

ANÁLISE COMPARATIVA PARA AVALIAÇÃO DE TECNOLOGIAS DE BANCO DE DADOS PARA DISPOSITIVOS MÓVEIS

Centro de Ensino Superior de Juiz de Fora, Juiz de Fora, MG

Taiany COELHO

Giuliano PRADO

Resumo: Com o aumento do desenvolvimento de aplicativos móveis houve a necessidade de adequar às ferramentas de armazenamento da informação. Com isso os Sistemas de Banco de Dados foram adaptados de forma a atender a requisitos impostos pela mobilidade, através da evolução dos Bancos de Dados Distribuídos para atender a plataforma móvel e suas aplicações diversas. Três dos principais bancos de dados móveis disponíveis no mercado serão analisados nesse projeto, para que funcionalidades e desempenho dos mesmos possam ser testados, de forma que uma análise comparativa possa ser realizada, a fim de indicar possíveis soluções na persistência de dados móveis e desafios futuros para a evolução dos bancos de dados para esse fim.

Palavras-chave: tecnologia, sistema, desenvolvimento mobile, aplicativos

1 INTRODUÇÃO

O sistema de informação móvel surgiu para atender as necessidades de informação acessível independente da localização. Esses novos sistemas estão ganhando destaque no mercado popular, com suas ferramentas de comunicação e entretenimento.

As aplicações móveis estão associadas com a capacidade de locomoção do hardware, dados e software em aplicações computacionais. Ela tornou-se possível graças à integração das comunicações móveis e das tecnologias que envolvem microcomputadores. (Cortês; Lifschitz, 2003)

Com o surgimento da possibilidade dos sistemas serem móveis, criou-se a responsabilidade de fazer com que essas aplicações estejam contextualizadas com a sua dinâmica. Novos sistemas operacionais surgiram com interfaces voltadas à praticidade, novas redes de comunicação e os sistemas de bancos de dados tiveram suas características reformuladas ou adaptadas, os quais são objetos desse trabalho. Grande variedade de componentes é utilizada no desenvolvimento das aplicações que, ao se mesclarem, concedem programas que se interligam. Cada dispositivo móvel possui uma plataforma, ou seja, as aplicações precisam ser desenvolvidas para uma plataforma específica, seja ela Android, Windows Phone, dentre outras.

Com toda a estrutura da aplicação móvel funcionando, houve a necessidade da informação ser armazenada, para que a mesma possa ser acessada a qualquer momento. Desta forma, se fez necessário um sistema de gerenciamento de dados, os SGDB Móveis, que é uma adaptação dos SGBD (Sistemas Gerenciadores de Bases de Dados) tradicionais para uma nova função: serem executados em um dispositivo móvel agindo com um papel de servidor de dados local e global.

Os Sistemas de Banco de Dados (SBD) e seus componentes básicos oferecem todas as ferramentas para confiança, segurança, disponibilidade, acesso eficiente aos dados e integridade. Com isso os SBDs, podem se localizar em redes fixas, desempenhando uma função tradicional dos servidores de dados para todos os seus projetos, como também funcionar com um novo papel de servidores de dados locais, porém, agindo em um dispositivo móvel (Rainone, 2007).

Portanto o primeiro objetivo deste trabalho é realizar um levantamento de pesquisa sobre banco de dados móveis, assim como mostrar os novos recursos oferecidos e as adaptações realizadas no mesmo. Pretende-se, também, fazer um estudo comparativo de desempenho e funcionalidades dentre os três principais bancos de dados móveis disponíveis no mercado.

2 ESTRUTURA DA ARQUITETURA PARA APLICAÇÕES COM BD MÓVEIS

Para que as aplicações funcionem de forma a atender as exigências do ambiente móvel foi necessário encontrar alternativas para fazer a aplicação funcionar adequadamente. Essas alternativas servem para que as aplicações móveis consigam executar suas funções remotamente. Os chamados Agentes Móveis têm essa função principal, de levar as instruções, dados e um estudo de execução. Para que a informação seja acessada remotamente precisa existir em volta da computação uma arquitetura montada, a qual irá dar suporte a toda estrutura para a aplicação.

Agentes móveis são programas enviados de uma máquina para executar uma tarefa remotamente. As características que estão por volta desse programa têm como função a cooperação, autonomia, representatividade com a finalidade de suprir as necessidades para o bom funcionamento de modelos que utilizam esse paradigma. Dois fatores principais motivam a utilização dos agentes móveis em Banco de Dados Móveis. Primeiramente, os agentes móveis dispõem de um método ágil e eficiente para obter informações ou serviços. Clientes móveis são lançados na rede e procuram em toda parte pela informação ou serviço desejado. Outra característica

persistência dos dados de todo o cenário citado anteriormente, tornam-se necessárias mudanças na política tradicional dos bancos de dados. O banco de dados móveis é uma adaptação dos Sistemas de banco de dados distribuídos, com o novo conceito da mobilidade na qual existem restrições impostas pelos ambientes de comunicação móvel: a limitada largura de rede de comunicação sem fio, a mobilidade e as frequentes desconexões dos dispositivos móveis, assim como a mobilidade dos dados (Conceição, 2005).

