



Departamento de Engenharia Elétrica
Escola Politécnica – Universidade de São Paulo
Área de Concentração: Engenharia da Computação

Evolução de Bancos de Dados Utilizando Planejamento Automático

Márcia Beatriz Pereira Domingues
Jorge Rady de Almeida Júnior



Agenda

- ✓ Introdução a Evolução de Banco de Dados
- ✓ Refatoração de Banco de dados
- ✓ Proposta do Artigo
- ✓ Aplicação da proposta
- ✓ Conclusões

Refatoração

A técnica utilizada para alterar código e banco de dados é chamada de **refatoração**.

- Segundo Martin Flower:

Uma refatoração de código é uma pequena alteração que melhora o *design*, mas não altera a semântica do código, nem adiciona novas funcionalidades (FOWLER et al., 1999).

- Scott Ambler ampliou a definição para banco de dados:

“Uma pequena mudança no esquema que melhora o projeto, não adiciona funcionalidades e não altera a semântica informacional ou comportamental do esquema do banco de dados (AMBLER; SADALAGE, 2006).”

Refatoração

Amblor desenvolveu um Catálogo de refatorações

Classificadas como:

- ✓ **Estruturais**
Renomear ou remover colunas;
- ✓ **Qualidade de Dados**
Valor padrão para uma coluna;
- ✓ **Integridade Referencial**
Adicionar ou remover restrição de chave estrangeira
- ✓ **Arquitetural**
Adicionar índices

Processo

Grandes alterações necessitam de um conjunto de refatorações

Necessidades:

- ✓ Executar todas as refatorações em um único processo.
- ✓ Automatizar o processo de refatorações

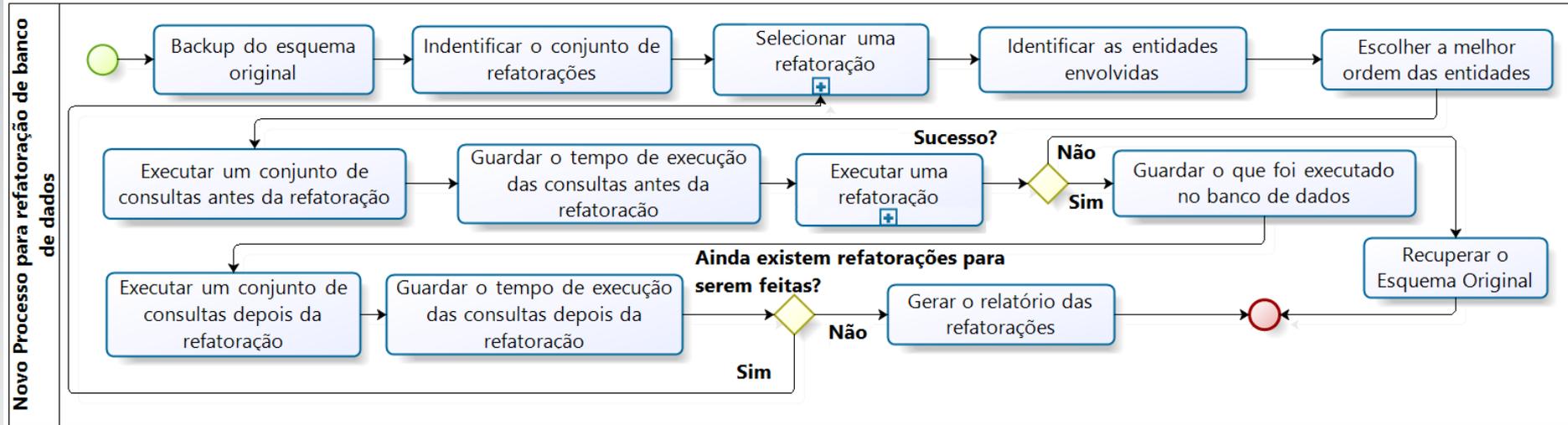
Processos de refatoração de banco de dados disponíveis na literatura:

Limitações:

- ✓ Realizar diversas refatorações em um mesmo processo;
- ✓ Salvar o histórico dos passos dessas refatorações;
- ✓ Recuperar o modelo inicial caso ocorra um problema e
- ✓ Gerar relatórios.

Processo para refatoração de Banco de Dados

Tese de Doutorado – Domingues (2014)



- Permite a execução de várias refatorações em um mesmo processo.
- Inclui tarefas importantes para consistência do banco de dados após a refatoração.
- Previne erros.
- Foi escrito utilizando a notação BPMN.

Objetivo Artigo WTA 2015

Utilizar técnicas de planejamento automático para modelagem do processo de refatoração de banco de dados.

