

NOVA ORGANIZAÇÃO PARA ESTRUTURA DE DADOS EM BANCOS RELACIONAIS: ESTUDO DE CASO

Aluno: João Paulo Poffo

Orientador: Marcel Hugo

Banca: Alexander Roberto Valdameri
Adilson Vahldick

Semestre: 2009-1

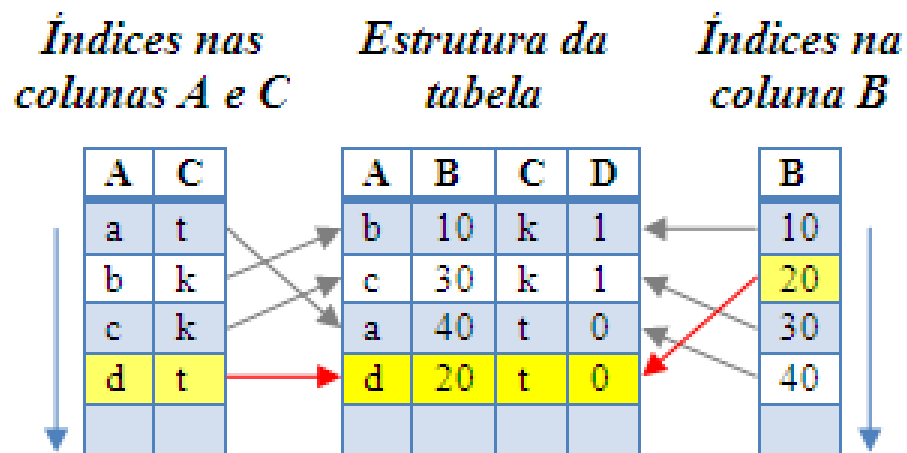
Roteiro

- Introdução
- Contexto atual do tema
- Fundamentação teórica
- Desenvolvimento do Tipo de Tabela e da prova de conceito:
 - Requisitos
 - Especificação
 - Implementação
 - Resultado
- Conclusão