3.1 BANCOS DE DADOS MÓVEIS

Para possibilitar a existência dos bancos de dados móveis, as características dos bancos de dados centralizados e distribuídos foram integradas e adaptadas ao ambiente móvel. Adaptadas, pois eles possuem características em comum que precisam ser levadas em consideração. Devido à influência das características de um ambiente móvel tais como acesso remoto sem fio, limitação de processamento, largura de banda móvel, dentre outros, o gerenciamento de dados e o controle de concorrência em ambientes de computação móvel tem sido objeto de pesquisas (Gonçalves, 2006).

A seguir são definidas as semelhanças entre os bancos de dados móveis e os bancos de dados distribuídos.

3.2 O GERENCIAMENTO DE DADOS DISTRIBUÍDOS POR BANCO DE DADOS MÓVEIS

Bancos de dados móveis podem ser baseados em gerenciamentos de dados distribuídos em duas possíveis situações:

1. As informações dos dados estão organizadas principalmente entre os componentes com fio. Uma estação base faz seu próprio gerenciamento de dados, tendo um SGBD com funcionalidades adicionais para procurar unidades móveis e para gerenciar consultas e transações do ambiente móvel.

2. A base de informações é distribuída pelos componentes com e sem fio. A responsabilidade da gestão dos dados é dividida entre as unidades móveis e as estações base.

Com isso, segundo Elmasri e Navathe (2005), muitas das funções de gerenciamento de dados distribuídos também se aplicam aos bancos de dados móveis, desde que os seguintes pontos sejam levados em consideração:

- Distribuição de dados e replicação – Os dados como já visto, tem a opção de não estar totalmente distribuídos entre as estações base e as unidades móveis. Devido a esse fator, as restrições de consistência aumentam a questão em volta do problema de gerenciar os dados em cache. Dessa forma, esta precisa adequar à unidade móvel o acesso àqueles dados mais repetidamente acessados e atualizados;

- Modelos de transação – No ambiente móvel a questão em relação as tolerância a falhas e a ocorrência das transações são mais difíceis de serem manipuladas. A transação móvel é concretizada através de diversas estações base e possivelmente e um agrupamento de dados, dependendo da localização e da locomoção da unidade móvel. A composição principal da implantação de uma transação fica escassa, por exemplo, quando se tem base dividida pelos componentes com e sem fio. Com isso, as propriedades ACID (atomicidade, consistência, isolamento e durabilidade) das transações podem precisar de ajustes e novos modelos de transação devem ser redefinidos de modo que atende as exigências da mobilidade.

3.2.1 Replicação de Dados e Sincronização

Replicação é a maneira no qual um arquivo ou vários arquivos são copiados de seu local de origem para outros locais inclusos em um sistema distribuído. Para garantir a disponibilidade da informação e o desempenho das aplicações sobre as unidades móveis, os dispositivos levam seus dados replicados, isto é, cópias das informações contidas no banco de dados distribuído. Após os dados serem distribuídos, os dados são mantidos atualizados de forma que o usuário tenha certeza que está disponível para ele à versão mais recente da aplicação, esse processo é chamado de Sincronização. Todos os outros locais que acessam os dados recebem a informação atualizada (Trasel; Veronez, 2005).

3.2.2 Recuperação de Falhas

O método de recuperação a falhas menciona à competência que um sistema tem de reservar a consistência do banco de dados devido a falhas que a aplicação

pode ter, de operação ou dos meios de comunicação. Por estar em um ambiente móvel a aplicação está sujeita a falhas tanto do *software* como do *hardware*. Uma parte essencial do sistema de banco de dados é um esquema de recuperação responsável pela detecção de falhas e pela restauração do banco de dados para um estado consistente que existia antes da ocorrência da falha. Para que o sistema não seja prejudicado devido às falhas em seus componentes, erros devem ser detectados o mais rápido possível, através de um diagnóstico apropriado. Para recuperar os dados, informações relevantes são armazenadas em um local fixo durante o processamento de transações. Em sistemas distribuídos, a recuperação de falhas é baseada em pontos de recuperação, conhecidos como *checkpoints*. No caso de falhas e desconexões, a aplicação usa o último *checkpoint* salvo para reiniciar sua execução (Ito, 2001).

3.2.3 Segurança

Os riscos de segurança de sistemas e de redes de computadores são agravados quando inseridos no ambiente móvel, muito mais propenso a ataques e falhas. Na computação móvel, a portabilidade dos dispositivos usados pode levar à perda das unidades, o que caracteriza perda de dados e de confidencialidade. A única forma de prevenir a falta de confiabilidade é o uso de encriptação (*encryption*, fase da criptografia) e de mecanismos que assegurem identificação, autenticação e controle de acesso. Estas não são características particulares de segurança em ambiente móvel, a não ser pelo fato de que a proteção de um dispositivo móvel deve ser simplificada, devido à escassez de recursos e pouco poder de processamento destes computadores (Camargo, 2003).

3.3 CATEGORIAS BANCOS DE DADOS MÓVEIS

Algumas categorias de bancos de dados móveis estão disponíveis no mercado, a seguir serão listadas as categorias que definem os três bancos analisados nesse artigo.

