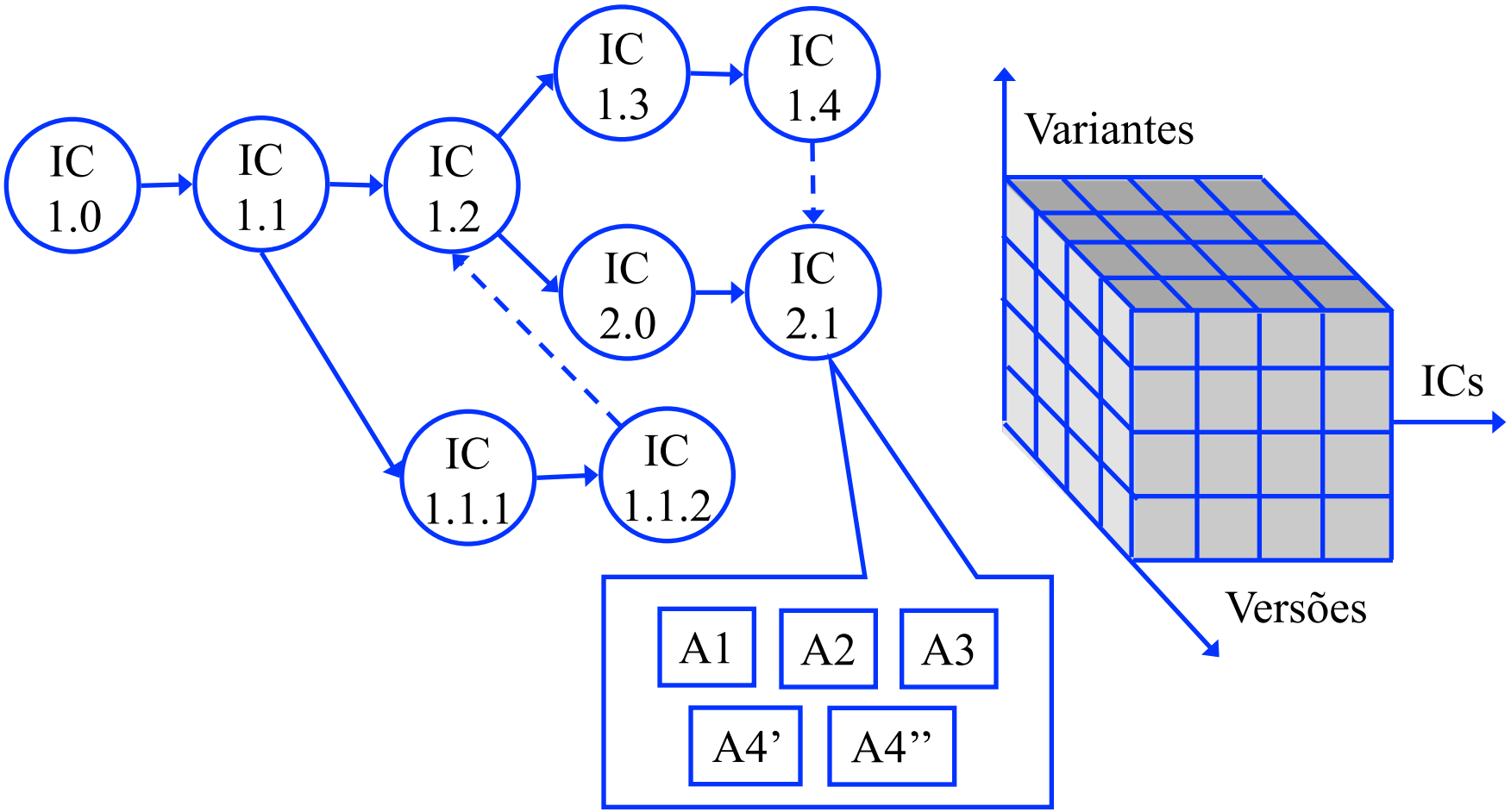
Versões

- 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)

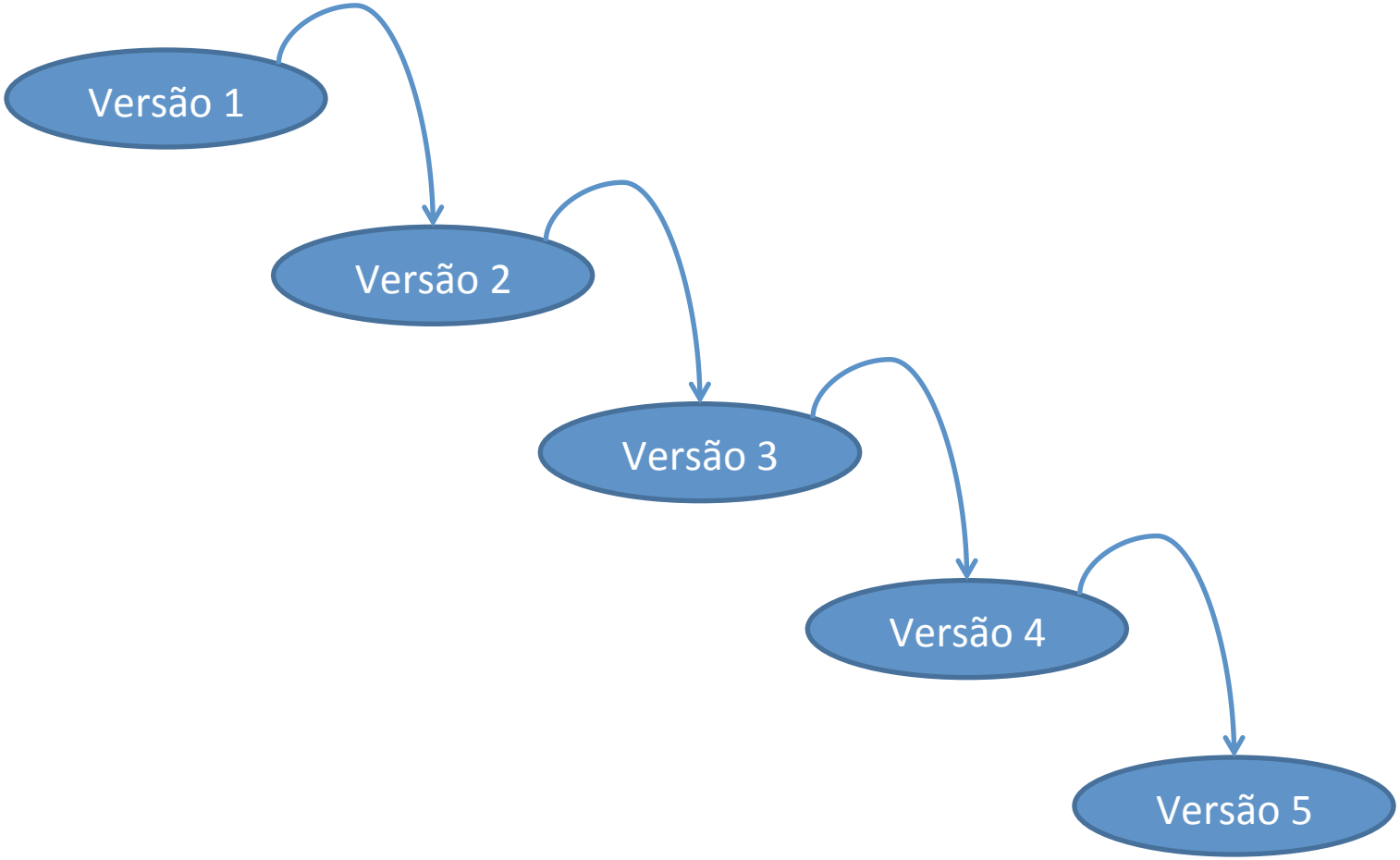
- **Variantes:** versões coexistentes, projetadas para propósitos distintos (e.g., em resposta a diferentes arquiteturas de hardware ou sistemas operacionais)