Introdução

- Problema
- Nova estrutura
- Objetivo

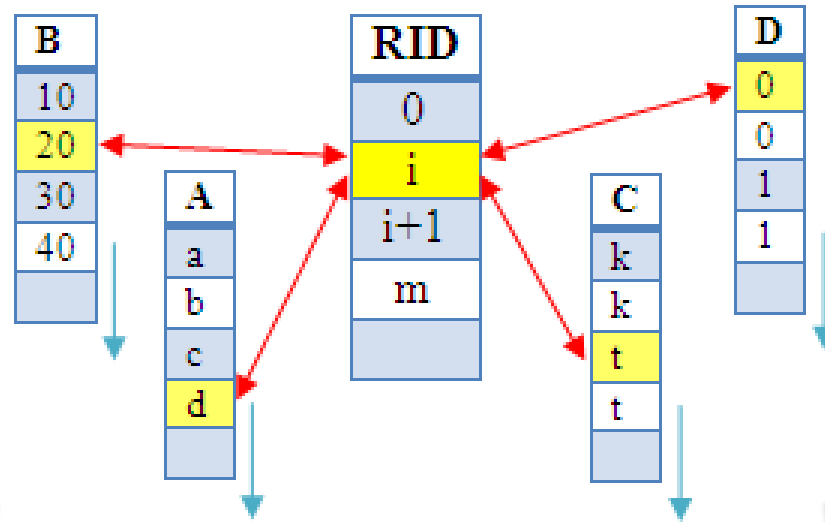
Problema



Redundância

Desempenho

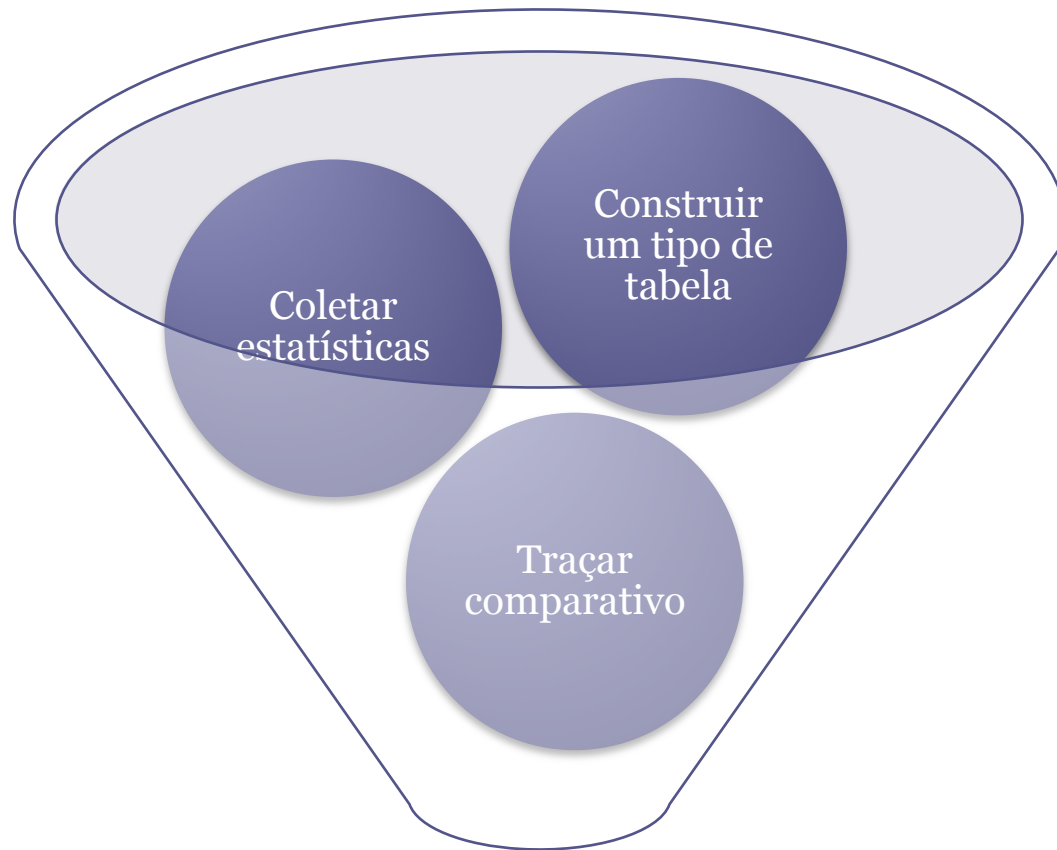
Nova estrutura



Redundância

Desempenho

Objetivos



Desenvolver uma arquitetura para armazenamento e obtenção de dados em um SGDB

Contexto atual do tema

- Trabalhos correlatos
- Novas tecnologias

Trabalhos correlatos

Maria: Tipo de tabela

- Tipo de tabela do MySQL