3.3.1 Banco de Dados Móveis Relacional

Em um banco de dados relacional, todos os dados são guardados em tabelas. Estas têm uma estrutura que se repete a cada linha. São os relacionamentos entre as tabelas que as tornam “relacionais” (Praciano, 2013).

3.3.2 Banco de Dados Móveis Orientado a Objetos

Um banco de dados orientado a objeto é um banco em que cada informação é armazenada na forma de objetos, e só pode ser manipuladas através de métodos definidos pela classe que esteja o objeto. Existem pelo menos dois fatores que levam a adoção desse modelo, a primeira é que banco de dados relacional se torna difícil trabalhar com dados complexos. A segunda é que aplicações são construídas em linguagens orientadas a objetos (Java, C++, C#) e o código precisa ser traduzido para uma linguagem que o modelo de banco de dados relacional entenda, tornando essa tarefa muito tediosa (Galante; Moreira; Brandão, 1999).

4 BANCOS DE DADOS MÓVEIS UTILIZADOS PARA A AVALIAÇÃO COMPARATIVA

Nessa seção serão apresentados os principais Bancos de dados móveis gratuitos utilizados no desenvolvimento de aplicações móveis, segundo o critério de usabilidade pelo mercado. A escolha destes bancos de dados foi feita de acordo com o número de sistemas móveis que os utilizam, já que no levantamento de dados para a realização desse artigo foi constatado um maior volume de material disponível para pesquisa, tais como: artigos e dissertações publicadas, assim como fóruns de discussões em volta de banco de dados móveis.

4.1 SQLITE

O SQLite é um Banco de Dados Open Source e relacional, sendo uma influente biblioteca de banco de dados baseado em SQL (Structured Query Language) que tem como função ser parecido com um gerenciamento de banco de dados, no qual alguma das suas funções é manter o controle de diversos bancos de dados e

simultaneamente suas tabelas. O desenvolvedor cria o banco de dados e as tabelas, assim como manipula seus dados através dos comandos DDL (Data Definition Language) e DML (Data Manipulation Language), que são comando do SQL Padrão. Toda o serviço de persistência de dados é fornecido pelo SQLite. Toda a estrutura do banco de dados fica armazenado junto com a aplicação em um único arquivo, no qual possui extensão “.db”. A base de dados pode ter tamanho superior a 2 *tera-bytes*. É um gerenciador de banco de dados “auto-suficiente”, ou seja, independente de uma estrutura cliente-servidor. Neste caso ele necessita de um apoio mínimo de bibliotecas externas ou até mesmo do Sistema Operacional, se tornando muito fácil para ser adaptado para o uso de dispositivos embarcados. Isto o torna flexível para muitas plataformas (Lecheta, 2013).

4.2 DB4OBJECTS

O DB4Objects é um banco de dados orientado a objetos criado a princípio para a plataforma Java, podemos encontrar no mercado uma versão disponível também para .NET. Não existe necessidade de instalar e configurar um servidor de banco de dados para esse banco. O tamanho de utilização de memória desse banco é de 400kb, com isso o DB4Objects diminui consideravelmente o uso de recurso computacional. Com a utilização de um banco de dados orientado a objetos exclui o uso de ferramenta e códigos para mapear o objeto relacional, os quais levam ao aumento da complexidade da manutenção do código.

As formas possíveis de obter uma conexão entre a aplicação e o Db4Objects são:

- Conexão Direta, no caso de aplicações embarcadas;
- Cliente / Servidor: A forma padrão de conexão com um SGBD;
- Cliente / Servidor Embutido: Não há necessidade de um servidor externo, bastando instanciar um servidor na maquina de aplicação.

Essa interface fornece métodos para armazenar, consultar e excluir objetos e representa uma conexão com um servidor ou um banco de dados local (Santana, 2010).

4.3 NEODATIS

O Neodatis é um banco de dados móvel orientado objeto, o armazenamento e a recuperação dos dados são realizados apenas com uma única linha de código sem a necessidade de mapear seus objetos para alguma tabela. NeoDatis suporta transações ACID para garantir a integridade do banco de dados de dados. É suportada na Plataforma Java e esta sendo implantado para .Net, é uma ferramenta *Open Source*. Seu funcionamento pode ser como um banco de dados embutido no aplicativo ou em um modo tradicional *Client Server* (Jack, 2007).