Versões

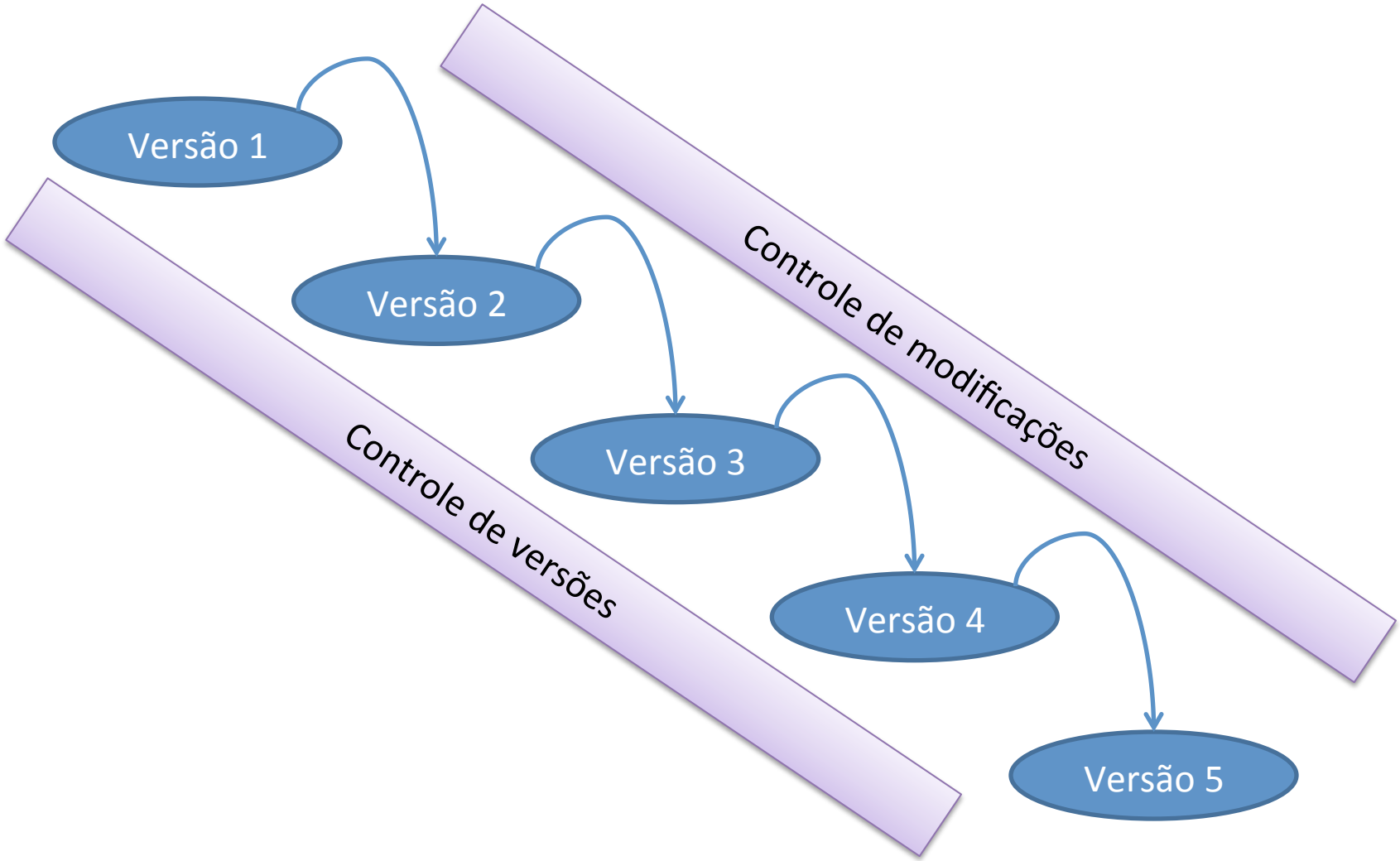


[Pressman, 1997] ICs, versões e variantes

Sistema de Gerência de Configuração



Sistema de Gerência de Configuração



Sistema de Gerência de Configuração

Versão 1

Versão 2

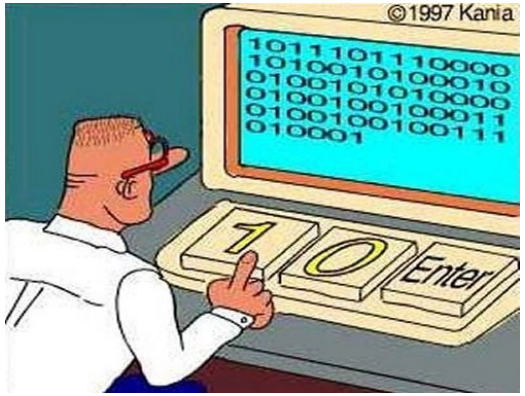
Versão 3

Versão 4

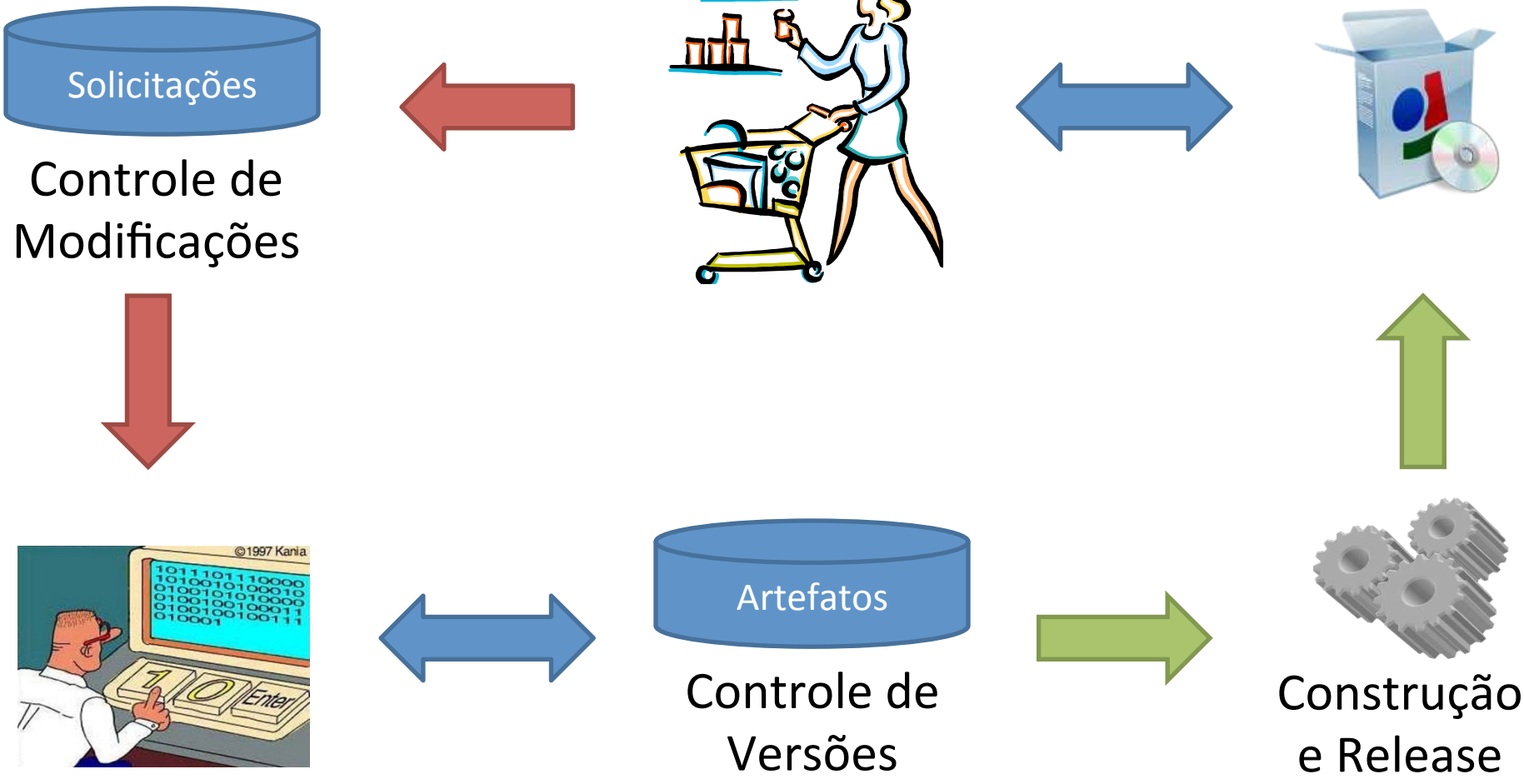
Versão 5

Controle de modificações

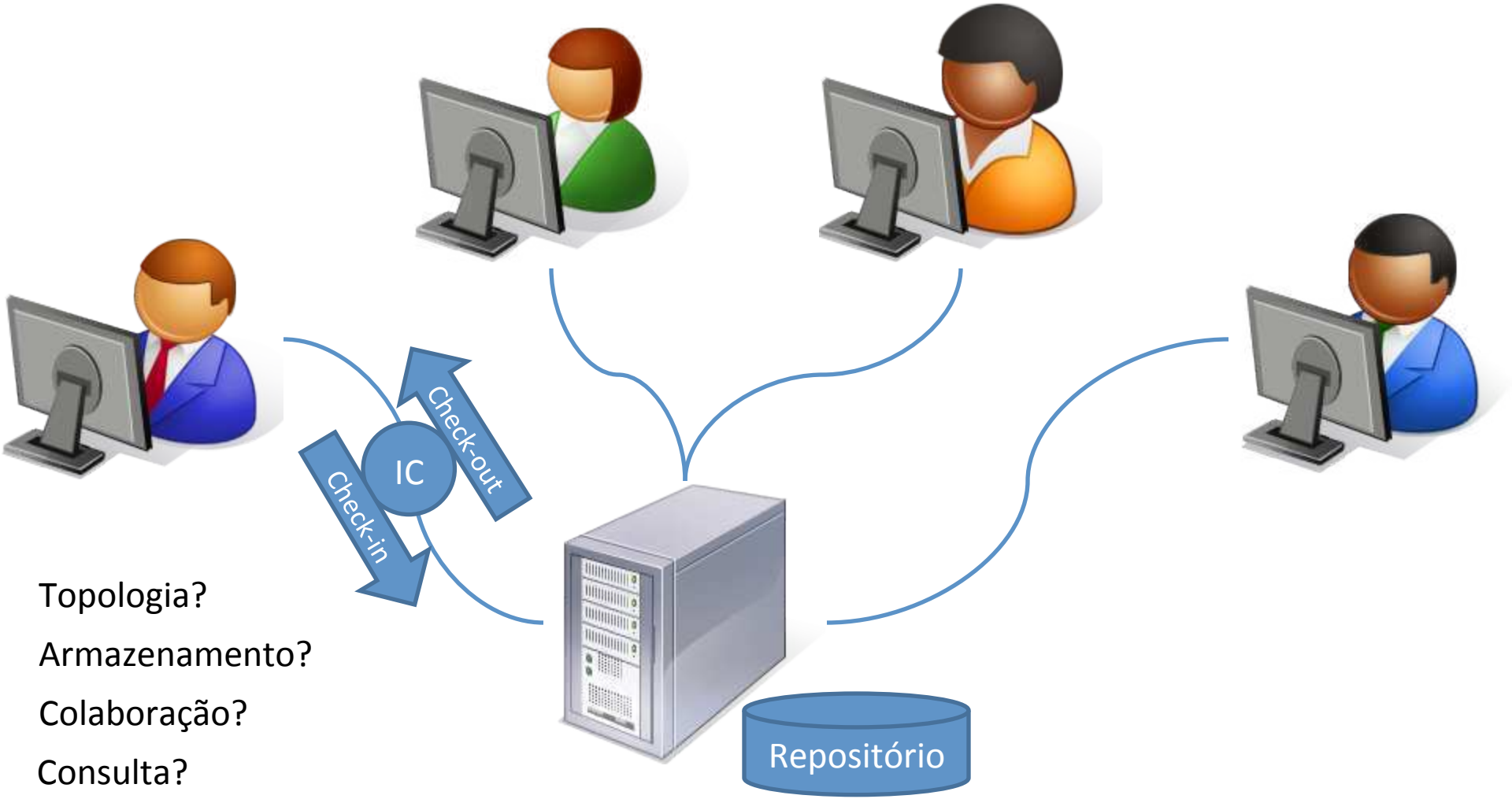
Controle de versões



Sistema de Gerência de Configuração



Controle de versões



Topologia?

Armazenamento?

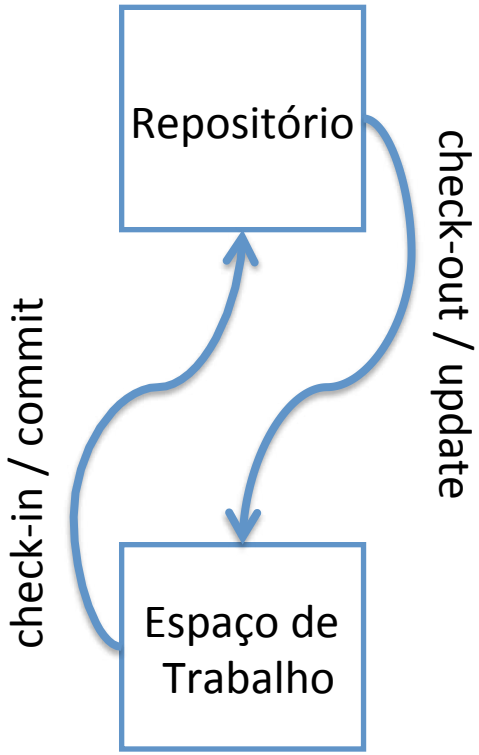
Colaboração?

Consulta?

