

FACULDADE DE CIÊNCIAS APLICADAS “SAGRADO CORAÇÃO”

DIRETORIA DE ENSINO SUPERIOR

COLEGIADO DE SISTEMAS DE INFORMAÇÃO

GUSTAVO ANDRÉ DE FREITAS

RILIANE ALPOIM PARIS

FIREBIRD

LINHARES

2007

GUSTAVO ANDRÉ DE FREITAS
RILIANE ALPOIM PARIS

FIREBIRD

Trabalho Acadêmico do Curso de Bacharelado em Sistemas de Informação da Faculdade de Ciências Aplicadas "Sagrado Coração" - UNILINHARES, apresentado como requisito para avaliação em seminário interdisciplinar.
Orientador: Prof. Rodrigo Smarzaró.

LINHARES
2007

SUMÁRIO

1	INTRODUÇÃO	5
2	CONCEITOS E DEFINIÇÕES	6
2.1	SGBD	6
2.2	STORED PROCEDURES	7
2.3	TRIGGERS	8
2.4	VIEWS	9
3	HISTÓRIA	10
3.1	VERSÃO 1.0	10
3.2	VERSÃO 1.5	11
3.3	VERSÃO 2.0	12
3.4	LINGUAGEM PSQL	13
3.5	O FUTURO DO FIREBIRD	14
3.6	TIPOS DE DADOS	15
3.7	EMPRESAS QUE CONFIAM NO FIREBIRD	17
3.8	VERSÕES DO SERVIDOR	18
3.8.1	Classic	18
3.8.2	SuperServer	19
3.8.3	Embedded	19
4	CARACTERÍSTICAS	20
4.1	FERRAMENTAS	20
4.2	COMPONENTES	21
4.3	PRINCIPAIS RECURSOS	22
4.3.1	Sistema Multi-Geracional	22
4.3.2	Replicação	23
4.3.3	Sombreamento	23
4.3.4	Suporte a Diversos Protocolos de Conexão	24
4.3.5	UDF	25

4.3.6	Instalação Reduzida	26
4.3.7	Tamanho do Banco de Dados	27
4.3.8	Backup On-line	27
4.4	VANTAGENS	29
4.4.1	Suporte	29
4.4.2	Portabilidade	29
4.4.3	Open Source	30
4.5	LIMITAÇÕES	31
4.5.1	Segurança	31
4.5.1.1	Usuário SYSDBA	31
4.5.1.2	Criptografia	32
4.5.1.3	Ataques de Força Bruta	33
4.5.2	Número Máximo de Clientes ou Conexões Simultâneas	33
4.5.3	Interface para Administração do Banco	34
5	ESTUDO DE CASO	35
6	CONCLUSÃO	36
6.1	VALE A PENA UTILIZAR?	36
7	REFERÊNCIAS	38

1 INTRODUÇÃO

Na era da informação, seja nas organizações, na medicina ou na educação a informação é o bem mais valioso. Os bancos de dados vieram suprir a necessidade de otimização, armazenamento e recuperação dos dados. Na busca por maior segurança dos dados, surgem os sistemas gerenciadores de bancos de dados (SGBDs).

O Firebird é o SGBD apresentado nesse trabalho, sendo um sistema gerenciador de banco de dados gratuito e open source desenvolvido pela *FirebirdSQL Foundation*, uma organização sem fins lucrativos que visa o aperfeiçoamento e continuação do seu desenvolvimento.

No presente trabalho a proposta não foi desenvolver um banco de dados, mas sim entender o funcionamento do sistema de gerenciamento do mesmo, no caso o Firebird, vislumbrando a história de surgimento, as ferramentas, os principais recursos e limitações, e no final apresentamos um estudo de caso real.

2. CONCEITOS E DEFINIÇÕES

2.1 SGBD

“[...] imagine que a principal finalidade do computador é armazenar dados, que possam ser acessados a qualquer momento e em qualquer lugar do mundo. Quanto maior o número de computadores, maior o número de dados a serem gerenciados e processados. Como a tendência é que esse número aumente a cada ano, a complexidade dos dados armazenados deverá ser um grande problema na área de informática.” (MEDEIROS, 2006, p.4).