5ANÁLISE COMPARATIVA ENTRE OS BANCOS DE DADOS MÓVEIS

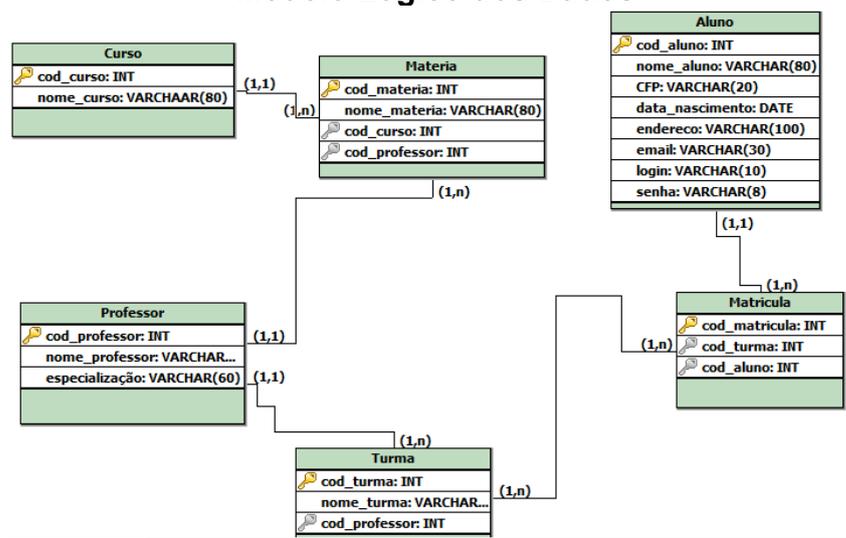
A proposta deste artigo é fazer um estudo avaliativo através da comparação dos bancos de dados citados, visando uma análise de desempenho e funcionalidades dos mesmos. Objetiva-se, portanto, com esse estudo ajudar aos engenheiros e desenvolvedores de aplicações móveis na tomada de decisão em problemas diversos dentro de um projeto, tais como agilidade nos processos, estrutura, facilidade de implementação, menor custo de ferramentas, etc. Para esta comparação, foram utilizadas as últimas versões dos bancos de dados móveis:

- SQLite 3.4
- Neodatis 1.9
- DB4Objects 8.0

Tendo como base Pires;Nascimento;Salgado (2007) e Colares (2007) foram definidos os critérios de comparação presentes nesse artigo. Primeiramente foi utilizada uma ferramenta para testar o desempenho dos bancos analisados, o *Benchmark*. Um *benchmark* é definido, segundo o dicionário *Webster* (Houassis, 2003), como sendo “*um padrão para medida ou avaliação*”. As métricas analisadas medem agilidade (em qual velocidade uma determina ação foi executada) ou vazão (quantidade de execução por tempo). Também será feita uma análise das funcionalidades que o desenvolvedor se depara diariamente, que podem ou não ajudar na concretização da aplicação com mais facilidade. Com isso, as abordagens dos critérios vão ser estabelecidas de características que envolvem: suporte a *stored procedures*; suporte a *triggers*; suporte a *Domains*; suporte a *Cursor*; Exportação XML; Tipo de dados para moeda; *Views*; Transação; Tipos de bloqueio; Ferramentas; Portabilidade. Para dar suporte a realização deste estudo, foi criado um projeto Android, na plata-

forma Eclipse, na linguagem Java. O objetivo dessa aplicação era realizar o procedimento de matrícula de um aluno a um curso disponível num sistema para cursos técnicos. Segue a estrutura de tabela dos bancos de dados implantados nesse artigo, de modo que todos os bancos móveis analisados apresentam a mesma estrutura, conforme figura 2:

Figura 2
Modelo Lógico dos Dados



Fonte: Do autor

Os recursos analisados serão transcritos nas próximas seções, com um quadro comparativo entre os bancos analisados.

5.1 Stored Procedures

São procedimentos nos quais vários comandos são criados, ou seja, sub-rotinas, de forma a ser reutilizado em momentos necessários. São guardados no servidor a fim de serem reutilizadas (Caserta, 2002).

Tabela 1
Stored Procedures

SBD	Suporta	Comentários
SQLite	Não	Banco embarcado, não há necessidade de reutilização de procedimentos.
Neodatis	Não	Banco embarcado e orientado a objeto.
DB4Objects	Não	Banco embarcado e orientado a objeto.

Como todos os bancos utilizados são embarcados, ou seja, não há conexão com o Servidor, faz com que, a implementação de *Stored Procedures* no seu código, não seja necessária, conforme tabela 1. E também, o DB4Objects e o Neodatis são bancos de dados orientados a objeto, os procedimentos são armazenados no objeto da classe, fazendo com que *stored procedures* seja desnecessária.

5.2 Triggers

Uma *trigger* é uma ação na qual alguma instrução (*delete, insert, update*) do banco for disparada. Evita a inconsistência dos dados (Colares; Martins, 2007).

Tabela 2
Triggers

SBD	Suporta	Comentários
SQLite	Sim	É suportada para cada gatilho de linha, não para cada gatilho de instrução.
Neodatis	Sim	Utiliza <i>triggerem</i> seus métodos.
DB4Objects	Sim	Utiliza <i>triggerem</i> seus métodos.

De acordo com a tabela 2, pode-se concluir que em todos os bancos de dados móveis analisados podem utilizar *triggers* como funcionalidades. No caso do SQLite a *trigger* não tem a mesma função que os demais bancos, seu funcionamento é de acordo com uma cláusula (*When*), que implica que as instruções que estão na *trigger* pode ser executada para cada linha do banco de dados que está sendo inserido, atualizado ou excluído.

5.3 Cursor

Em vez de executar uma consulta inteira de uma só vez, é possível configurar um *cursor* que encapsula consulta e permite, então, ler o resultado em poucas linhas por vez (Vieira, 2009).