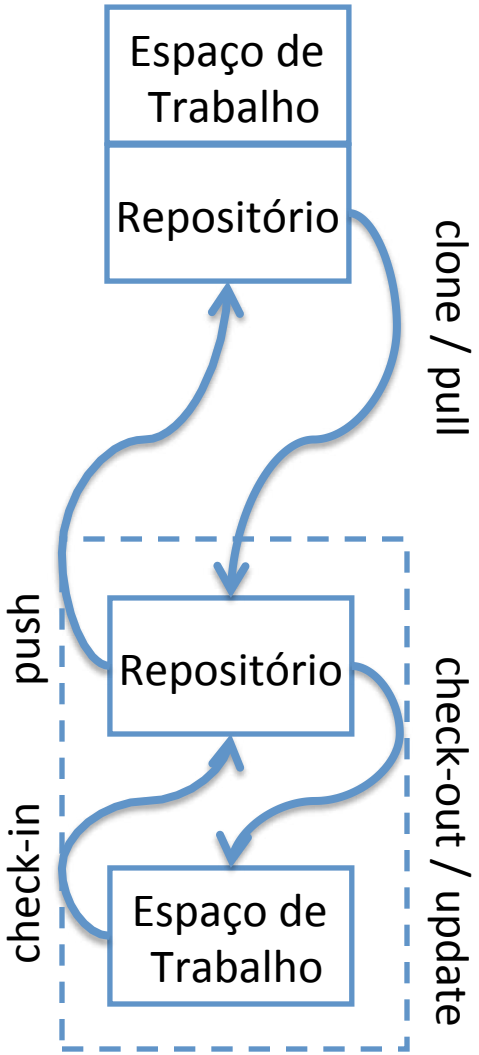
O Repositório

- 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

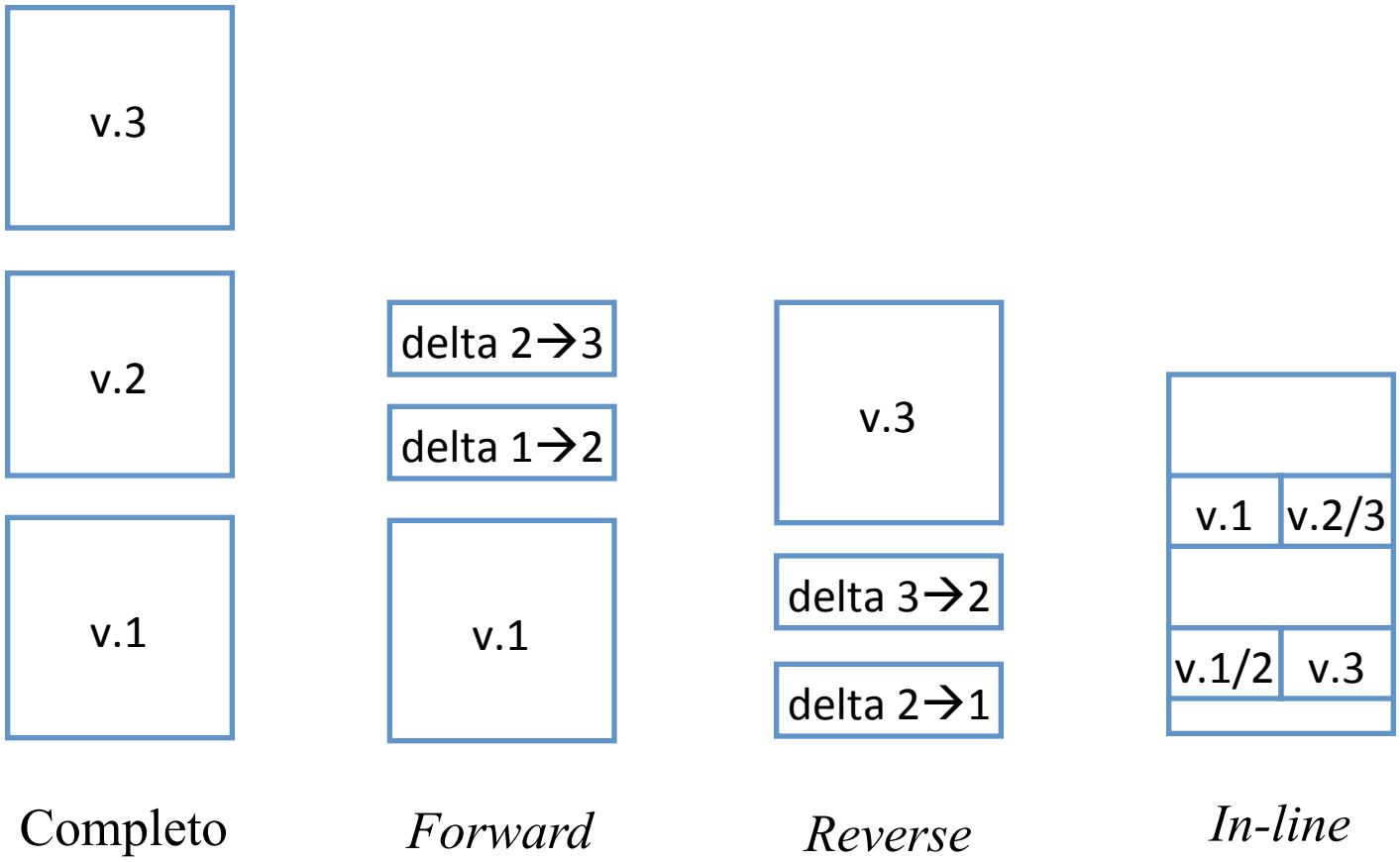


Distribuído

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* e *in-line*

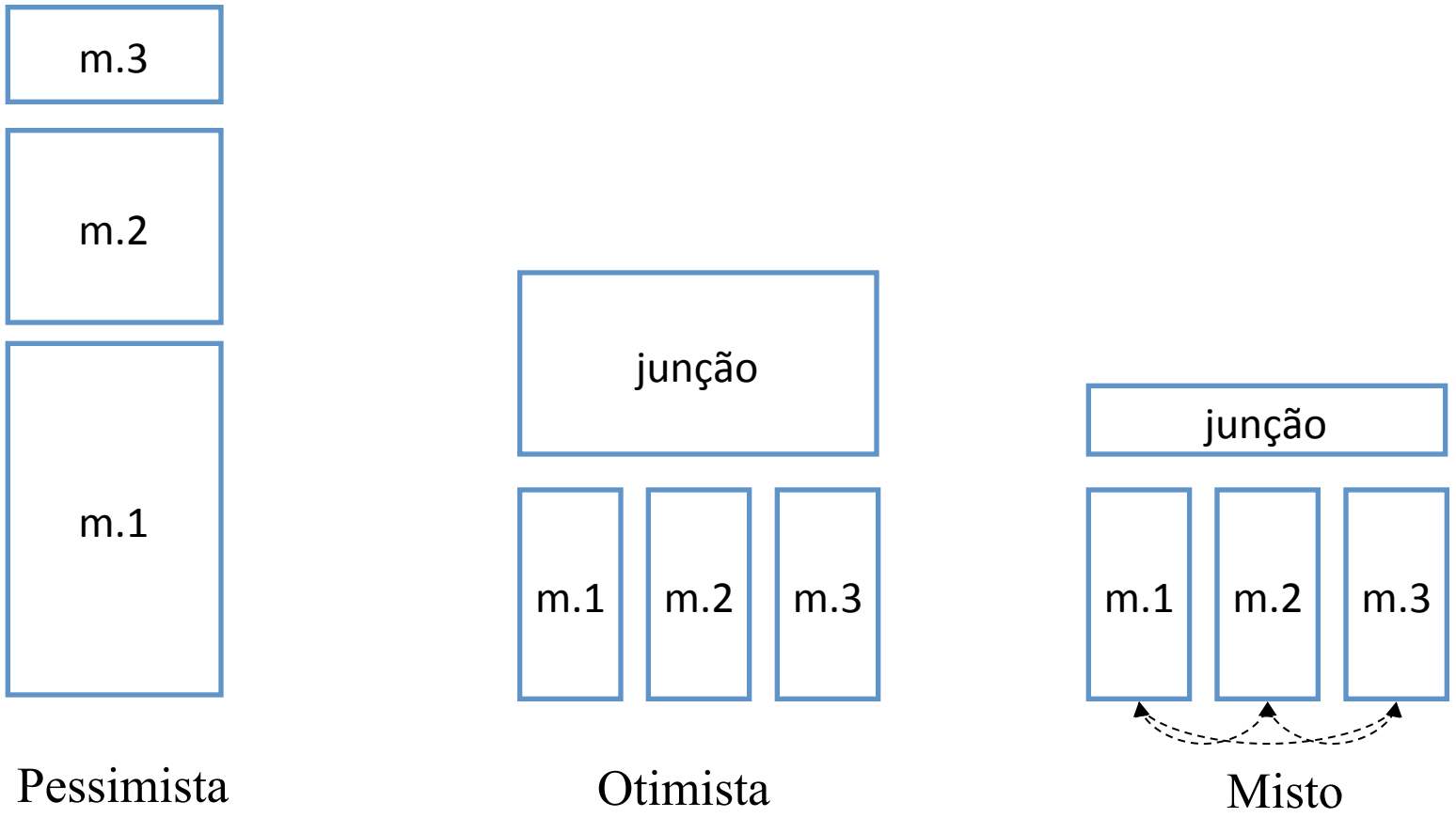
Armazenamento



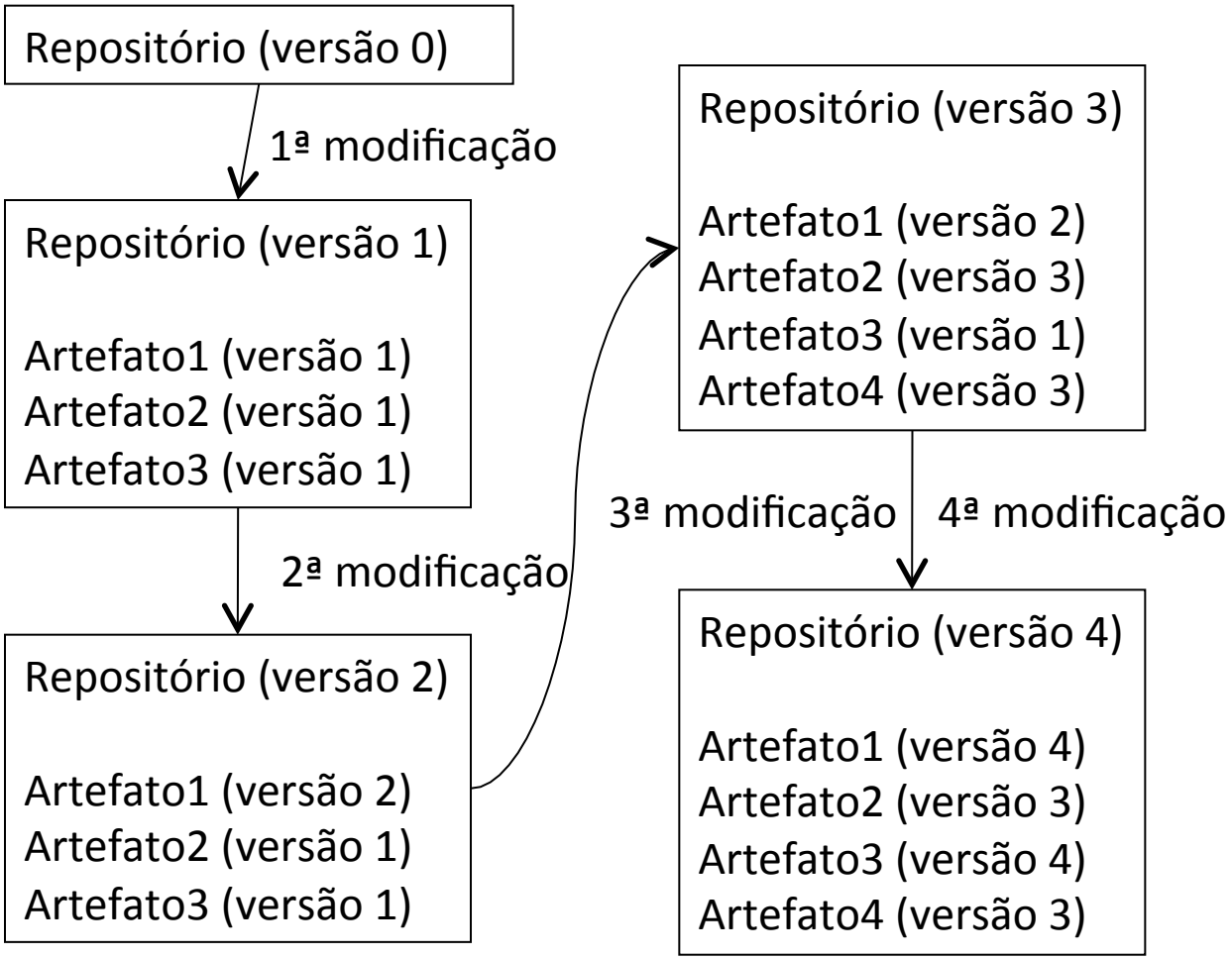
Colaboração

- 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

Colaboração



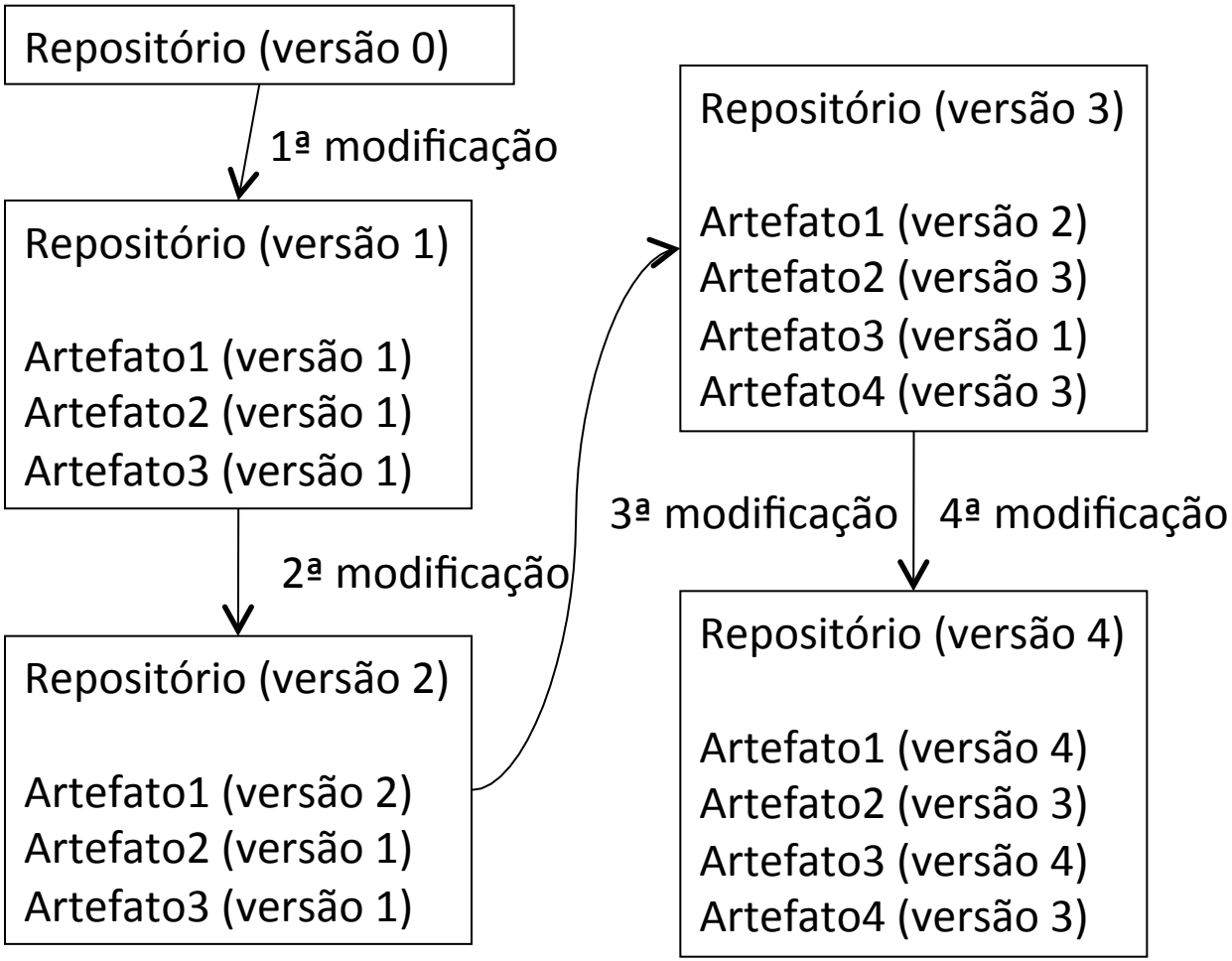
Consulta



Consulta por artefato

- Artefato1
  - Versão 1
  - Versão 2
  - Versão 4
- Artefato2
  - Versão 1
  - Versão 3
- Artefato3
  - Versão 1
  - Versão 4
- Artefato4
  - Versão 3

Consulta



Consulta por modificação

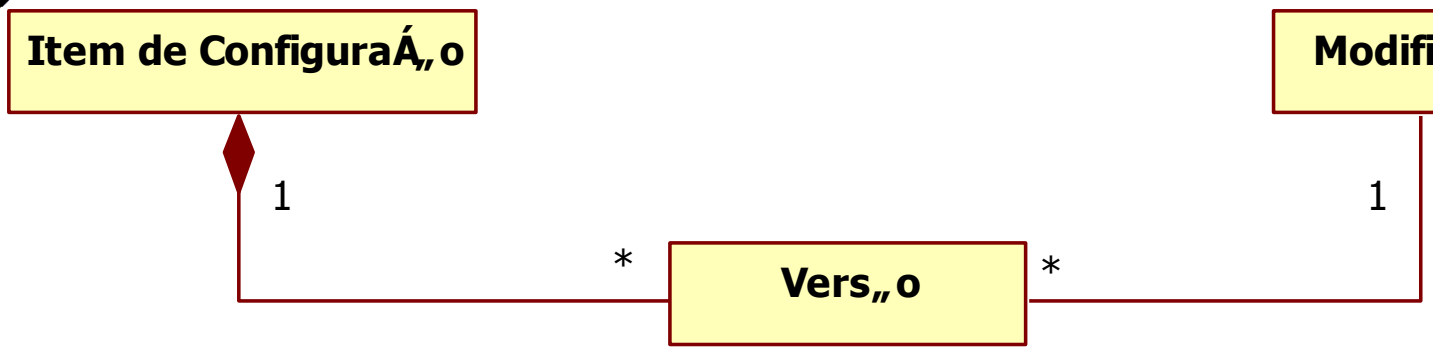
1ª modificação
- Artefato1 adicionado
- Artefato2 adicionado