- Especialização do MyISAM
- Controle transacional
- Três formatos de linha

Indexação Bitmap

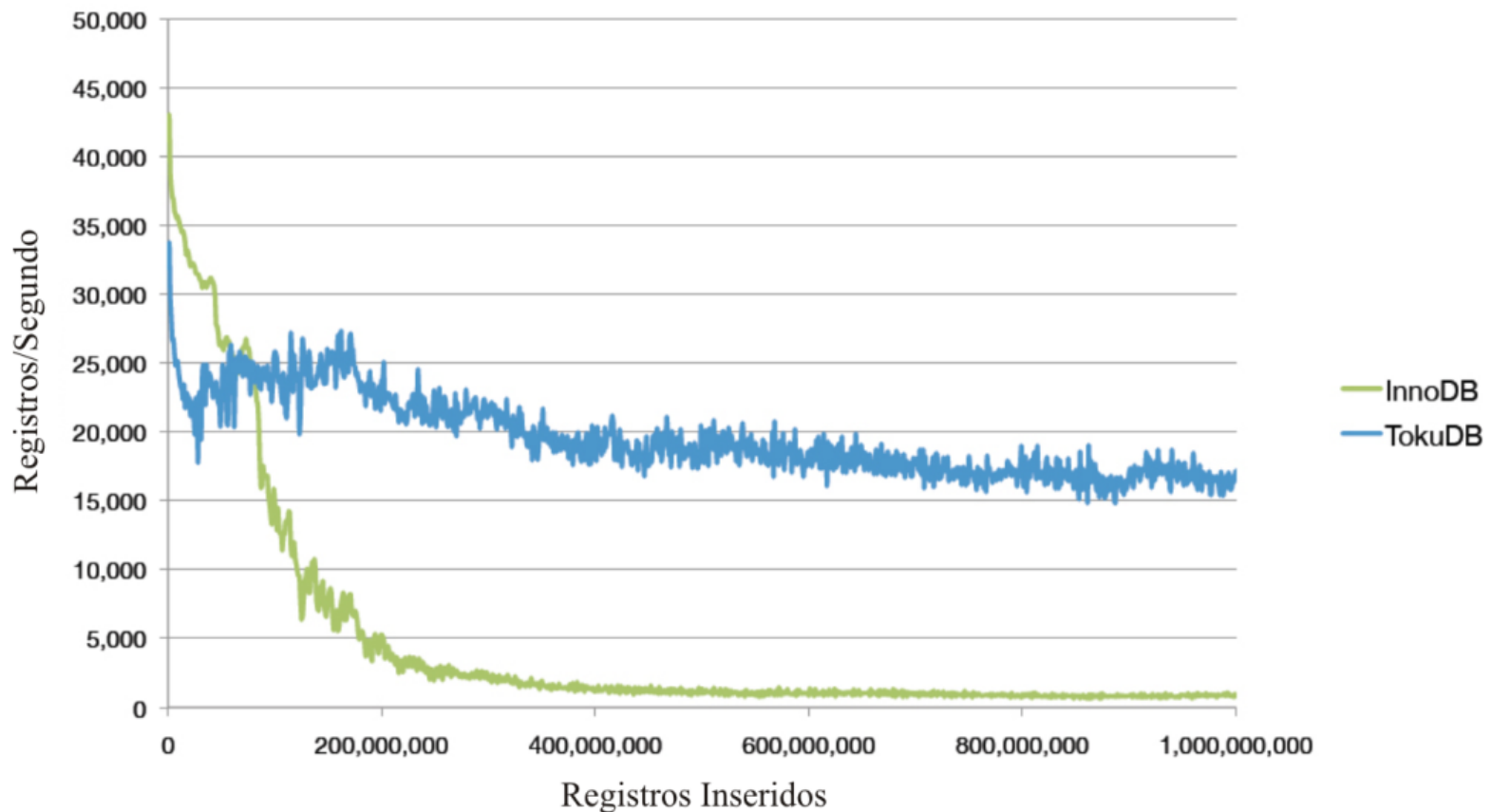
- Mapeamento de bits
- Características
 - Acesso direto ao dado
 - Intersecção de índices
 - Otimização de consultas

Comparativo desempenho entre bancos de dados de código aberto

- Comparativo entre MySQL e PostgreSQL
- Ferramenta OSDB
- Testes: Carga e estrutura; Seleções; Junções; Projeções; Agregações.