Conceitos Relacionados:

Planejamento Automático (IA)

Redes de Petri (Modelagem, Análise e Rastreamento)

UML (Análise de diagramas de estado)

Engenharia de Software (Projeto de Sistemas)

Banco de Dados (Evolução de esquemas)

Planejamento Automático

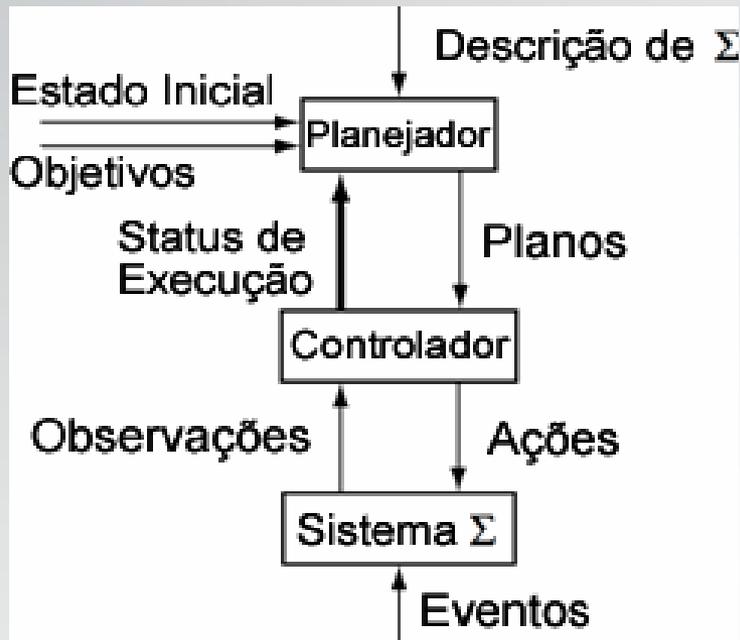
Planejamento: Processo de deliberação que escolhe e organiza ações para antecipar seus resultados esperados. Esta deliberação tem por objetivo atingir, da melhor forma possível, alguns objetivos conhecidos.

Planejamento automatizado: Área de Inteligência Artificial (IA) que estuda, computacionalmente, este processo de deliberação. (GHALLAB, NAU e TRAVERSO, 2004).

Plano:

- Sequência de ações para se atingir um objetivo.
- Estrutura que segue ações apropriadas.
- O objetivo pode ser especificado como um estado objetivo (*goal state*) S_g , ou um conjunto de estados objetivos.
- O objetivo é alcançado por qualquer sequência de transições de estado que termina em um dos estados objetivos.

Planejamento Automático



Planejador (*Planner*) recebe como entrada a descrição de um sistema Σ , um estado inicial e objetivos que sintetizam o plano para o controlador executar e atingir o objetivo;

Controlador (*Controller*) recebe o estado atual (s) do sistema e providencia como saída uma ação de acordo com o plano escolhido pelo planejador;

Sistema estado-transição Σ (*System Σ)* evolui de acordo com as ações e eventos que recebe.

O sistema estado-transição pode ser representado por um **grafo direcionado** onde os nós são estados e os arcos são ações ou eventos) (VAQUERO, 2007).

$$s \xrightarrow{u} s'$$

Estado transição de s para s'

Planejamento Automático

No modelo clássico de planejamento o sistema estado-transição pode ser definido formalmente como uma 4-tupla $\Sigma = (S, A, E, \gamma)$, onde:

$S = \{s_1, s_2, \dots\}$ é o conjunto finito ou recursivo dos estados;

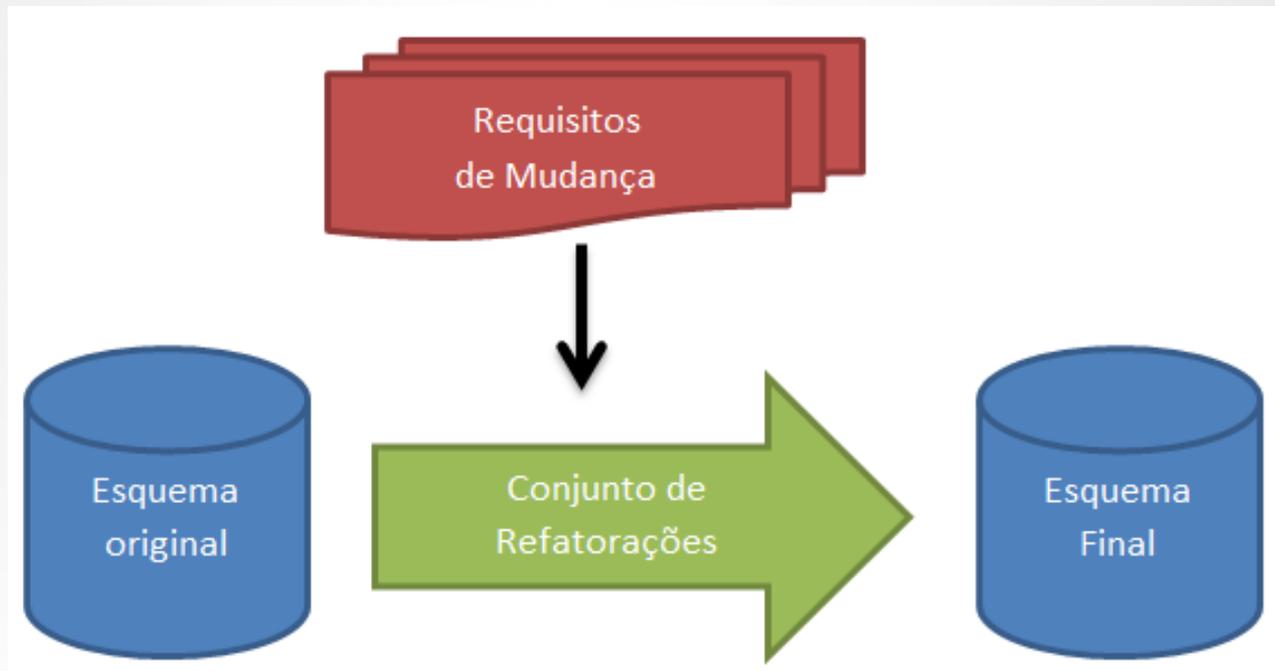
$A = \{a_1, a_2, \dots\}$ é o conjunto finito ou recursivo das ações;

$E = \{e_1, e_2, \dots\}$ é o conjunto finito ou recursivo dos eventos exógenos (não controlados pelo agente);

$\gamma: S \times (A \cup E)$ é a função de estado-transição.