- Artefato3 adicionado

2ª modificação
- Artefato1 modificado

3ª modificação
- Artefato2 modificado
- Artefato4 adicionado

4ª modificação
- Artefato1 modificado
- Artefato3 modificado

Consulta



Arquivo 5
- Versão 1
- Versão 2
- Versão 4

Modificação 4
- Arquivo 2
- Arquivo 5
- Arquivo 7

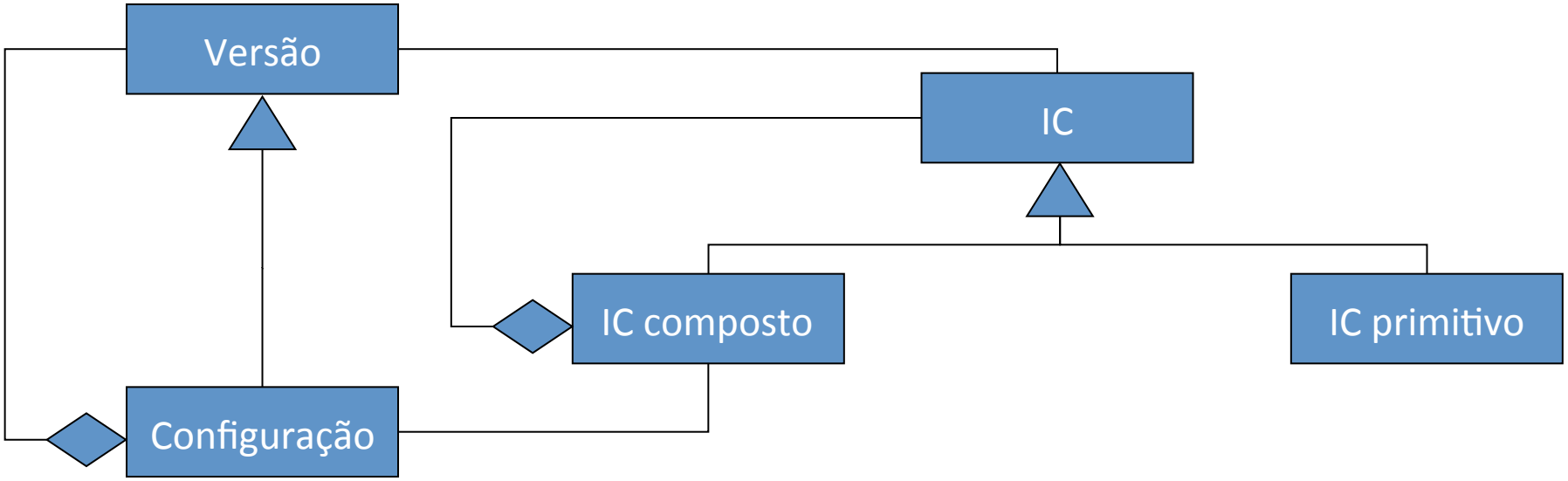
Espaço de trabalho

- 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*)

Configuração

- Um conjunto de versões de ICs, onde existe somente uma versão selecionada para cada IC do conjunto
- Uma configuração pode ser vista como um IC composto de outros ICs
- Exemplos
  - Configuração do sistema
  - Configuração do processo
  - Configuração do módulo X
  - Configuração dos requisitos do sistema
  - Configuração do código fonte