Novas tecnologias: Árvores fractais

Teste de Inserção de 1 Bilhão de Registros



Novas tecnologias: C-Store

Michael Stonebraker et al., “C-Store: A Column-oriented DBMS”,
Proc 2005 VLDB Conference,
Trondheim, Norway, Sept 2005.

Fundamentação teórica

- Estrutura de dados
- SGDBs
- Tipos de tabela
- Prova de conceito

Estruturas de dados

Tipos primitivos

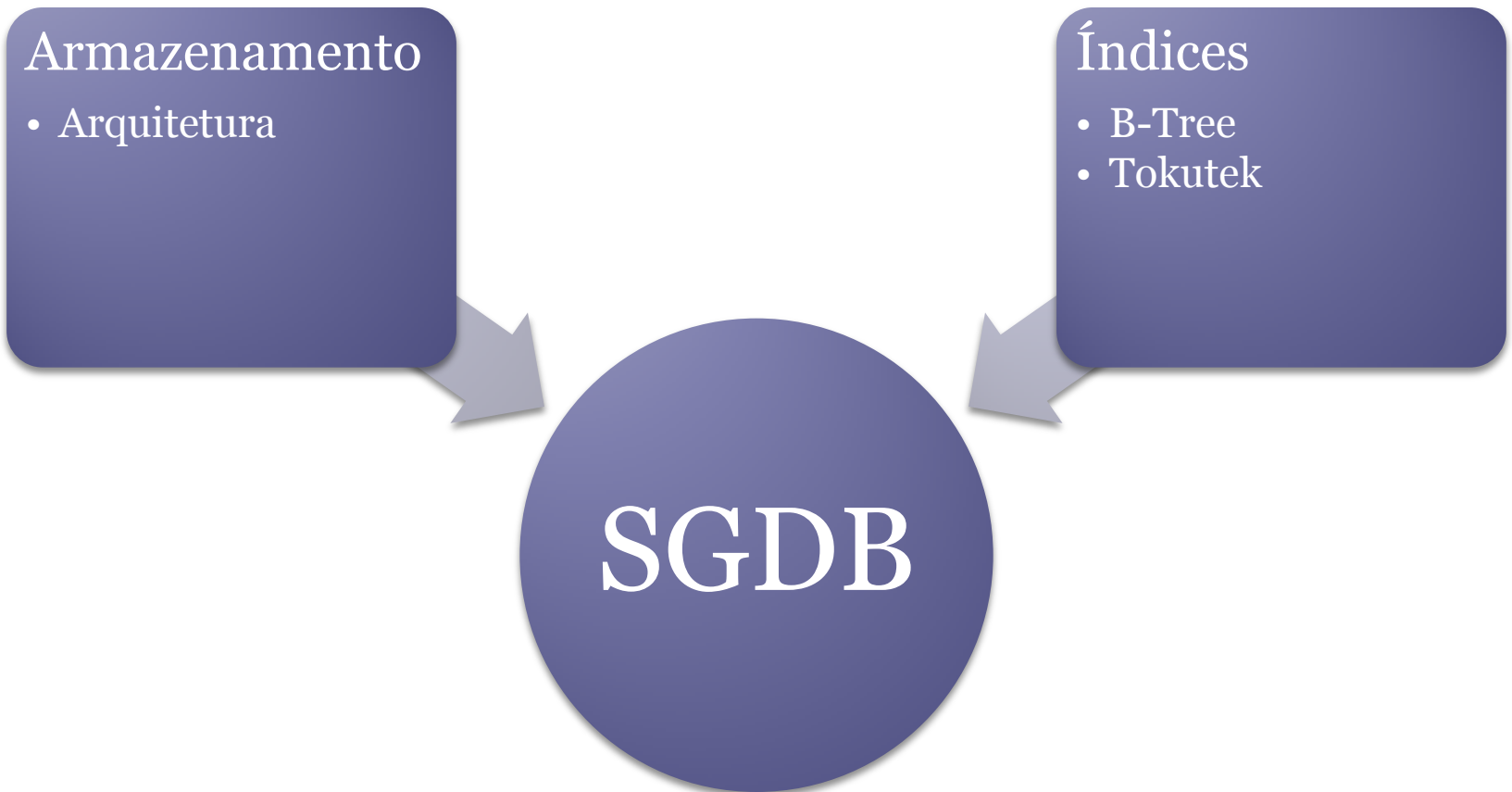
- Bit
- Byte
- Word
- Int
- Real

Coleções de dados

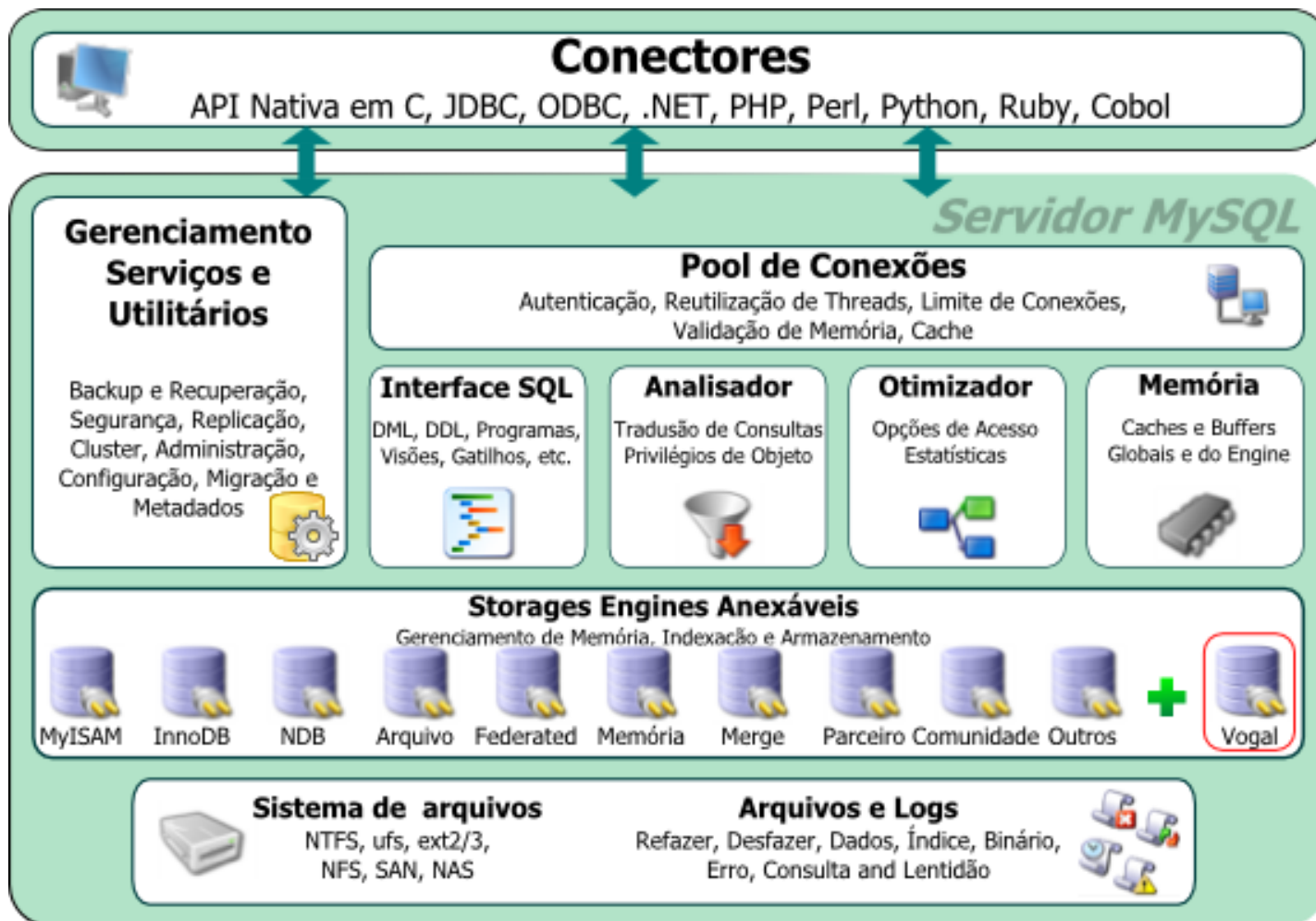
- Listas
- Pilhas
- Árvores

Bancos de dados

Sistemas Gerenciadores de Bancos de Dados



Tipos de tabela



Prova de conceito

Benchmarking

- Testes automatizados
- Métricas de comparação
- Problemas comuns
 - Confiabilidade
 - Utilidade

OSDB

- Ferramenta de benchmark
- Características importantes
 - Portáteis
 - Fácil compreensão
 - Escaláveis
- Código aberto