Contexto

- ✓ Esquema inicial;
- ✓ Requisito de mudanças;
- ✓ Conjunto de refatorações;
- ✓ Esquema Final.



Proposta

- **Objetivo:**

Planejar uma grande alteração no banco de dados utilizando técnicas de Planejamento Automático.

- **Justificativa:**

Uma grande alteração pode ser definida como um conjunto de refatorações que atendam aos requisitos do usuário.

Esse conjunto de refatorações:

- Necessita de um planejamento de execução;
- A ordem pode influenciar na duração de uma grande alteração no banco de dados.

Formalização

Considerando:

- **Conjunto de refatorações** = conjunto de ações a serem executadas no esquema inicial do banco de dados;
- A cada refatoração (ação) executada, o esquema de dados atinge um estado diferente.

Tem-se, o sistema proposto: $\Sigma = (S, A, \gamma)$, onde:

$S = \{s_1, s_2, \dots\}$ é o conjunto finito de estados do esquema do banco de dados;

$A = \{a_1, a_2, \dots\}$ é o conjunto finito de ações (refatorações e transformações);

$\gamma: S \times A$ é a função de estado-transição.

Formalização

Dado $\Sigma = (S, A, \gamma)$ um estado inicial s_0 e um conjunto de estados objetivo S_g :

Deve-se encontrar:

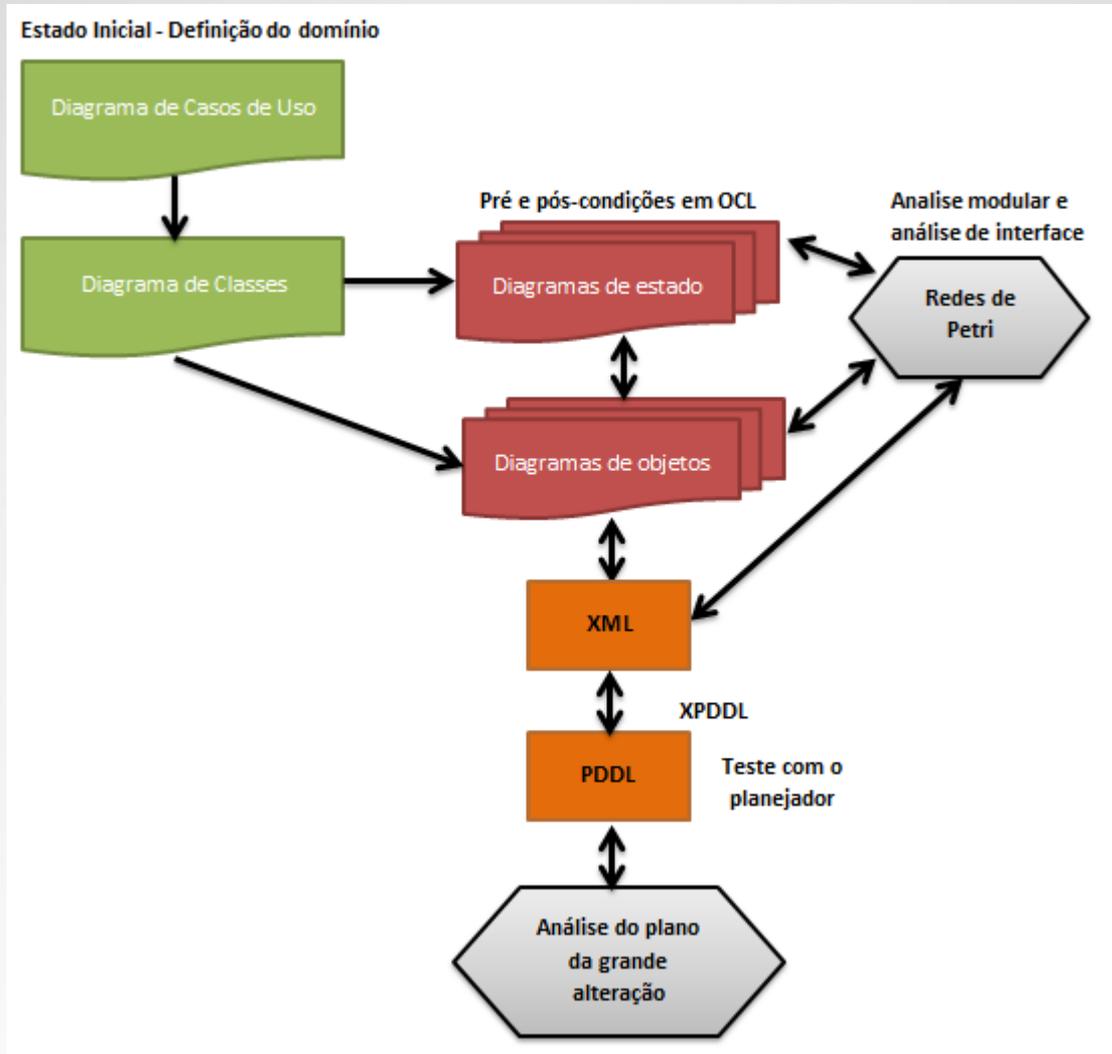
- uma sequência de ações (a_1, a_2, \dots, a_k) que corresponda a uma sequência de estados-transições (s_1, s_2, \dots, s_k) tal que:

$$s_1 \in \gamma(s_0, a_1),$$

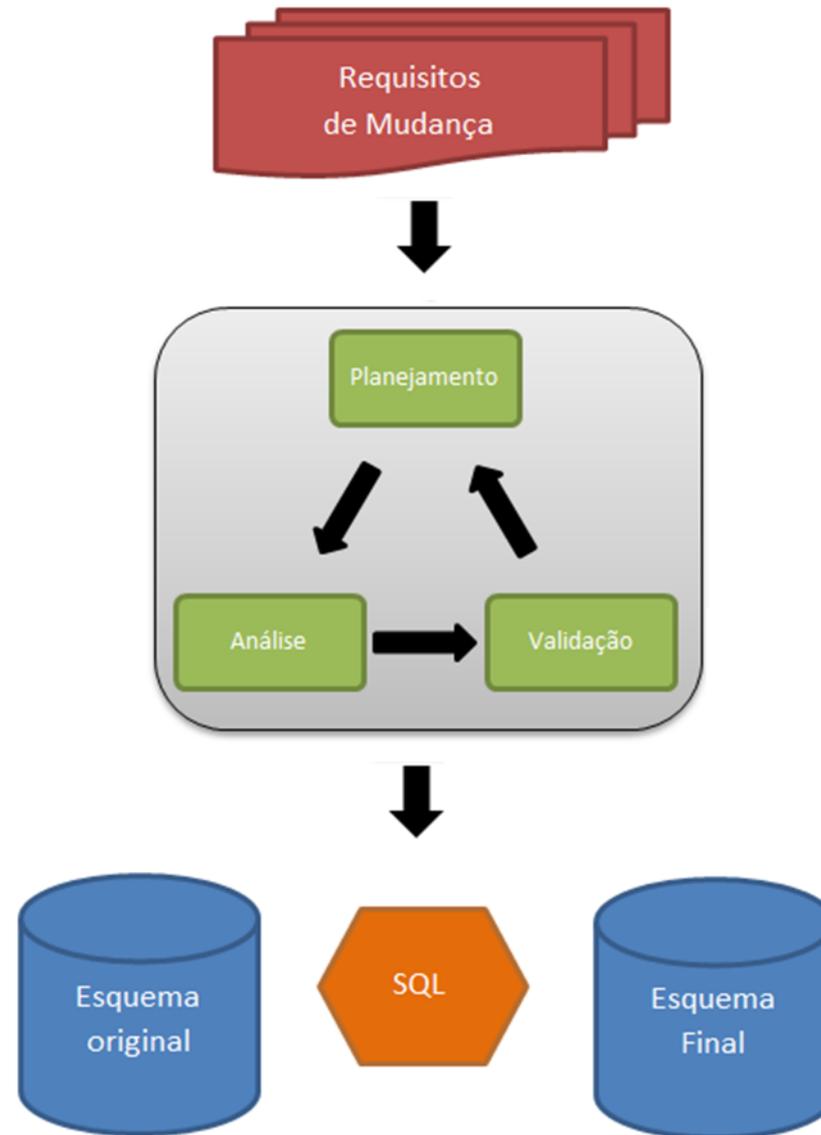
$$s_2 \in \gamma(s_1, a_2), \dots,$$

$$s_k \in \gamma(s_{k-1}, a_k), \text{ onde } s_k \text{ pertença à } S_g.$$

Modelo Proposto Para o Planejamento