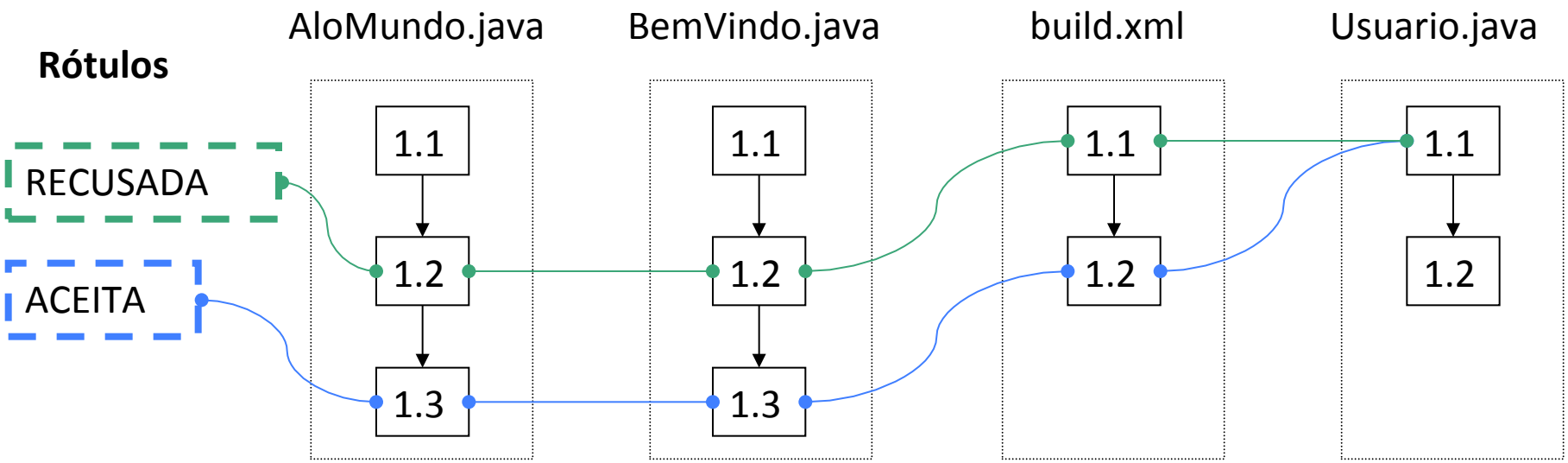
Configuração x versão



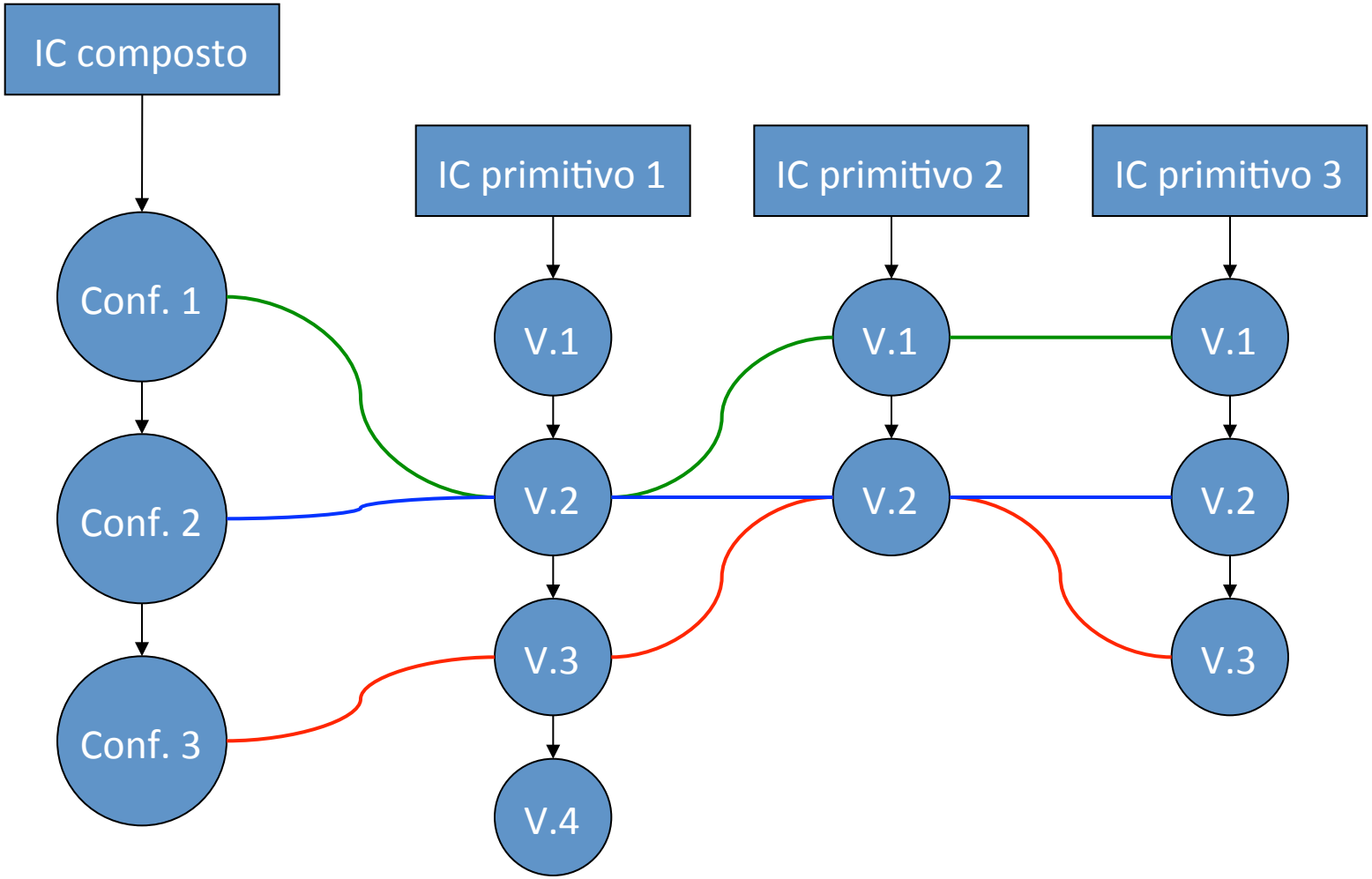
- Genericamente
  - O sistema S é composto pelos arquivos X, Y e Z
- Concretamente
  - A configuração 5 do sistema S é composta pela versão 2 do arquivo X, versão 4 do arquivo Y e versão 6 do arquivo Z

Rótulo (*label*)

- 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: etiqueta (*tag*)



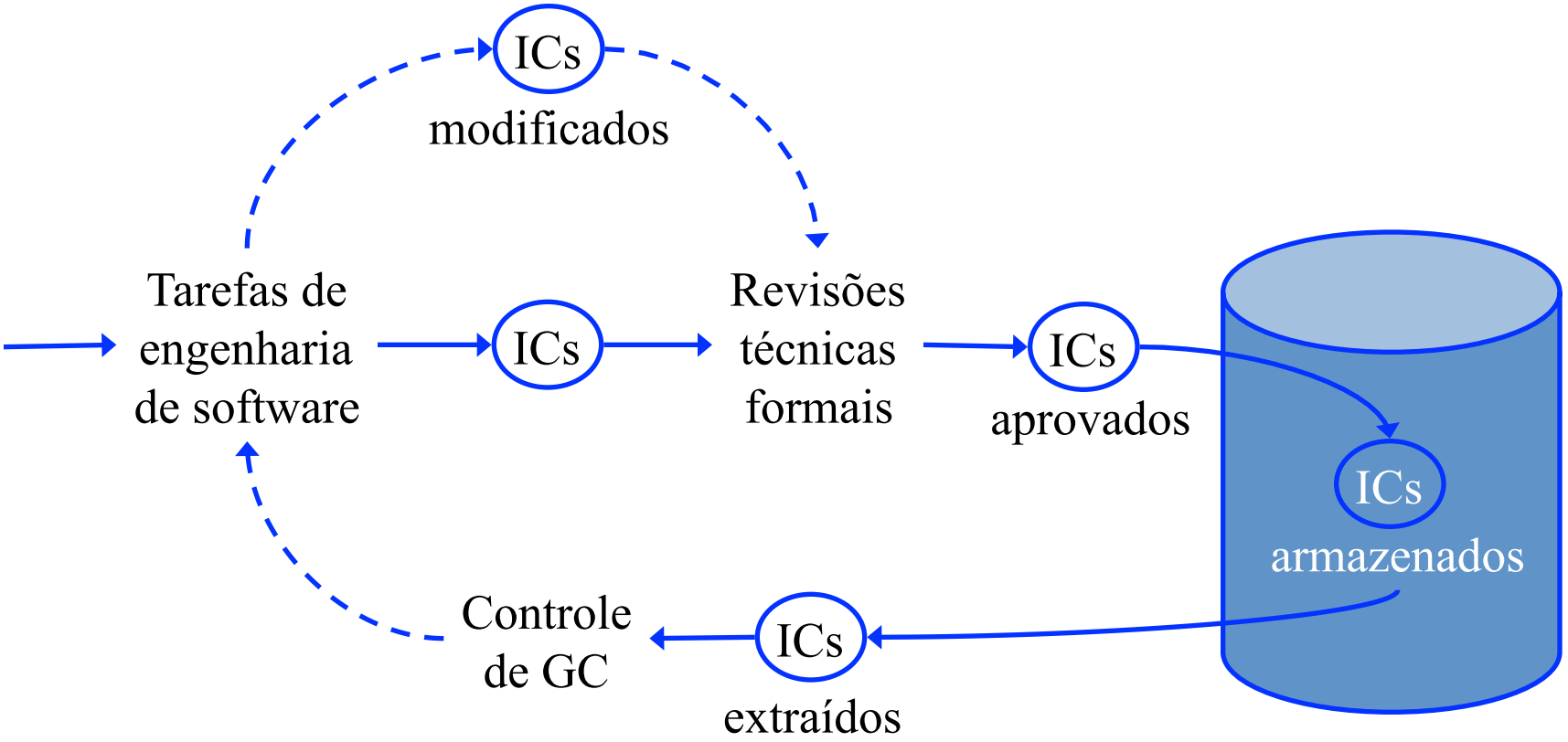
Configuração x versão



Baseline

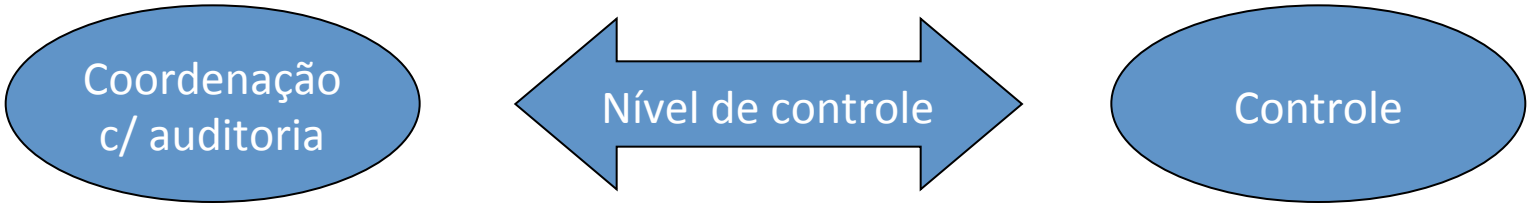
- Configuração revisada e aprovada que serve como base para uma próxima etapa de desenvolvimento e que somente pode ser modificada via processo formal de GCS
- São estabelecidas ao final de cada fase de desenvolvimento: análise (*functional*), projeto (*allocated*) e implementação (*product*)
- Momento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

Baseline



[Pressman, 1997] Processo de atualização de configurações de referência

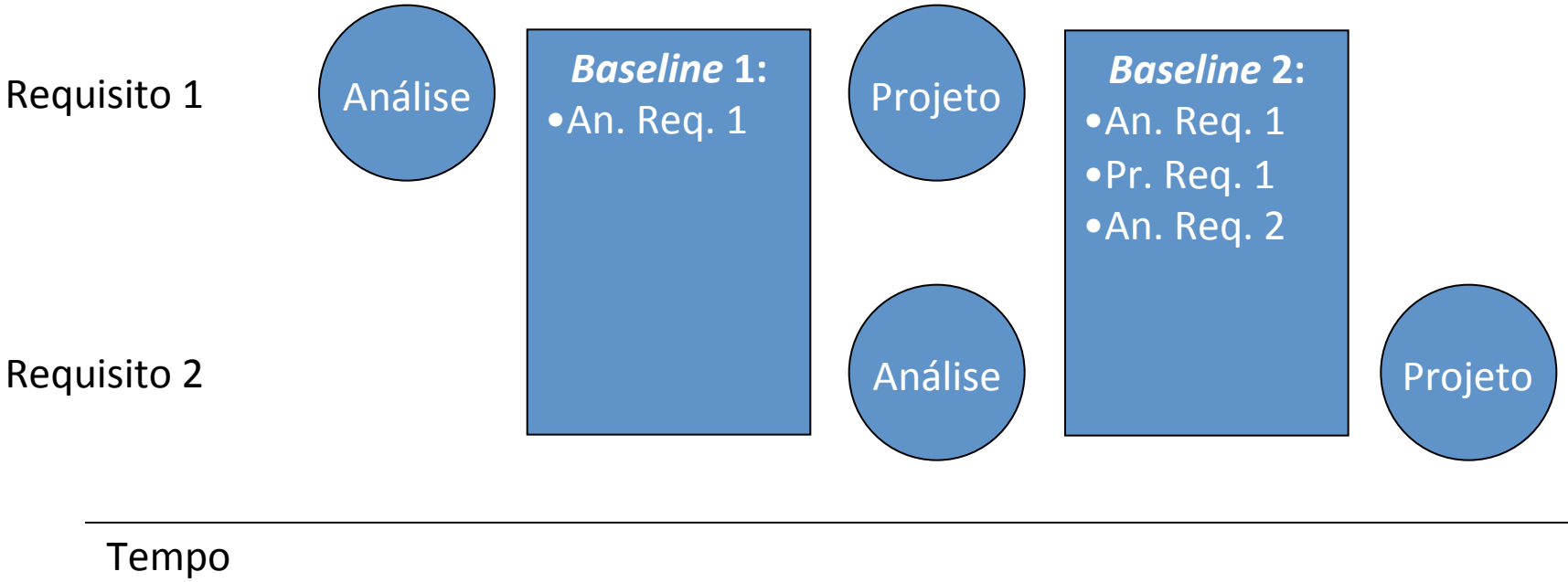
Baseline (níveis de controle)



Pré *baseline*:
- Informal
- Sem requisição
- Sem aprovação
- Sem verificação
- Ágil
- Ad-hoc

Pós *baseline*:
- Formal
- Com requisição
- Com aprovação
- Com verificação
- Burocrático
- Planejado

Baseline (níveis de controle)