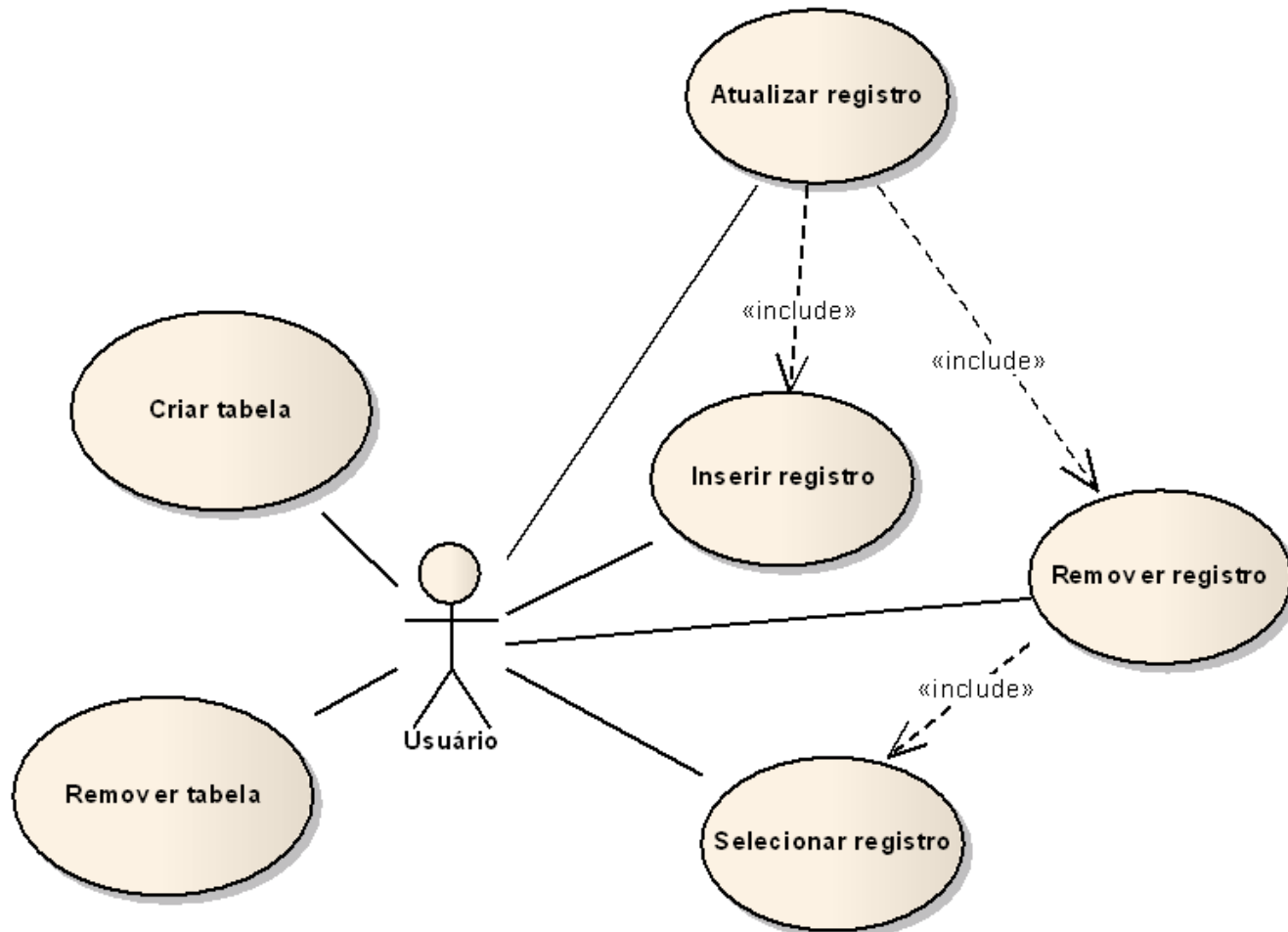
Desenvolvimento

- Tipo de Tabela
- Prova de conceito

Desenvolvimento do Tipo de Tabela

- Requisitos
- Especificação
- Implementação
- Resultado

Requisitos Funcionais e Especificação



Outros Requisitos

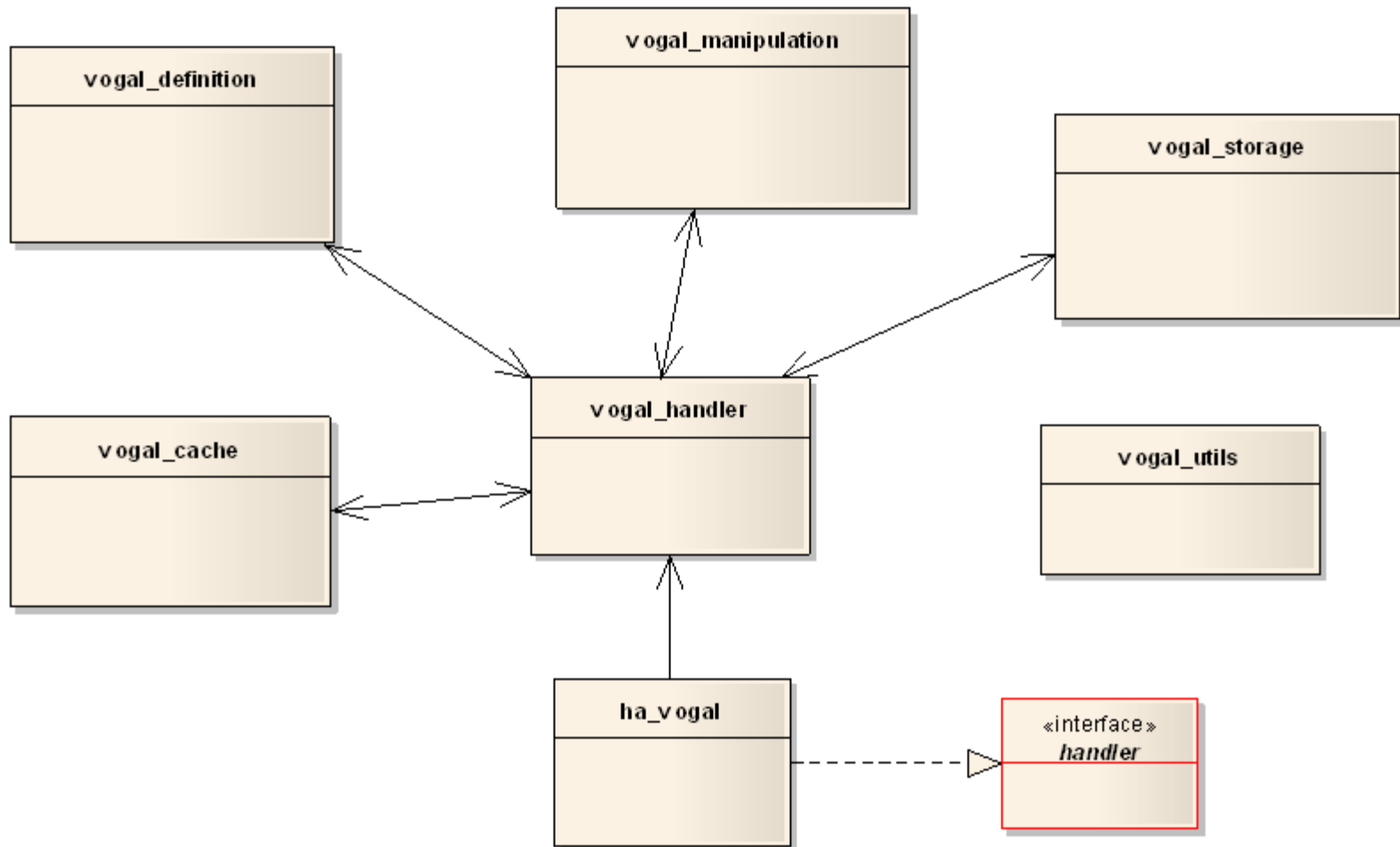
Não funcionais

- C++ em Linux
- Armazenamento dos dados e metadados

Regras de negócio

- SQL padrão MySQL
- Campos texto ou inteiro
- Limite de 127 caracteres
- Inteiros entre -2147483648 e 2147483647
- Sem validação de limitações
- Tabelas definidas com os campos e seus tipos