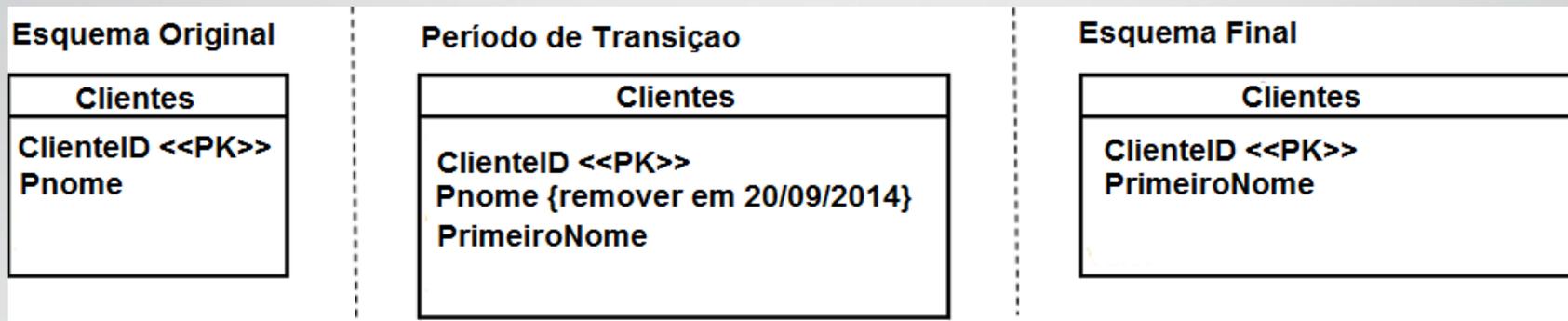


Modelo Proposto Para o Planejamento

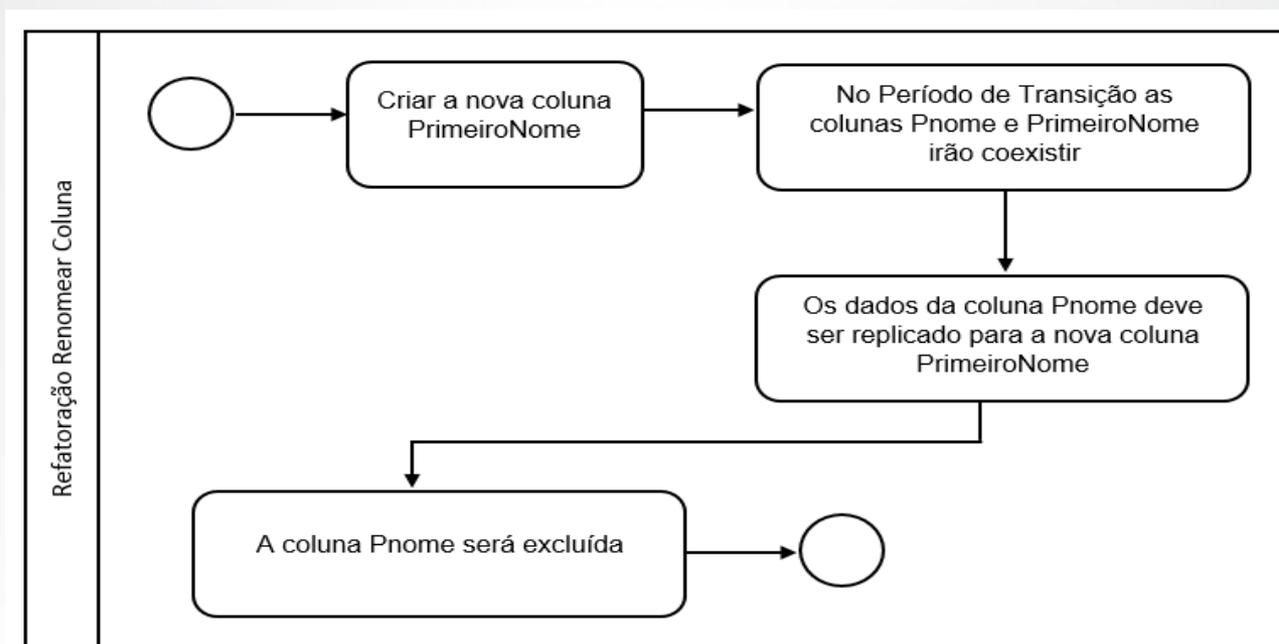


Aplicação

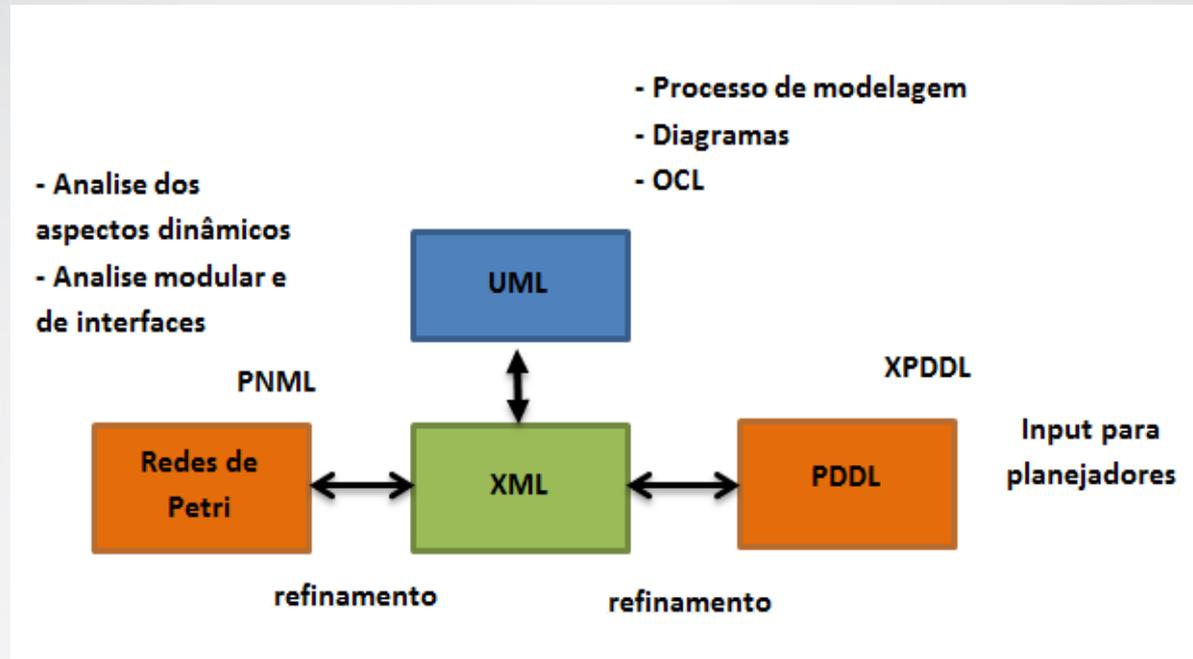
Requisito: Renomear a coluna Pnome da Tabela Cliente para PrimeiroNome



Passos:



ItSIMPLE

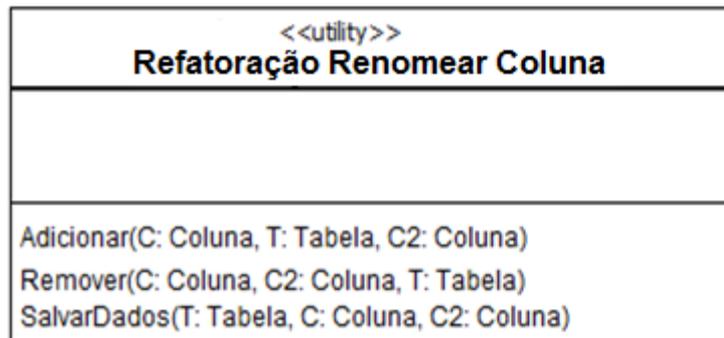
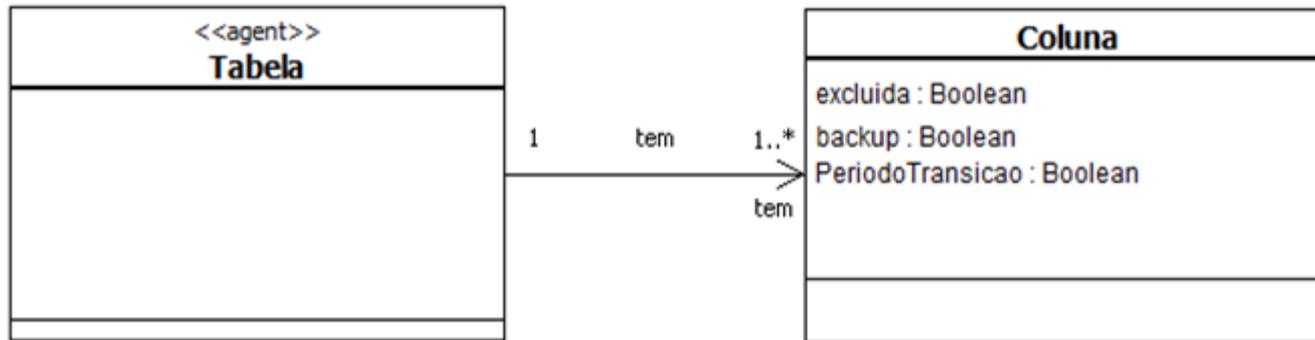


Arquitetura das linguagens integradas e Modelo de domínio utilizados no itSIMPLE

VAQUERO, T. S. ITSIMPLE: ambiente integrado de modelagem e análise de domínios de planejamento automático. Escola Politécnica, Universidade de São Paulo, São Paulo. 2007.