Req.
1
2

Análise	Projeto
Inform.	-
-	-

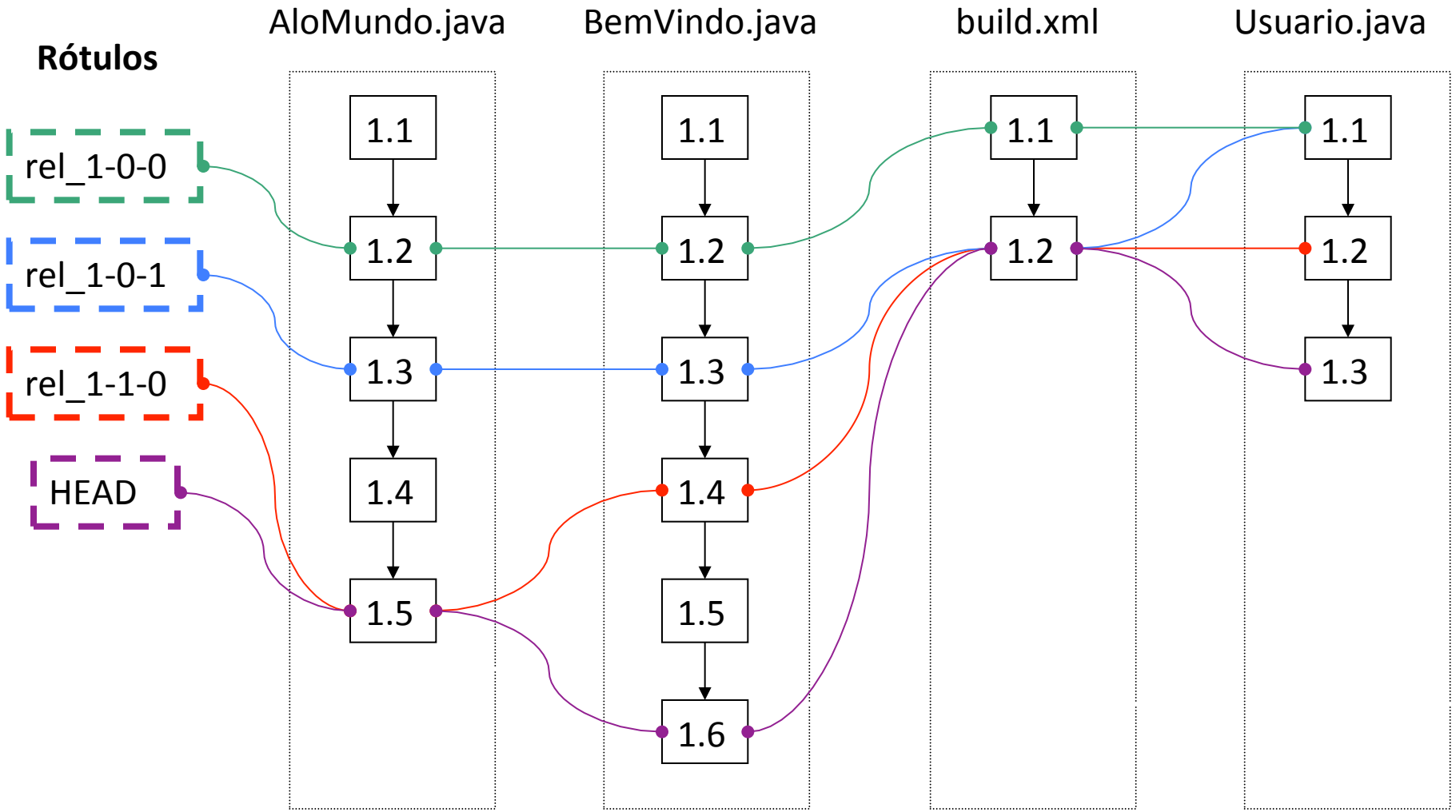
Análise	Projeto
Formal	Inform.
Inform.	-

Análise	Projeto
Formal	Formal
Formal	Inform.

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

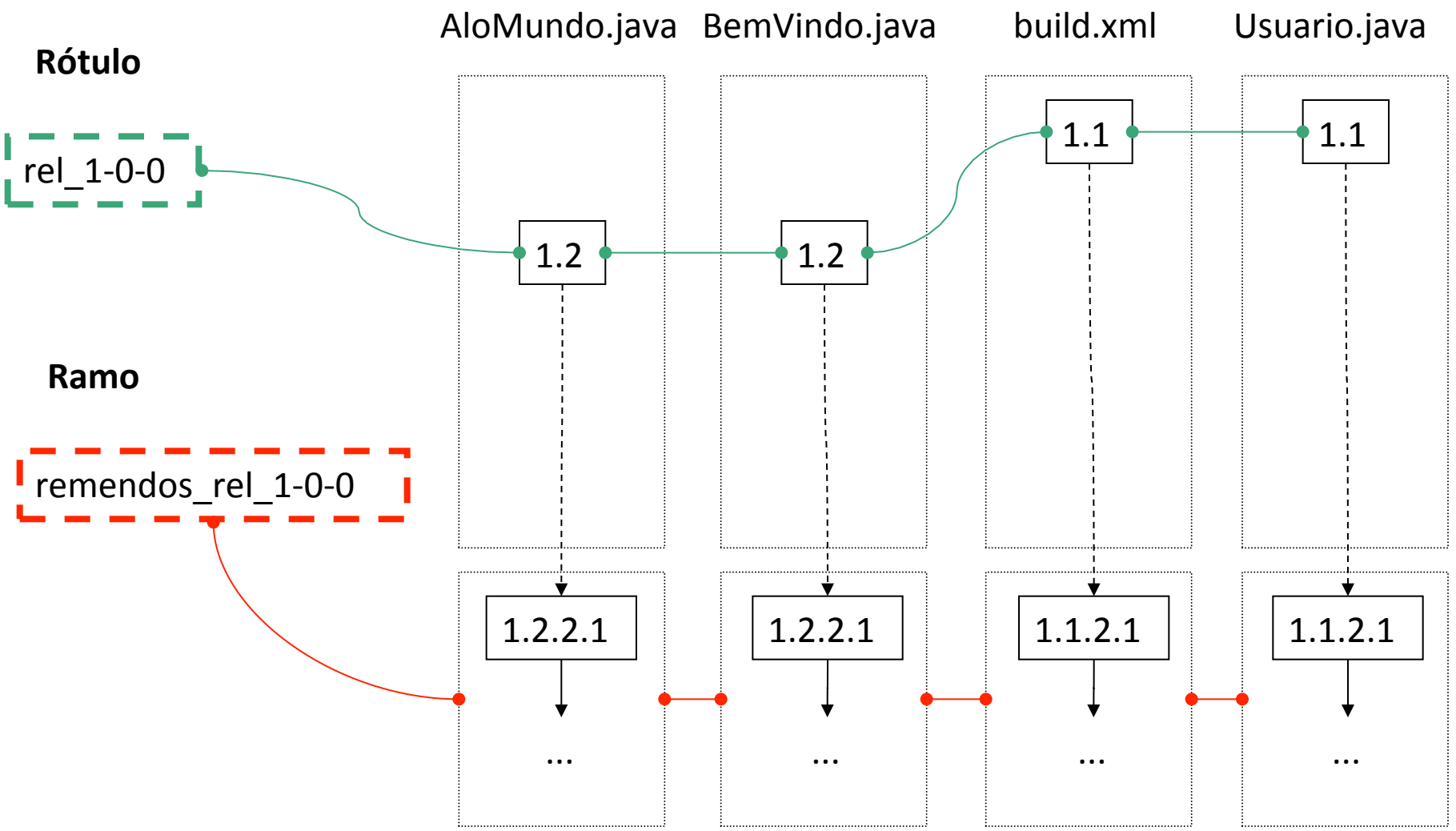
Exemplo de liberações



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)

Exemplo de ramo

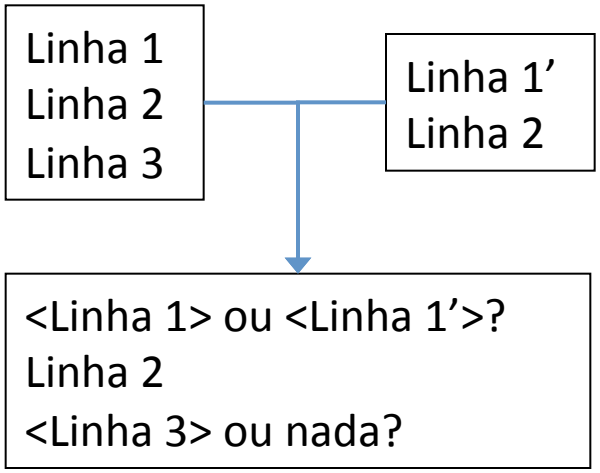


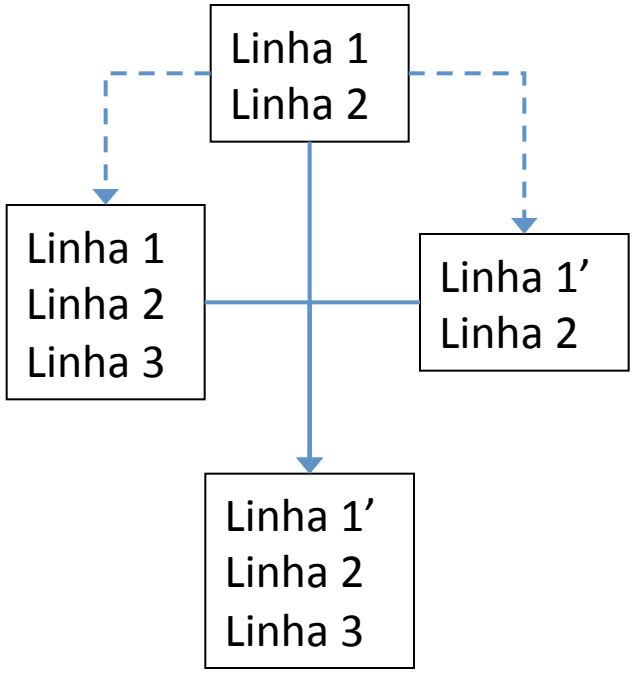
Junção

- Processo de migração de
  - Espaços de trabalho
  - Ramos
- É necessário inclusive em desenvolvimento seqüencial (*check-out* reservado) quando são utilizados ramos
- Algoritmos automáticos se dividem em duas categorias
  - Genéricos (servem para qualquer linguagem)
  - Específicos (levam em conta a estrutura semântica da linguagem)

Junção

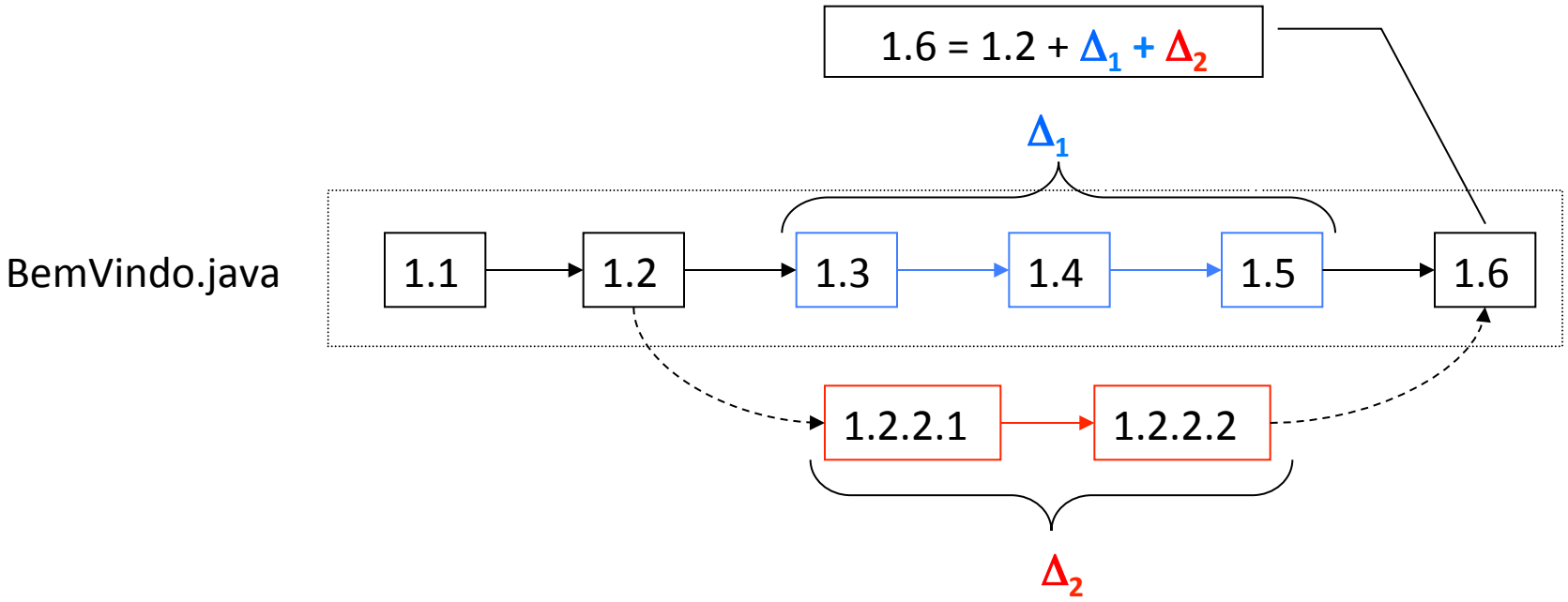
- As ferramentas de GCS usualmente utilizam algoritmos genéricos
  - *2-way merge*
  - *3-way merge*