Especificação



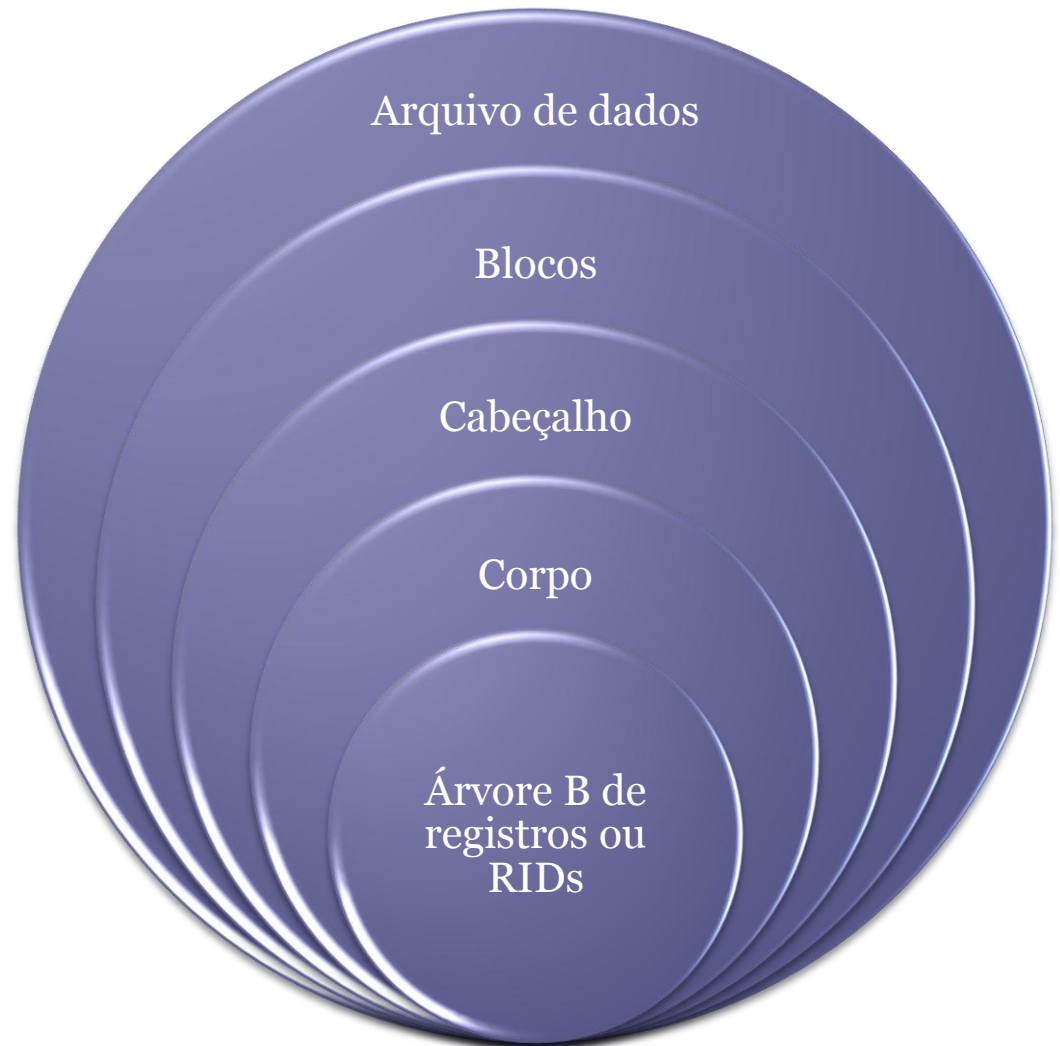
Especificação

Características

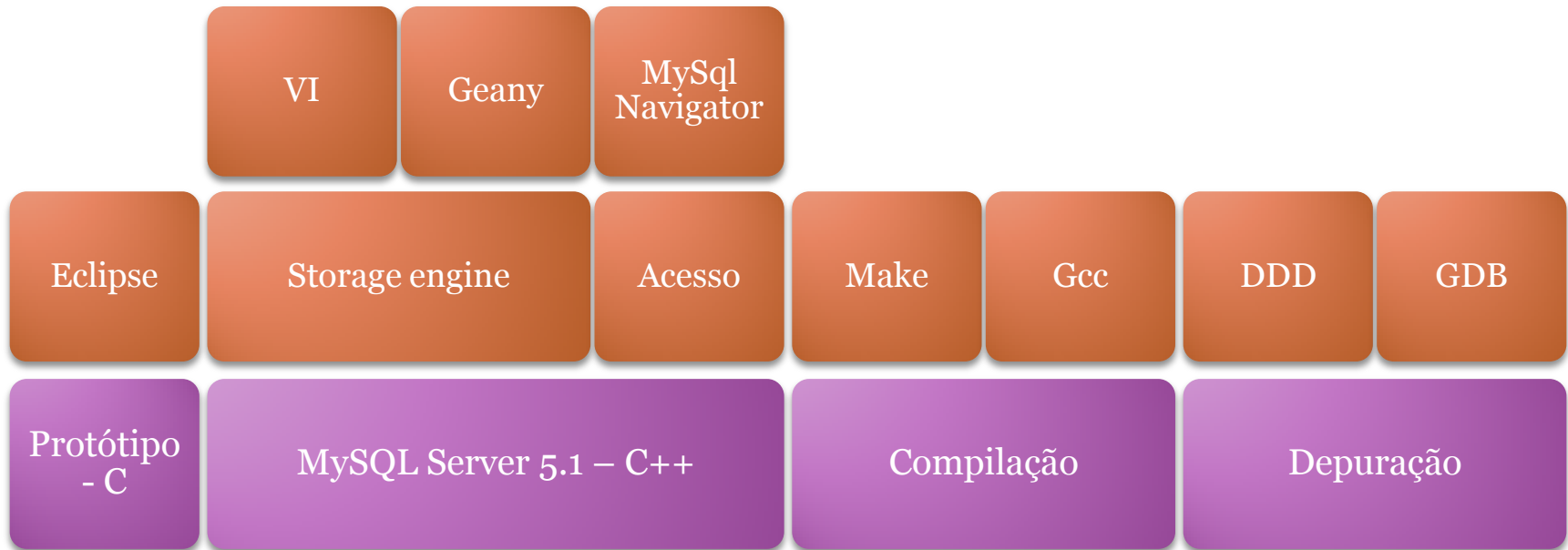
Tamanho variável

Tamanho potencialmente ilimitado

Espaço mínimo ocupado



Implementação: Ferramentas



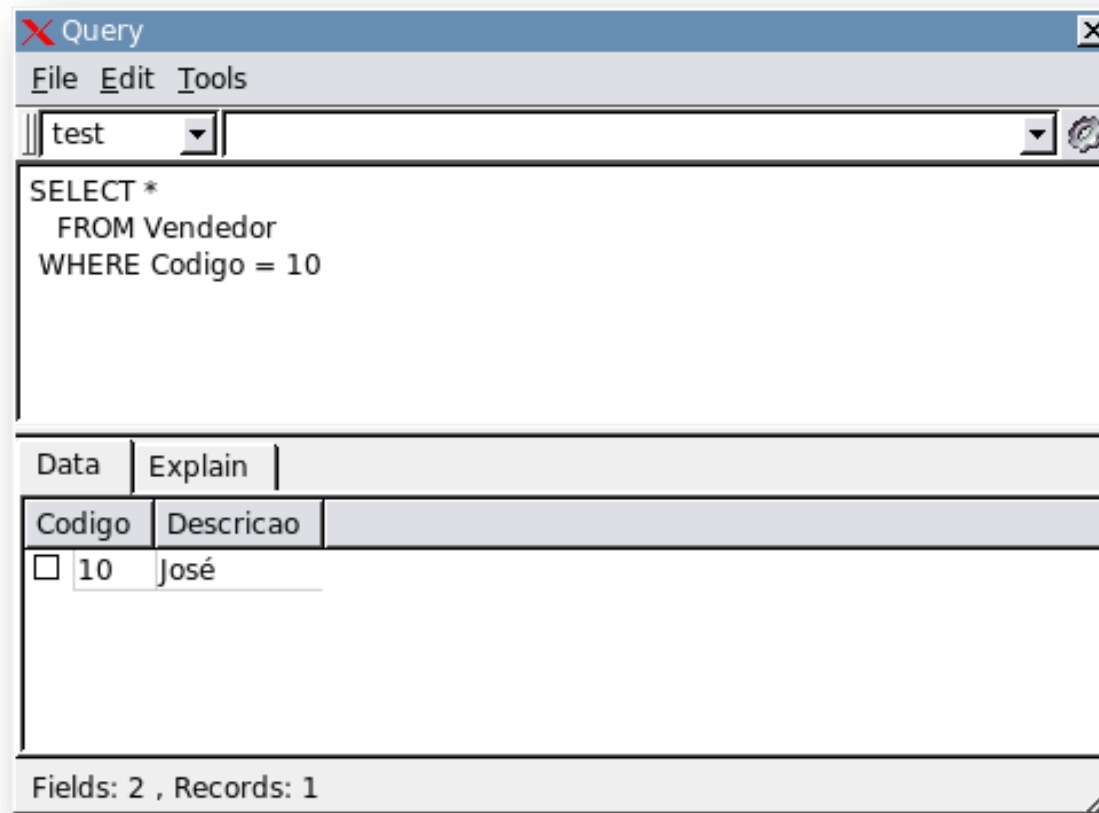
Linux – Ubuntu (VM)

Windows - Sun Virtual Box

Implementação: Código desenvolvido

```
1 // Lê buffer carregado do arquivo para a memória e separa as informações relevantes
2 int vogal_definition::parseBlock(CursorType * cursor, ColumnCursorType * column, BlockCursorType * block) {
3     // Declara variáveis
4     ValueListRoot *list = NULL;
5     NodeType *node = NULL;
6     BlockOffset *neighbor;
7     BigNumber dataCount;
8     GenericPointer p;
9
10    // Monta lista de rids e offsets do bloco
11    if (block->nodesList)
12        vlFree(&block->nodesList);
13    block->nodesList = vlNew(true);
14    if (block->offsetsList)
15        vlFree(&block->offsetsList);
16    block->offsetsList = vlNew(true);
17
18    // Posição atual
19    p = block->buffer + sizeof(BlockHeaderType);
20
21    // Lê a quantidade de registros do bloco
22    if (!m_Handler->getStorage()->readDataSize(&p, &dataCount))
23        ERROR("Erro ao ler a quantidade de registros no block");
24
25    if (dataCount > 0) {
26        for (int i = 0; i < dataCount; i++) {
27            // Auxiliares
28            neighbor = new BlockOffset();
29            // Lê primeiro nó a esquerda
30            if (!m_Handler->getStorage()->readSizedNumber(&p, neighbor))
31                ERROR("Erro ao ler nó a esquerda da árvore de registros");
32            // Adiciona informação à lista
33            vlAdd(block->offsetsList, neighbor);
34            node = m_Handler->getManipulation()->parseRecord(cursor, column, block, &p);
35            // Adiciona informação à lista
36            vlAdd(block->nodesList, node);
37            node = NULL;
38        }
39    }
40    // Se a quantidade for maior que zero
41    // Depois do último nó sempre tem o nó a direita
42    neighbor = new BlockOffset();
43    if (!m_Handler->getStorage()->readSizedNumber(&p, neighbor))
44        ERROR("Erro ao ler nó a direita da árvore de registros");
45    // Adiciona informação à lista
46    vlAdd(block->offsetsList, neighbor);
47 }
```