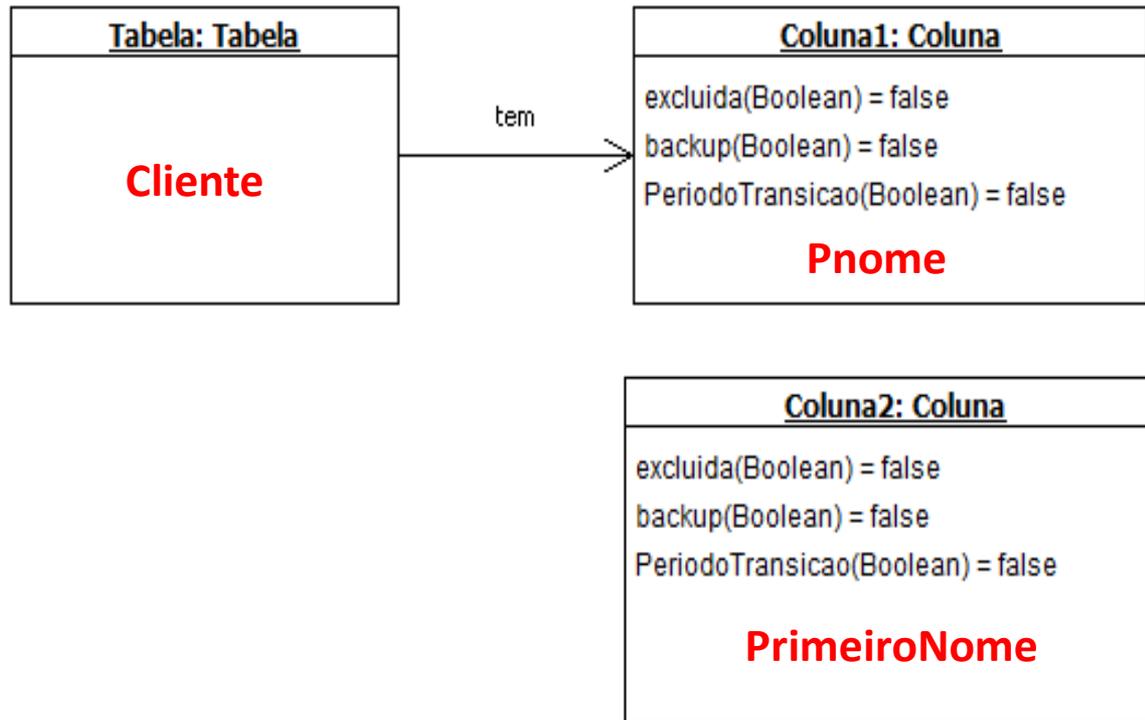
Exemplo: Refatoração Renomear Coluna

Diagrama de Classes



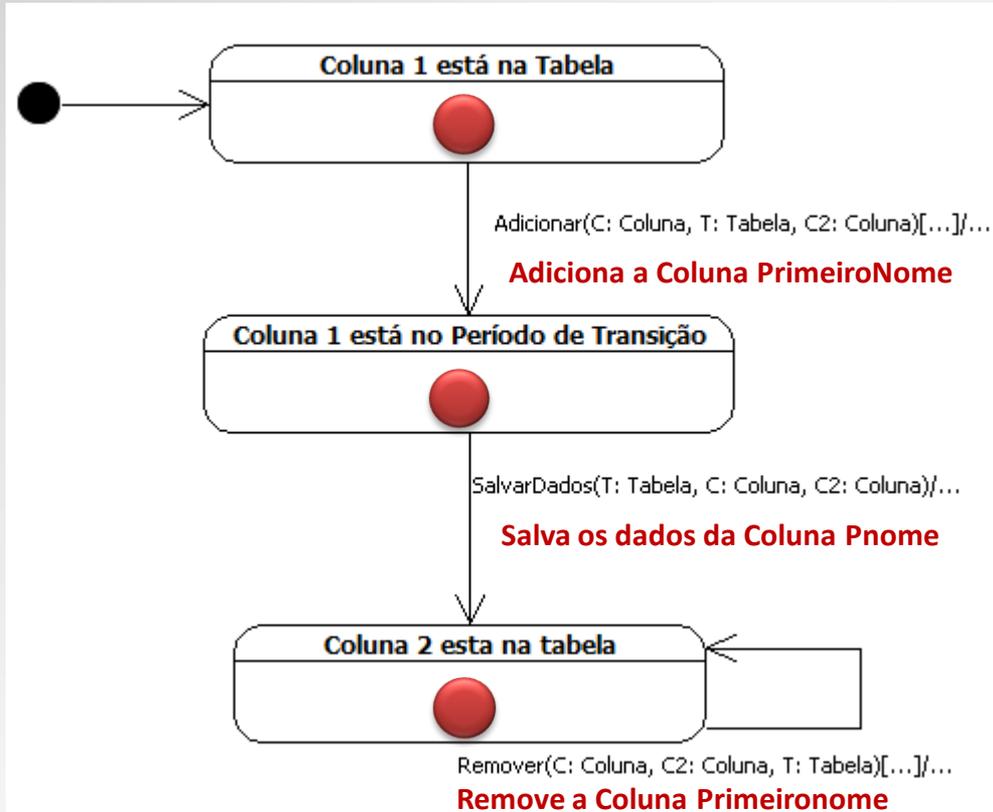
Exemplo: Refatoração Renomear Coluna

Planejamento: Início



Exemplo: Refatoração Renomear Coluna

Diagrama de Estados



Pnome: Coluna

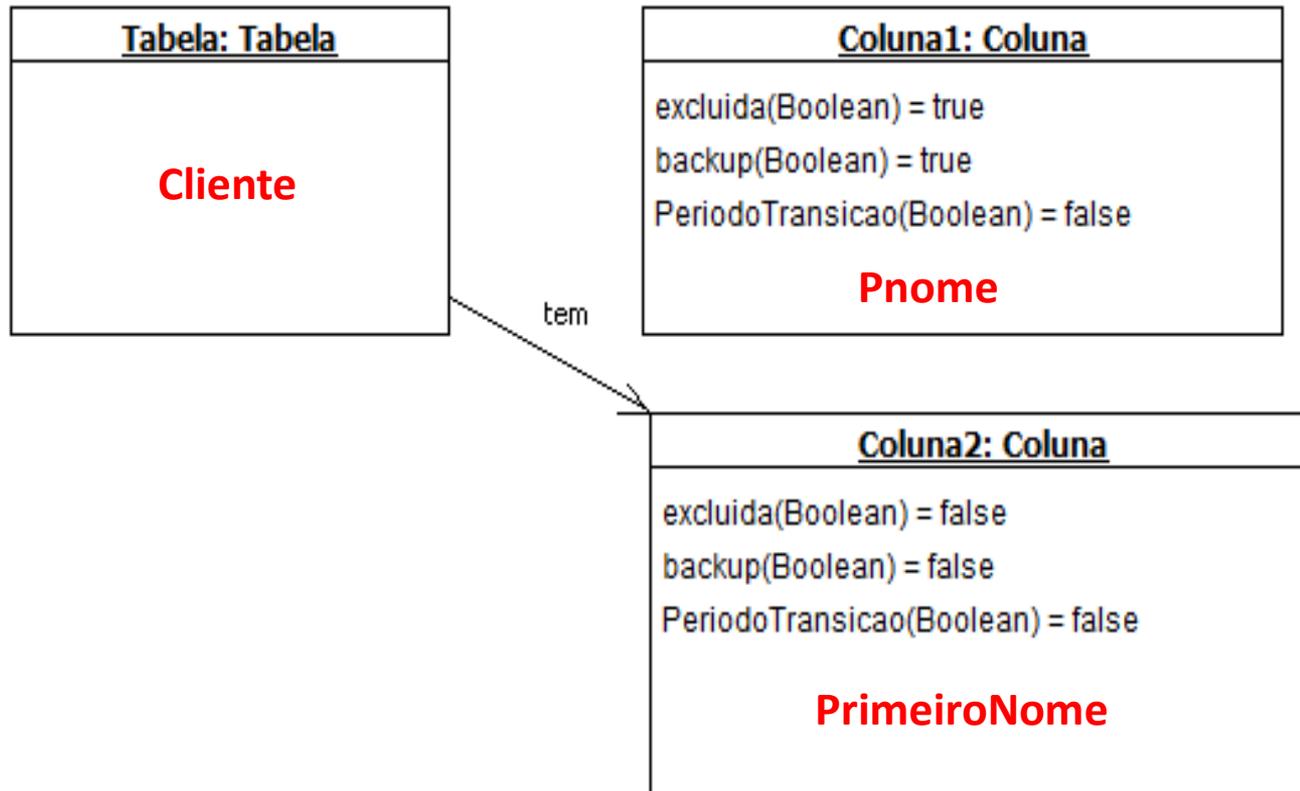
excluída: **True**
backup: **True**
periodoTransicao: **False**