Exemplo de junção

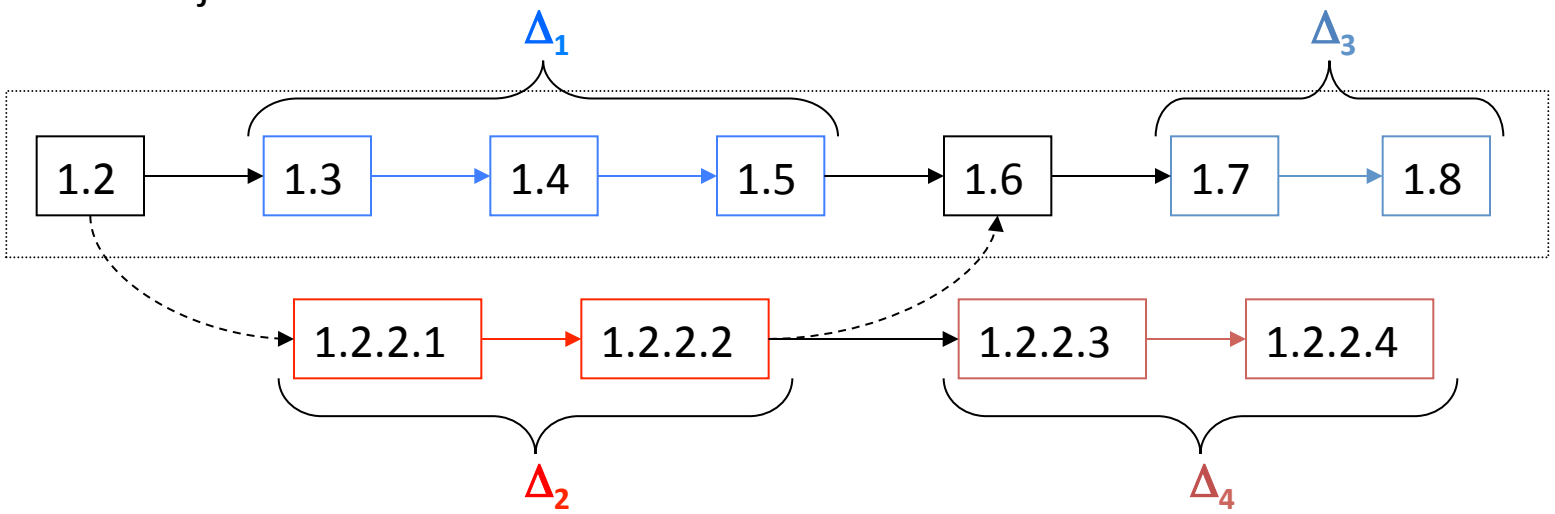
- A junção é efetuada para cada artefato do ramo
- São levadas em consideração todas as modificações desde o ancestral em comum



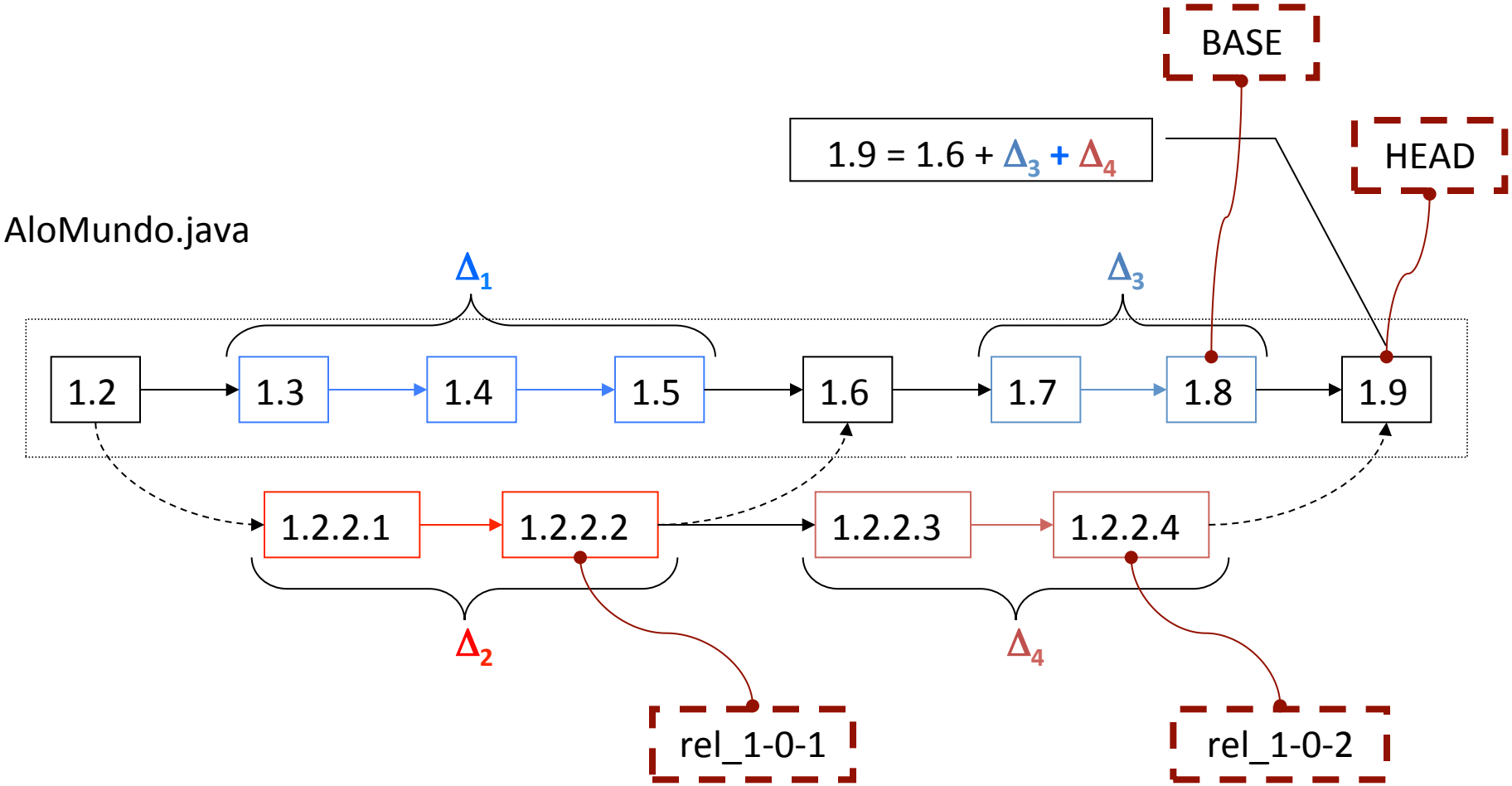
Exemplo de junção (incremental)

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

AloMundo.java



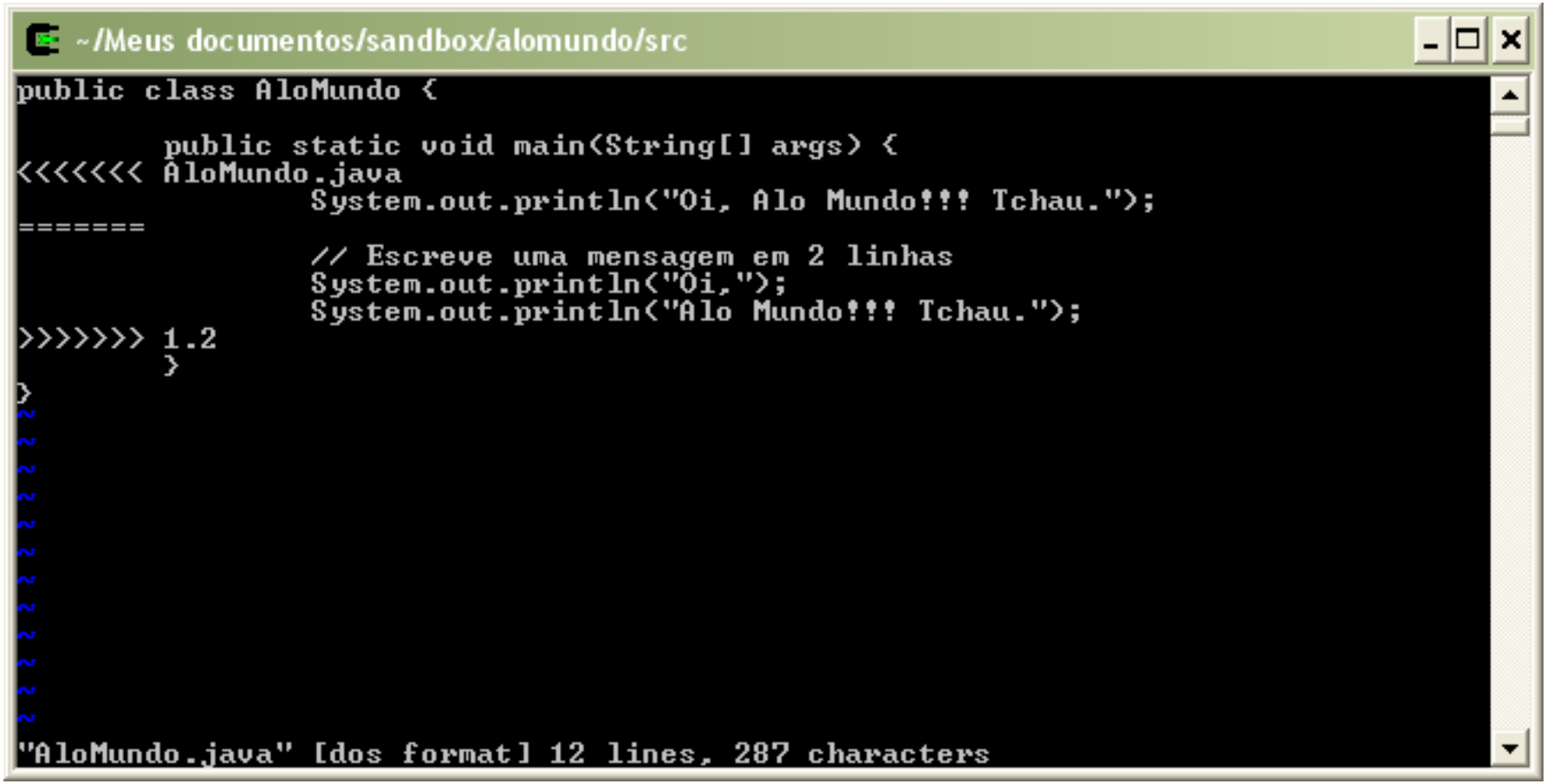
Exemplo de junção (incremental)