Com o dinamismo da evolução tecnológica, os computadores possuem um custo mais acessível, o que acarreta um grande aumento de pessoas e empresas que adotam o computador para a resolução de problemas do seu dia-a-dia. Visto que uma das principais, senão a maior, finalidade do computador é o armazenamento de dados, há uma preocupação para que a utilização desses dados armazenados seja feita da forma mais eficaz possível, considerando que pode haver um grande número de usuários interagindo com esses dados ao mesmo tempo.

Diante da necessidade de melhorias no armazenamento e organização dos dados no computador, surgiram os bancos de dados, que são softwares que se aplicam à otimização dessa tarefa, sendo utilizados especificamente na área de banco de dados.

“[...] um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros; em outras palavras, é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar.” (DATE, 2003, p.6).

A utilidade dos bancos de dados é a de proporcionar uma melhor organização e armazenamento dos dados, além de oferecer ao usuário uma melhor facilidade na obtenção das informações ali contidas, visto que estarão num mesmo lugar de armazenamento, possibilitando um melhor aproveitamento do computador no que diz respeito à otimização do espaço de armazenamento, com maior agilidade na busca e atualização dos dados.

Os bancos de dados pioneiros trabalhavam com registros de estruturas semelhantes, por exemplo, em uma organização onde informações similares seriam mantidas para cada funcionário, cada setor, cada registro de salário, etc. Os bancos de dados atuais trabalham com relacionamentos entre dados, buscando a não redundância dos mesmos.

“Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre usuários e aplicações.” (ELMASRI, 2005, p.4).

O sistema gerenciador de banco de dados é um software que se situa entre o usuário e o banco de dados físico do computador. Sua principal finalidade é facilitar a construção e utilização de bancos de dados, separando o usuário dos detalhes em nível de hardware, isto é, ocultando ao usuário os detalhes de armazenamento dos dados no hardware, e oferecendo somente uma visão em um nível cliente.

Como o próprio nome diz, o SGBD é o gestor das atividades relacionadas ao banco de dados, implicando na definição de que estrutura de armazenamento das informações será utilizada e de que forma essas informações serão fornecidas ao usuário. Outra característica do SGBD é a de proporcionar segurança contra eventuais problemas do sistema e tentativas de acesso não autorizado.

Alguns SGBDs utilizados atualmente são Oracle, DB2, Interbase e SQL Server, sistemas gerenciadores de banco de dados comerciais, e na versão *open source* o MySQL, o PostgreSQL e o Firebird sendo este o tema abordado neste trabalho.

2.2 STORED PROCEDURES

São procedimentos selecionáveis que possuem uma ou mais SQL *statements* para realizar cálculos e fazer transformações em determinadas linhas no lado servidor, um tanto úteis quando se quer obter um determinado valor de uma tabela que seria impossível obter numa instrução SELECT convencional. (MECENAS, 2006).

```

/*exemplo de criação de uma stored procedure*/
CREATE PROCEDURE FU_ABS(
    WVAR DOUBLE PRECISION)
    RETURNS (
    R_ABS DOUBLE PRECISION)
    AS
    begin
    /*
    devolve o valor Absoluto de um número.
    -123,47 ou 123,47 devolve sempre 123,47
    */
    if (:wvar is not null) then begin
    if (:wvar < 0.0000) then
    r_abs = :wvar * -1;
    else
    r_abs = :wvar;
    end
    suspend;
    end

```

Figura 2.1 – exemplo de criação de uma stored procedure
 Fonte: www.firebase.com.br

2.3 TRIGGERS

“Triggers são blocos de comandos que são executados automaticamente quando ocorre algum evento associado a uma tabela que satisfaça uma condição de INSERT, UPDATE ou DELETE.” (SILVA, 2006, p. 9).

No Firebird pode haver vários triggers para cada tabela, sendo que são disparados antes ou depois das operações de Updates, Deletes ou Inserts.

```

/* exemplo de criação de um trigger */
CREATE TRIGGER TableView_Delete FOR TableView BEFORE DELETE AS
BEGIN
DELETE FROM Table1
WHERE ColA = OLD.ColA;
DELETE FROM Table2
WHERE ColA = OLD.ColA;
END;