Implementação: Operacionalidade



Resultado: Discussão

- Árvore B / Árvore B+
- Atualização de ponteiros
- Gerenciamento de memória
- MySQL em C++ parcialmente orientado a objetos
- Interface específica para índices

Resultado: Limitações

Limitações	Tipo de tabela	Maria	Vogal
controle transacional		Possui	Não possui
integridade referencial		Possui	Não possui
ACID		Não possui	Não possui
tamanho máximo do banco de dados		256TB	100MB
tamanho máximo de cada registro		64KB	1KB
máximo de colunas por tabela		4096	1024
tamanho máximo de campos LOB		4GB	Não possui
tamanho máximo de campos de texto		64KB	127B
tamanho máximo de campos numéricos		64 <i>bits</i>	32 <i>bits</i>
menor data válida		1000	Não possui
maior data válida		9999	Não possui

Desenvolvimento da prova de conceito

- Requisito e Especificação
- Implementação
- Resultado

Requisito e Especificação

Consulta

- 100% (alt/lar)
- 10% (alt/lar)
- Reg. único

Comparativo entre Vogal, InnoDB e MyISAM (OSDB)

Limitações

- Máximos
- Mínimos
- ...

Carga e estrutura

- Criação
- Carga
- Atualização
- Exclusão

Capacidades

- Conjunto
- Junção
- ACID
- ...

Implementação

	Resultado esperado	Função OSDB
Carga e estrutura	criação das tabelas	create_tables
	carga de dados	load
	atualização de um registro	upd_unique
	atualização de 100% dos registros	upd_100pct
	exclusão de um registro	del_unique
	exclusão de 100% dos registros	del_100pct
Consulta	100% dos registros (projeção total)	selproj_100_100pct
	100% dos registros (projeção parcial)	selproj_100_50pct
	10% dos registros (projeção total)	selproj_10_100pct
	10% dos registros (projeção parcial)	selproj_10_50pct
	um registro (projeção total)	selproj_1_100pct
	um registro (projeção parcial)	selproj_1_50pct

Resultado: Limitações e Capacidades

Limitações	Tipo de tabela	InnoDB	MyISAM	Vogal
tamanho máximo do banco de dados		64TB	256TB	100MB
tamanho máximo de cada registro		8KB	64KB	1KB
máximo de colunas por tabela		1000	4096	1024
tamanho máximo de campos LOB		4GB		Não possui
tamanho máximo de campos de texto		64KB		127B
tamanho máximo de campos numéricos		64 bits		32 bits
menor data válida		1000		Não possui
maior data válida		9999		Não possui

Capacidades	Tipo de tabela	InnoDB	MyISAM	Vogal
união, diferença		Sim	Sim	Não
junção interna				
junção externa				
consulta interna (<i>sub-select</i>)				
LOBs				
domínio				
programa				
ACID				
integridade referencial				
controle transacional				

Resultado: Testes com 100mil registros

	Tipos de tabela	InnoDB	MyISAM	Vogal
	Testes (segundos)			
Carga e estrutura	criação das tabelas	0,06	0,03	40
	carga de dados	19,16	4,77	77000 (~22h)
	atualização de um registro	0,32	0,81	30
	atualização de 100% dos registros	2,69	2,65	65000 (~18h)
	exclusão de um registro	1,44	0,12	40
	exclusão de 100% dos registros	1,75	0,01	2400 (~40m)
Consulta	100% dos registros (projeção total)	2,26	1,91	40
	100% dos registros (projeção parcial)	1,25	1,14	20
	10% dos registros (projeção total)	0,52	0,37	30
	10% dos registros (projeção parcial)	0,36	0,25	20
	um registro (projeção total)	0,34	0,13	30
	um registro (projeção parcial)	0,70	0,14	40

Conclusão

- Resultado
- Problema
- Limitações/Extensões

Resultado

- O objetivo geral deste trabalho foi alcançado
- Os objetivos específicos foram alcançados
- O desempenho não foi o esperado
- Operacionalidade maior que a esperada (MySQL)

Problema

- Gerenciamento de memória
- Mecanismo de índices do MySQL
- Reorganização de deslocamentos
- Testes comparando com tabelas com índices secundários

Limitações / Extensões

- Tipos de campo de ponto flutuante, data, LOBs, entre outros
- Gerenciamento de memória mais robusto
- Gerenciamento de transações
- Validar obrigatoriedade e limites de campos
- Aproveitar mecanismo de incremento automático no valor das colunas presente no MySQL
- Implementar seleção por faixa (LIMIT)
- Configurar o arquivo de dados, seu tamanho, e adicionar possibilidade de que ele aumente automaticamente
- ...