O *Cursor* armazena o retorno de uma consulta em todos os bancos analisados, com isso se faz necessário criar uma função para pegar os dados do *Cursor* e

criar um Objeto para ser utilizada na aplicação, isto no caso de aplicações Orientada a objeto, como é o caso analisado. De acordo com a tabela 3, todos os bancos analisados suportam essa funcionalidade.

Tabela 3
Cursor

SBD	Suporta	Comentários
SQLite	Sim	Objeto do <i>cursor</i> é utilizado na aplicação.
Neodatis	Sim	Objeto do <i>cursor</i> é utilizado na aplicação.
DB4Objects	Sim	Objeto do <i>cursor</i> é utilizado na aplicação.

5.4 Exportação XML

Em algumas aplicações, existe a necessidade de exportar os dados armazenados no banco de dados para arquivos XML, de forma a trocar as informações dinamicamente.

Tabela 4
Exportação XML

SBD	Suporta	Comentários
SQLite	Sim	Exportação realizada por função.
Neodatis	Sim	Exportação realizada por métodos.
DB4Objects	Não	Não há exportação XML.

No DB4Objects, conforme tabela 4, não foi implantada uma biblioteca para a exportação de arquivo XML, todos os outros bancos móveis suportam essa exportação.

5.5 Tipos de dados para moeda

O tipo de moeda determina o tamanho do armazenamento da informação, com isso os tipos podem ser arredondados seguindo regras aritméticas do mesmo. Esse arredondamento, que pode acontecer ou não, acarreta problemas sérios. Co-

mo, por exemplo, um centavo em mil pedidos de compra ou dez centavos em cem transferências bancárias.

Tabela 5
Tipos de dados de moeda

SBD	Suporta	Comentários
SQLite	Sim	Real, armazenado em 8 bytes
Neodatis	Sim	Float, 32 bites e Double 4 bits
DB4Objects	Sim	Float, 32 bites e Double 4 bits

Ao fazer a declaração do tipo de uma variável no SQLite, o tipo da moeda pode ser Float ou Double porém, durante a execução do comando o SQLite converte esses tipos para sua classe de armazenamento. Essa classe de armazenamento só aceita Real para valores de moeda. No Neodatis e no DB4Objects, por serem bancos orientados a objetos o tipo de dados dos atributos das classes serão implementados no objeto do banco, conforme tabela 5.

5.6 Domains

Um domínio é criado a partir de um conjunto de restrições, ou seja, ao criar uma tabela, ou um campo de uma tabela em um domínio todas as características do domínio são atribuídas para a tabela. Domínios são utilizados para sistemas muito grandes (Colares, 2007).

Tabela 6
Domains

SBD	Suporta	Comentários
SQLite	Não	Tamanho Pequeno.
Neodatis	Não	Tamanho Pequeno.
DB4Objects	Não	Tamanho Pequeno.

Os bancos analisados, de acordo com a tabela 6, por não serem bancos de dados com um tamanho muito elevado, não tem suporte da estrutura de domínios.

5.7 Views

Views funcionam como tabelas virtuais, na qual é formada uma tabela com um conjunto de resultados de uma ou mais tabelas (Jaruzo, 2010).

Tabela 7
Views

SBD	Suporta	Comentários
SQLite	Sim	Somente Leitura.
Neodatis	Não	Utiliza os métodos para formar o objeto.
DB4Objects	Não	Utiliza os métodos para formar o objeto.

No SQLite pode ser incluídas *views*, porém, elas servem apenas para leitura, ou seja, não há como inserirmos, atualizarmos ou excluirmos dados direto na *view*. Nos outros dois bancos as *views* não são suportadas, conforme tabela 7, já que a finalidade do banco orientado a objeto é apenas para a persistência de dados, não havendo necessidade de “tabelas virtuais”.

5.8 Transação

Uma transação é um conjunto de ações que devem ser realizadas com sucesso em tabelas, antes que essa ação seja feita permanentemente. Serve para manter a integridade do banco de dados.

Tabela 8
Transação

SBD	Suporta	Comentários
SQLite	Sim	Instrução executa.
Neodatis	Sim	Instância o objeto para a persistência.
DB4Objects	Sim	Instância o objeto para a persistência.

As transações em bancos orientados são abertas quando se instancia o objeto para o banco de dados. No SQLite, a transação é aberta quando um conjunto de instruções é executada, de acordo com a tabela 8.

5.9 Tipos de Bloqueio

O bloqueio em banco de dados tem como função: impedir que duas transações simultâneas acessem o mesmo dado ou recurso ao mesmo tempo. Nos bancos citados na tabela 9, existem funções ou métodos que bloqueiam a informação até que o comando tenha sido executado com sucesso, evitando assim, que outros comandos interfiram na ação desejada.

Tabela 9
Tipos de Bloqueio

SBD	Suporta	Comentários
SQLite	Sim	Funções.
Neodatis	Sim	Métodos.
DB4Objects	Sim	Métodos.

5.10 Ferramentas de Administração e Gerência

Bancos de dados móveis possuem uma ampla coleção de ferramentas. Para a comparação neste documento, iremos tomar como base, as ferramentas de manutenção e administração dos bancos, as quais são as mais citadas em listas de discussões sobre aplicações móveis.