```

Figura 2.2 – exemplo de criação de um trigger
 Fonte: www.firebase.com.br

2.4 VIEWS

São tabelas criadas a partir de outras tabelas, sendo que estas derivadas são chamadas de visões porque não existem fisicamente, são tabelas virtuais. Quando se tem a necessidade de consultar uma tabela constantemente que contenha dados de outras três tabelas, por exemplo, isso é facilitado pelas visões, onde se pode agregar em uma única tabela virtual os dados das outras três, sem nenhuma limitação para consultas.

```
/* exemplo de criação de uma view */  
CREATE VIEW TableView AS  
SELECT Table1.ColA, Table1.ColB, Table2.ColC  
FROM Table1, Table2  
WHERE Table1.ColA = Table2.ColA;
```

Figura 2.3 – exemplo de criação de uma view
Fonte: www.firebaseio.com.br

3. HISTÓRIA

O Firebird é um sistema gerenciador de banco de dados que surgiu a partir do Interbase 6. A proposta da Borland iniciou-se em 2000 quando foi aberto o código do Interbase, porém para que somente fossem feitas sugestões, sem que nenhuma alteração do código pudesse ser feita por alguém que não fosse interno à companhia. Isso trouxe bastante descontentamento aos desenvolvedores que em julho de 2000 criaram a Firebird Tree no Source Forge¹, para que houvesse um espaço aberto a todos.

Em 2001 a Borland volta com versões comerciais, tendo então duas versões do Interbase 6, a *open source* e a comercial. A primeira versão do Firebird foi lançada em março de 2002 e é licenciado sob a IPL (InterBase Public License), a qual tem os mesmos termos da Mozilla Public License 1.1, e é totalmente compatível com o padrão ANSI SQL-92.

“Utiliza-se do padrão ANSI SQL-92 para prover a manipulação dos bancos de dados e algumas funções que complementam tal padrão, bem como a possibilidade de customização.” (FREITAS, 2007).

3.1 VERSÃO 1.0

Essa versão é praticamente igual a versão *open source* do Interbase 6, com linguagem de código em C, usando, por exemplo, os mesmos nomes dos arquivos de extensão, chaves de registro no Windows, entre outros, porém com muita correção de *bugs* e documentação daqueles que ainda não foram corrigidos.

¹ “Sourceforge.net é o maior website do mundo voltado ao desenvolvimento de software de código aberto, hospedando mais de 100.000 projetos e mais de 1.000.000 de usuários registrados com um recurso centralizado de gerenciamento de projetos, comunicações e código.” (UNDER-LINUX.ORG, 2006).

3.2 VERSÃO 1.5

A partir da versão 1.5 o Firebird distanciou-se ainda mais do Interbase. Quando começou a ser desenvolvido era apenas para consolidar o SGBD com a conversão da linguagem de código, que era em C, e a partir desta versão passou a ser C++, porém foram tantas as implementações que ela acabou se tornando uma versão maior do SGBD.

Algumas das inovações foram:

- Mudanças nos nomes dos arquivos de extensão, de .gdb para .fdb, desvinculando ainda mais o Firebird do Interbase.
- Comandos CASE, COALESCE e NULLIF. O CASE permite retornar um valor de acordo com as condições definidas e, a partir dele, foi possível implementar a COALESCE, comando que retorna o primeiro valor não nulo de uma seqüência de valores passados como parâmetros, e o NULLIF que é um comando que recebe dois valores de entrada, se os dois valores forem iguais ele retorna NULL, se forem diferentes ele retorna o valor do primeiro parâmetro.
- SQLs dinâmicos, um recurso que permite inserir e executar dinamicamente comandos SQL através dos comandos *Execute Statement* e *For Execute Statement*, dentro de *Stored Procedures* e *Triggers*.
- Versão *Classic* do servidor para Windows.
- Versão *Embedded*, ou embarcada, que é uma variação do servidor onde se tem um servidor completo Firebird composto por apenas alguns arquivos.
- Disponibilidade de rodar muitos servidores simultaneamente na mesma máquina.
- *Triggers* Universais, possibilidade de associar um único *trigger* a vários eventos.
- Travamento pessimista, a arquitetura *versioning* do Firebird permite que seja praticamente desnecessária a utilização de travamentos pessimistas nas aplicações. A versão 1.5 tem a implementação no select de duas novas cláusulas, a *WITH LOCK* e *FOR UPDATE*, que têm a função de evitar que

outras transações possam alterar os registros selecionados enquanto permanecem bloqueados.

- Apelidos (alias) de banco de dados, que permite a utilização de apelidos atribuídos ao BD nos *strings* de conexão, proporcionado pela implementação de um arquivo, *aliases.conf*, que se encontra no diretório de instalação do Firebird.