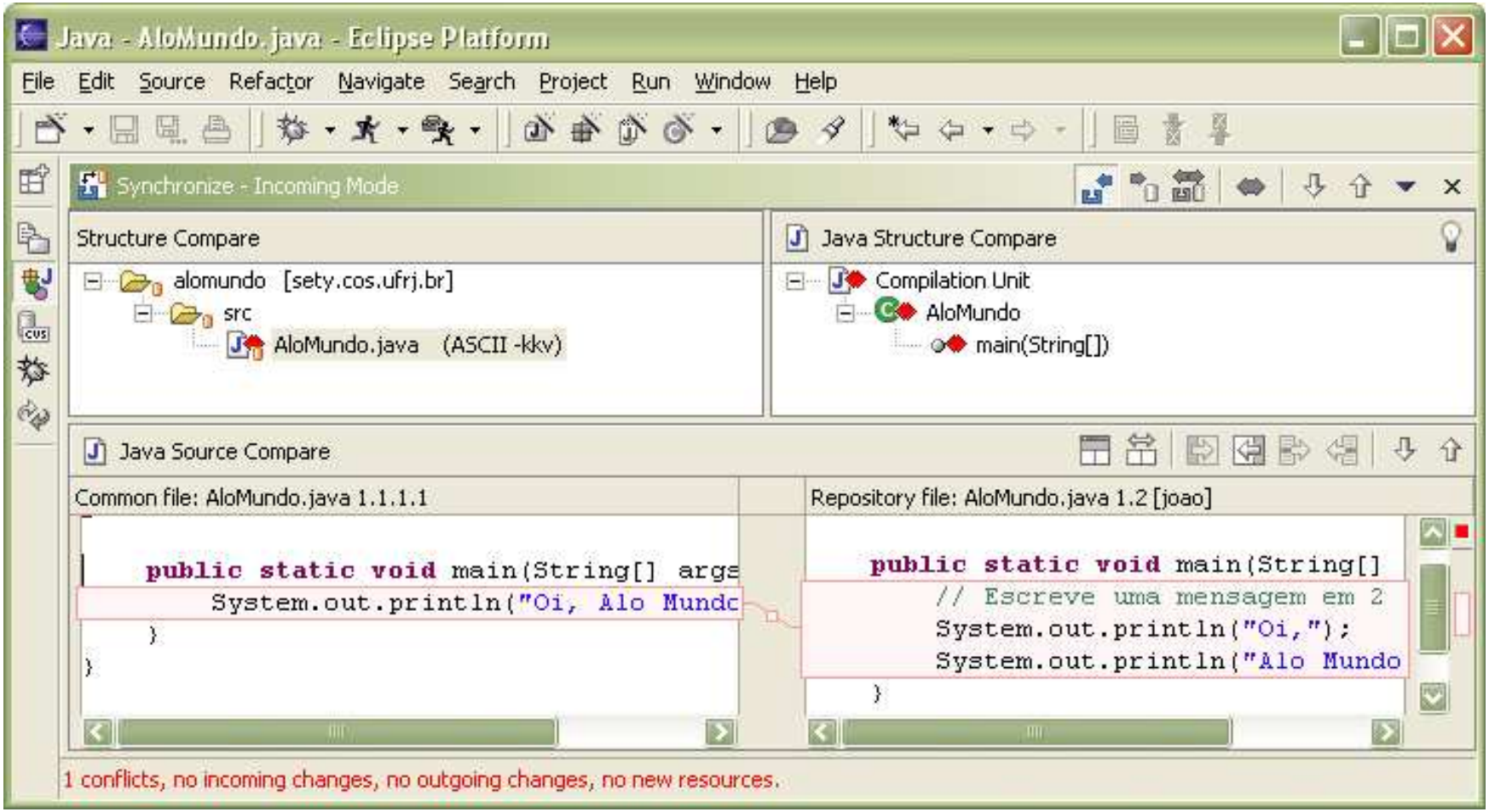
Conflitos

- Situação onde não é possível executar a junção de forma automática
- Tipos
  - Físico (linha do arquivo)
  - Lógico (sintaxe do arquivo)
  - Semântico (conteúdo do arquivo)
- O suporte atual concentra no nível físico!
- Exemplos de conflitos físicos
  - Alterações em paralelo de uma mesma linha
  - Remoção e alteração em paralelo de uma mesma linha
  - Adições de linhas em paralelo na mesma região do arquivo

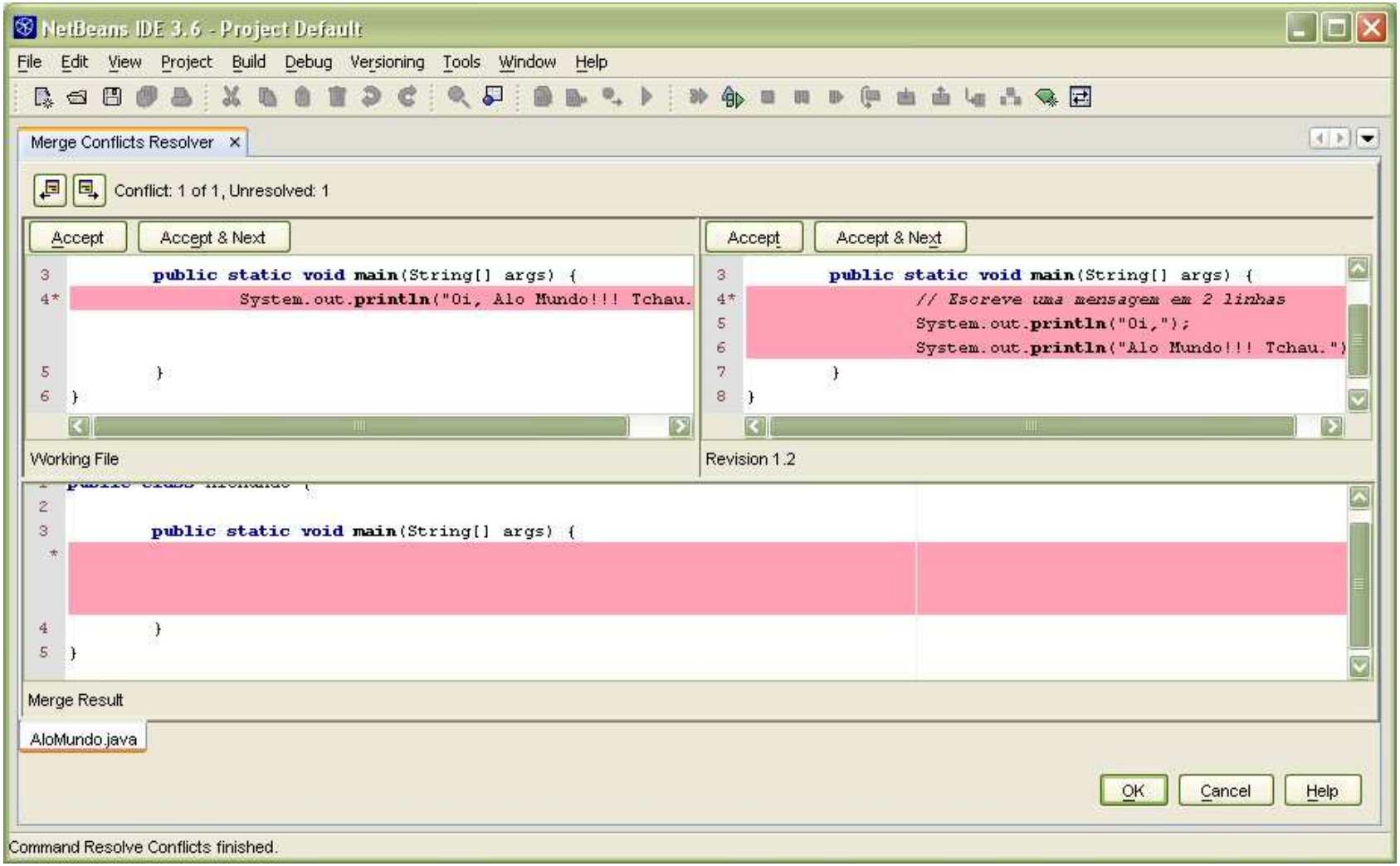
Conflitos no vi



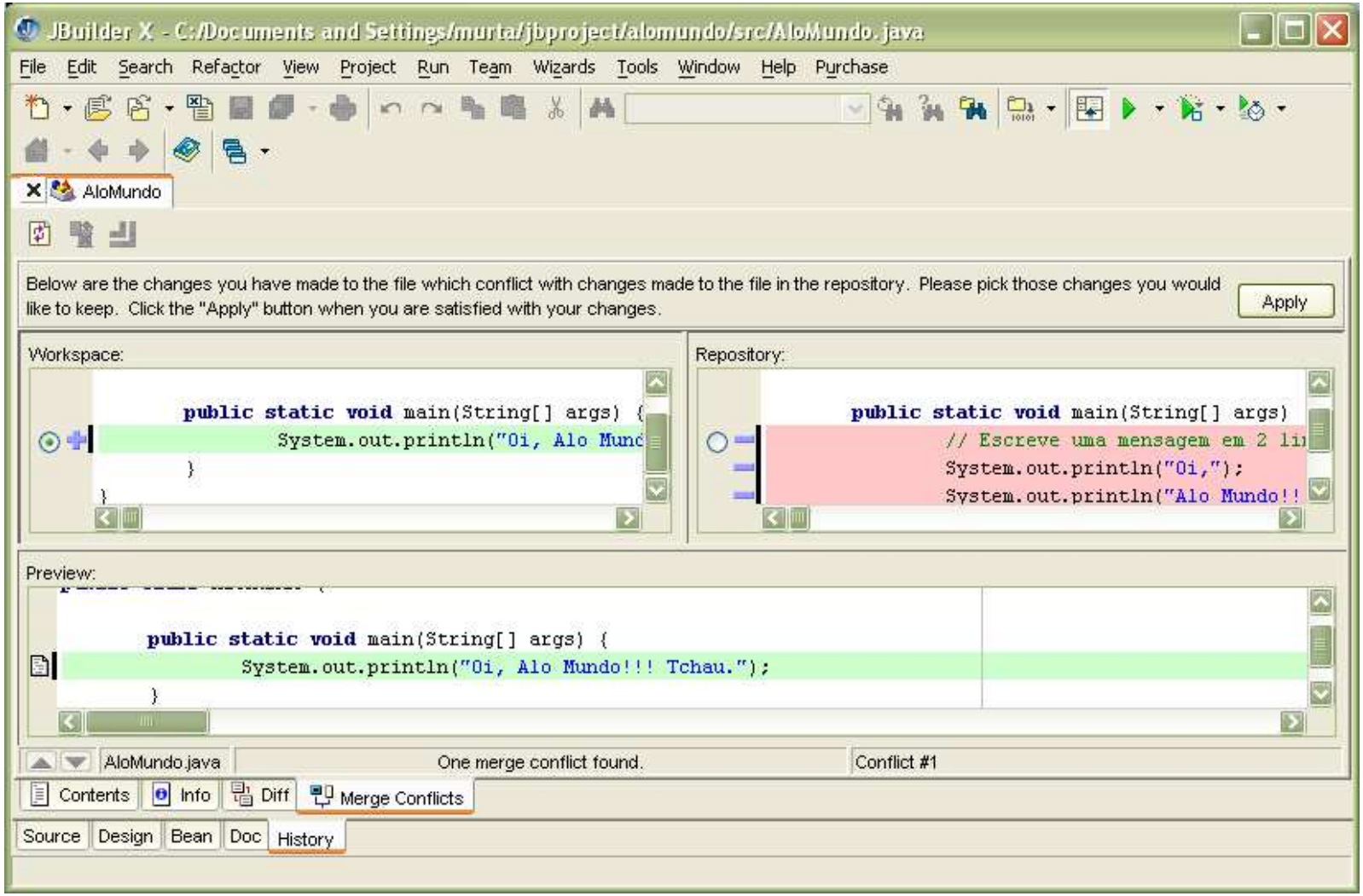
Conflitos no Eclipse



Conflitos no NetBeans



Conflitos no JBuilder



Instituto de Computação

Gerência de Configuração: Terminologia

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br