Tabela 10
Ferramenta

SBD	Suporta	Comentários
SQLite	Sim	SQLite Express Personal e SQLite Plus
Neodatis	Não	Nãopossuiferramenta.
DB4Objects	Não	Nãopossuiferramenta.

O SQLite Express Personal e o SQLite Plus são ferramentas na qual há possibilidade de visualizar todos os bancos de dados existentes assim como gerenciar os campos, índices, *views* e *triggers*. Sua interface é de simples e intuitiva. Nos outros dois bancos de dados não há ferramenta específica de manutenção e administração dos dados, de acordo com a tabela 10, já que a manutenção e a administração são feitas pelos métodos.

5.11 Portabilidade

Para o desenvolvimento de aplicativos móveis a empresa precisa saber qual a plataforma que o mesmo irá funcionar. Portanto, os bancos de dados móveis precisam suportar essa plataforma. A empresa que desenvolve aplicações móveis cria toda sua estrutura para o desenvolvimento e com isso os bancos de dados precisam se adaptar a essas condições. Por isso a portabilidade dos bancos móveis é um assunto de extrema importância.

Tabela 11
Portabilidade

SBD	Plataformas
SQLite	Blackberry 10 OS;Windows Phone 8;IOS;Android
Neodatis	Android
DB4Objects	Windows Mobile;Android

O SQLite é suportado em mais plataformas devido a sua facilidade de implementação e materiais de suporte para a implantação, conforme tabela 11. Nos bancos orientados a objeto são suportadas em menos plataformas já que suas funcionalidades ainda são um pouco mais escassas.

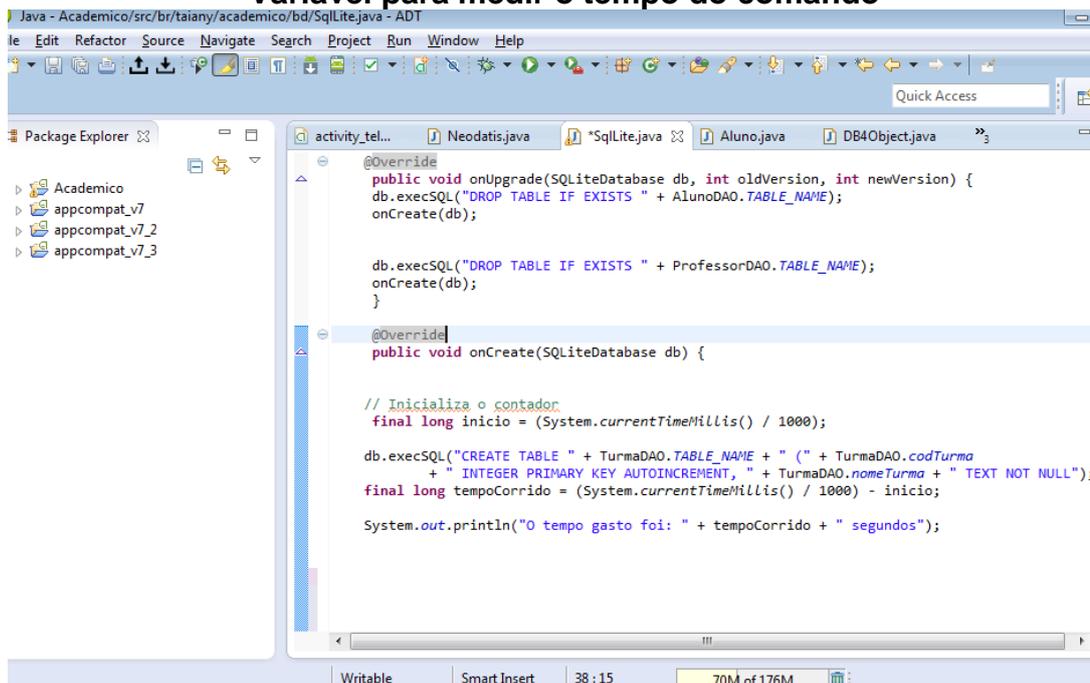
5.12 Análise de Desempenho (*Benchmark*)

No Android existem *Benchmarks* que medem o desempenho do sistema operacional, alguns exemplos são: AnTuTU Benchmark, Vellamo Benchmark, Quadrant Benchmark entre outros, disponíveis no site oficial da Google Play(Google,2014).

O SQLite disponibilizou uma versão do seu *Benchmark* no Google Play. Esse aplicativo gera uma comparação do desempenho do seu aparelho e versão Android, com os outros modelos e versões disponíveis no mercado. A aplicação dos testes a seguir foi baseada nos padrões dos testes do *Benchmark*, no qual é dividida por módulos, que podem ser executados em conjunto ou separadamente, no caso desse artigo iremos adotar a execução dos testes individualmente. Os módulos

que utilizaremos são: Carga e Estrutura, Seleções, Junções, Projeções, Agregações, Atualizações. Para cada módulo há vários setores, os quais não irão ser citados aqui, de forma a testarmos o desempenho específico. Para a medição das métricas foi criada em cada implementação dos bancos um contador, no qual antes de executar o comando é inicializado e após a execução com sucesso retorna a variável em tempo, conforme o exemplo da figura 3:

Figura 3
Variável para medir o tempo do comando



Fonte: Do autor

5.12.1 Carga e Estrutura:

Verifica o tempo de criação da estrutura dos bancos de dados, assim como a inserção dos dados.