PrimeiroNome: Coluna

excluída: False
backup: False
periodoTransicao: False

Exemplo: Refatoração Renomear Coluna

Planejamento: Objetivo



Exemplo: Refatoração Renomear Coluna

```
(define (domain Renomear_Coluna)
  (:requirements :typing :negative-preconditions)
  (:types
    Tabela - object
    Coluna - object
  )
  (:predicates
    (tem ?tab - Tabela ?col - Coluna)
    (excluida ?col - Coluna)
    (backup ?col - Coluna)
    (PeriodoTransicao ?col - Coluna)
  )
  (:action Adicionar
    :parameters (?C - Coluna ?T - Tabela ?C2 - Coluna)
    :precondition
      (and
        (not (excluida ?C))
        (not (backup ?C))
        (tem ?T ?C)
      )
    :effect
      (and
        (tem ?T ?C2)
        (not (excluida ?C2))
        (not (backup ?C2))
        (not (tem ?T ?C))
      )
  )
)
```

Definição do Domínio: PDDL

Planning Domain Definition Language

Exemplo: Refatoração Renomear Coluna

```
(:action SalvarDados
  :parameters (?T - Tabela ?C - Coluna ?C2 - Coluna)
  :effect
    (and
      (backup ?C)
      (PeriodoTransicao ?C)
    )
)

(:action Remover
  :parameters (?C - Coluna ?C2 - Coluna ?T - Tabela)
  :precondition
    (and
      (tem ?T ?C2)
      (backup ?C)
    )
  :effect
    (and
      (excluida ?C)
      (not (tem ?T ?C))
      (not (PeriodoTransicao ?C))
    )
)
)
```

Definição do Domínio: PDDL

Planning Domain Definition Language

Exemplo: Refatoração Renomear Coluna

```
(define (problem Planning_Problem)
  (:domain Renomear_Coluna)
  (:objects
    Tabela - Tabela
    Coluna1 - Coluna
    Coluna2 - Coluna
  )
  (:init
    (tem Tabela Coluna1)
  )
  (:goal
    (and
      (tem Tabela Coluna2)
      (not (excluida Coluna2))
      (not (backup Coluna2))
      (not (PeriodoTransicao Coluna2))
      (excluida Coluna1)
      (backup Coluna1)
      (not (PeriodoTransicao Coluna1))
    )
  )
)
```

Definição do Domínio: PDDL

Planning Domain Definition Language

Exemplo: Refatoração Renomear Coluna

Plan Report

Introduction

Project: Renomear Coluna

Domain: Planning Domain

Problem: Planning Problem

Date/Time: 2012-05-23 22:33:48

Plan validity: **Unknown. Plan not validated.**

itSIMPLE message:

Planner generated a solution.

Planner

Name: Metric-FF

Version: Windows

Author(s): J. Hoffmann

Institution(s):

Link: <http://members.deri.at/~joergh/ff.html>

Description:

Statistics

Tool total time: 0.154

Planner time: 0.00

Parsing time:

Number of actions: 3

Makespan:

Metric value:

Planning technique:

Additional:

Plan

0: (ADICIONAR COLUNA1 TABELA COLUNA2) [1]

1: (SALVARDADOS TABELA COLUNA1 COLUNA2) [1]

2: (REMOVER COLUNA1 COLUNA2 TABELA) [1]

Planner Console Output

Execução Planejador: Metric-FF

Conclusões

- Uma refatoração de banco de dados representa uma alteração no esquema do banco de dados, baseada no requisito do usuário.
- Os requisitos mudam constantemente, a aplicação e o banco de dados precisam acompanhar todas as mudanças sem prejudicar os dados existentes.
- Este trabalho teve como objetivo apresentar uma nova forma de realizar refatorações de banco de dados utilizando técnicas de planejamento automático.
- Estas técnicas tem como objetivo conhecer os passos e restrições existentes para se realizar uma refatoração.
- O processo de refatoração proposto por Domingues (2014) pode ser automatizado utilizando as técnicas de planejamento automático aqui apresentadas e implementados em ambiente de *software* amigável para suporte de apoio ao Administrador de Banco de Dados.

Sugestão para Trabalhos Futuros

- A sequência de passos do processo de refatoração é complexa e possui um conjunto de regras bem definidos.
- A aplicação dos passos e a ordem sugerem que o processo seja particularmente flexível e se adapte a qualquer conjunto de refatorações dado como entrada.
- O conjunto de refatorações pode variar de acordo com os requisitos do usuário.
- Assim, o uso de **Autômatos Adaptativos** e **Tabelas de Decisão** podem contribuir muito para a pesquisa de evolução de esquemas de banco de dados.