Tabela 12
Carga e Estrutura

	SQLite	Neodatis	DB4Objects
Criação de Tabelas	0,13s	0,30s	0,34s
Carga Tabelas	0,47s	0,50s	0,61s

5.12.2 Seleções:

Esse módulo verifica o poder dos bancos de dados de utilizarem índices em consultas. Os bancos de dados móveis precisam escolher corretamente se irão utilizar índices ou fazer a leitura completa da tabela. Baseia-se no operador SQL *Select*.

Tabela 13
Seleções

	SQLite	Neodatis	DB4Objects
Seleções de dados	0,3s	0,5s	0,3s

5.12.3 Junções e Projeções:

Nesse teste verifica a eficiência das consultas na junção de duas, três ou quatro tabelas. Em termos de Projeções, os testes baseiam-se no operador SQL *Distinct*. Neste caso as linhas que são duplicadas, viram apenas uma linha.

Tabela 14
Junções e Projeções

	SQLite	Neodatis	DB4Objects
Junções Tabelas	0,4s	0,8s	0,7s
Projeções dados	0,3s	0,5s	0,6s

5.12.4 Agregações e Atualizações:

O módulo de Agregações usa como base o operador SQL *Group By* para agregações, no qual agrupa os dados de acordo com uma coluna. Os teste para Atualizações verificam a integridade do banco e o desempenho das operações *insert, update, delete*.

Tabela 15
Agregações e Atualizações

	SQLite	Neodatis	DB4Objects
Agregações dados	0,2s	0,4s	0,6s
<i>Delete</i>	0,2s	0,4s	0,5s
<i>Insert</i>	0,1s	0,2s	0,1s
<i>Update</i>	0,2s	0,3s	0,3s

6 CONCLUSÃO

No cenário analisado, foi concluído que devido o SQLite ter comandos e funções herdadas de uma linguagem de banco de dados muito utilizada, o SQL, assim como sua fácil implementação e gerenciamentos dos dados, desencadeou uma maior aceitação nos ambientes móveis, e principalmente na plataforma Android, a qual foi utilizada nesse artigo. Há uma API na qual a própria aplicação se encarrega de criar e controlar a versão do banco de dados. A variedade de documentos encontrados sobre esse banco de dados é consideravelmente maior do que para os demais citados nesse artigo. Nos testes apresentados nesse estudo, foi verificado o maior desempenho do SQLite na maioria dos casos analisados, de acordo com a tabela 16, assim como há um maior suporte as principais funcionalidades que um desenvolvedor móvel precisa ter acesso, conforme tabela 17, de forma a facilitar o implementação do banco na aplicação.

Tabela 16
Quadro comparativo Desempenho

BD	Desempenho						
	Criação tabelas	Carga Tabelas	Seleções	Junções	Projeções	Agregações	Atualizações (Soma: Insert + Delete + Update)
SQLite	0,13s	0,47s	0,3s	0,4s	0,3s	0,2s	0,5s
DB4Objects	0,30s	0,5s	0,5s	0,8s	0,5s	0,4s	0,9s
Neodatis	0,34s	0,61s	0,3s	0,7s	0,6s	0,6s	0,9s

Tabela 17
Quadro Comparativo Funcionalidades

BD	Funcionalidades						
	Stored Procedures e Domains	Exportação XML	Tipos de Dados de moeda, Triggers e Cursor	Views	Tipos de Bloqueio e Transação	Ferramenta	Portabilidade
SQLite	Não suporta	Suporta	Suporta	Suporta	Suporta	Suporta	4 Plataformas
DB4Objects	Não suporta	Suporta	Suporta	Não suporta	Suporta	Não suporta	1 Plataforma
Neodatis	Não suporta	Não suporta	Suporta	Não suporta	Suporta	Não suporta	2 Plataformas

Na programação para Android, o SQLite já vem embutido no pacote de desenvolvimento, fazendo assim com que a única preocupação do programador seja herdar as classes do SQLite.

Nos demais Sistemas de Banco de Dados estudados, há a necessidade de importação das extensões para o seu projeto. Por serem Orientado a Objetos, conseguem captar a informação na forma de objetos e com isso, a manipulação é feita apenas por métodos definidos pela classe que define o objeto. Com isso o conceito de gerenciamento de banco de dados acaba se perdendo gradativamente, já que a gerência dos bancos é feita pelos métodos. Dessa forma, algumas funções básicas do conceito de Banco de Dados acabam se perdendo nos conceitos dos bancos Orientado a objetos, tais como: *views*, ferramenta de administração, ferramentas de relatório e padrões de recuperação. SGBDOO ainda estão sendo muito analisados e testados no mercado atual, com vários testes e funcionalidades novas sendo criadas, de forma a atender todo o cenário móvel.

Para a melhor adequação dos bancos de dados móveis nas aplicações, se faz necessária a implantação de melhores funcionalidades e ferramentas disponíveis em bancos de dados tradicionais (tais como o SQL Server, MySQL, Oracle, dentre outros). Exemplos de funcionalidades que precisam ser adicionadas aos BD móveis seriam implementação de *Stored Procedures*, criação de chaves estrangeiras para manter a consistência dos dados, serviços de relatório, criação de tabelas temporárias, variável temporária, ferramentas de criação de diagramas das tabelas.

Em pesquisas futuras, pode-se verificar quais desses pontos levantados anteriormente foram adequados aos bancos de dados móveis, permitindo assim que o cenário móvel de persistência dos dados se torne mais amplo e com uma carga de recursos maiores.

REFERÊNCIAS

BASTOS, Bruno Pinheiro; JERÔNIMO, Eduardo Bonfim. Sistemas Móveis – Parte 04. Publicação Artigo: **DevMedia – Mobile Magazine**, Maio.2009.

CAMARGO, Murilo Silva. Um estudo dos principais modelos de transações em Banco de Dados Móveis e uma proposta diferenciada do modelo Pro-motion. Dissertação Publicada: **Universidade Federal de Santa Catarina**. Florianópolis, Outubro.2003.

CASERTA, Thiago. Dominando Stored Procedures. Publicação Artigo: **DevMedia – Mobile Magazine**, 2012.

COLARES, Flávio Martins. Análise Comparativa de Bancos de Dados Gratuitos. Publicação Monografia: **Faculdade Lourenço Filho**, Fortaleza. 2007.

CONCEIÇÃO, Marcos Flávio Rocha. Banco de Dados Móveis. Publicação Artigo: **Universidade Presidente Antônio Carlos**, Barbacena. 2005.

CORTÊS, Sérgio da C.;LIFSCHITZ, Sérgio. Banco de dados para um ambiente de computação móvel. **XXIII Congresso da Sociedade Brasileira de Computação – SBC**. 2003.

ELMASRI, Ramez; NAVATHE, Shamkant B. Fundamentals of Database Systems Tradução de Marília Guimarães Pinheiro. Revisão Técnica de Luis Ricardo de Figueiredo, São Paulo. 2005.

FIGUEIREDO, Carlos Maurício Seródio. Introdução à Computação Móvel. Disponível em: <<http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/palestras/palestra3.pdf>>. Acesso em: 24 de Fevereiro de 2014.

GALANTE, Alan Carvalho; MOREIRA, Elvis Leonardo Rangel; BRANDÃO, Flávio Camilo. Banco de Dados Orientado a Objetos: Uma Realidade. Publicação Artigo: **Faculdade Salesiana Maria Auxiliadora**, Macaé. 1999.

GALLIANO, Eduardo. Bancos de Dados Móveis. Publicação artigo: **Centro de Informática da Universidade Federal de Pernambuco**, Londrina. 2007.

GONÇALVES, Luciana Siqueira. Banco de Dados Móveis – Um estudo introdutório. Defesa Mestrado: **Centro de Ensino Superior – CES**. Juiz de Fora. 2006.

GOOGLE. Google Play. Disponível em: <<https://play.google.com>>. Acesso em 15 de Maio de 2014.

HOUASSIS, Antônio. Mini Webster's Dicionário Inglês/Português. **Editora Record**, Rio de Janeiro. 2003.

ITO, Giani C. Bancos de dados móveis: uma análise de soluções propostas para gerenciamento de dados. Dissertação Mestrado: **Universidade Federal de Santa Catarina**, Florianópolis. 2001

JACK, Eti. Informação e Conteúdo sobre Ti: Conheça o NeoDatis ODB. Disponível em: <<http://www.jack.eti.br/conheca-o-neodatis-odb/>>. Acesso em: 22 de Maio de 2014.

JARUZO, Regis. View (Visão). Disponível em: <http://fortium.edu.br/blog/regis_jaruzo/files/2010/11/View.pdf>. Acesso em: 13 de Maio de 2014.

LECHETA, Ricardo R. Google Android: Aprenda a criar aplicações para dispositivos móveis com Android SDK. **Editora Novatec. Ed: 3.** São Paulo, Março. 2013.

PIRES, Carlos Eduardo Santos; NASCIMENTO, Oscar Rilson; SALGADO, Ana Carolina. Comparativo de Desempenho entre Bancos de Dados de Código Aberto. Publicação Artigo: **Universidade Federal de Pernambuco**, 2007. Recife.

PRACIANO, Elias. O que é um banco de dados Relacional? Disponível em: <<http://elias.praciano.com/2013/09/o-que-e-um-banco-de-dados-relacional/>>. Acesso em: 18 de Junho de 2014.

RAINONE, Flávia. Banco de Dados Móveis. Disponível em: <grenoble.ime.usp.br/movel/bancos_moveis.ppt>. Acesso em: 20 de Março de 2014.

SANTANA, Magno. Artigo webMobile 28 - Mobilidade com DB4Objects. Publicação Artigo: **DevMedia – webMobile 28**, 2010.

VIEIRA, Sidney. Tecnologia de Banco de Dados – Cursores. Disponível em: <http://www.sidneyvieira.kinghost.net/abas/disciplinas/download/ES_Tecnologia_BD_Cursores.pdf>. Acesso em 24 de Maio de 2014.