3.3 VERSÃO 2.0

Com uma plataforma de código mais robusta, a versão 2.0 do Firebird foi lançada durante a quarta conferência do Firebird em Praga, na República Tcheca, em 12 de novembro de 2006, trazendo tranquilidade à comunidade de desenvolvedores que o utilizam, demonstrando que veio para ficar, gratuito e com código aberto.

“A versão 2.0[...] é o que podemos chamar de fase madura do produto. Melhorias em performance, novas funcionalidades e novos recursos, com certeza já entrou para a história da informática e o seu futuro é muito promissor.” (BUBLITZ, 2007, p. 50).

Isso mostra que a *FirebirdSQL Foundation* não está parada, havendo uma movimentação intensa para que o Firebird evolua para um banco de dados melhor e mais confiável, o que já é vislumbrado nos aperfeiçoamentos contidos nesta versão, que resultou em melhorias na *performance* e segurança.

Algumas de suas características são:

- Índices com expressões, onde o valor a ser definido é o resultado de expressões arbitrárias, onde pode haver, inclusive, chamadas a UDFs.
- Minimização de limitações das versões anteriores no que se refere ao tamanho dos índices, que era de 252 bytes e tamanho das tabelas, que era de 30 Gb.
- O comando EXECUTE BLOCK, que permite que sejam criadas rotinas fora de *triggers* e *stored procedures*, sem a necessidade de criar uma *procedure*.

“[...] o EXECUTE BLOCK pode fazer qualquer tarefa que uma *procedure* executável poderia fazer menos chamar a si mesmo (recursividade).” (CANTU, 2006, p. 197)

- Suporte a plataforma 64 bit com o uso dos *releases* para AMD64 e IntelEM64T no Linux e versões para Windows 64 bits disponíveis em um novo *release*.
- Intensa revisão da segurança do servidor, incluindo criptografia forte das senhas e dispositivo de proteção contra ataques baseados em *brute – force* embutido no próprio banco.
- Capacidade de backup incremental.
- Completa re-arquitetura de “*serverless*” do protocolo de conexão local no Windows para eliminar a instabilidade herdada do protocolo IPServer.
- Implantação completa do serviço de API² em todas as plataformas.

3.4 LINGUAGEM PSQL

A linguagem PSQL (*Procedural SQL*) é uma linguagem nativa do Firebird para *stored procedures* e *triggers*, procedimentos de grande importância em um sistema gerenciador de banco de dados.

Temos como algumas de suas características:

- Suporte das variáveis a qualquer tipo de dado suportado pelo Firebird, com exceção somente de *Array* e *Blob*.
- Os comentários devem ser feitos entre asteriscos, como por exemplo:
variável = valor; /* atribuição de uma variável */.
- Os blocos terminam com um END seguidos por um terminador, este que é definido com o comando SET TERM.
- Possibilidade de gerar exceções dinâmicas podendo criar e exibir uma mensagem no momento em que a exceção é gerada.
- Possibilidade de chamar *stored procedures* em *triggers* e SPs internamente através do comando EXECUTE PROCEDURE.

² Application Programming Interface

- *Stored procedures* selecionáveis que tem o mesmo efeito de uma *view*, sendo possível utilizá-la como se fosse uma tabela contendo informações originadas de outras tabelas do banco de dados, servindo também como fonte de dados para os SELECTs, além dos comandos:
 - IF...THEN...ELSE;
 - POST_EVENT;
 - SELECT INTO;
 - EXIT e SUSPEND;
 - WHEND...DO;
 - CASE, COALESCE e NULLIF;
 - [FOR] EXECUTE STATEMENT;
 - LEAVE/BREAK.

3.5 O FUTURO DO FIREBIRD

Alguns projetos estão sendo desenvolvidos, como o Yaffil e o Vulcan, visando desenvolver e aperfeiçoar ferramentas para o Firebird.

“O projeto Vulcan atingiu a meta de usabilidade geral, com algumas poucas pontas soltas, e desejamos juntar ambas as bases de código o mais cedo possível. Dessa junção resultará no Firebird 3.0 com total suporte a multiprocessamento simétrico (SMP), arquitetura unificada (sem mais nenhuma necessidade de compilações classic / superserver / embedded) e outros melhoramentos.” (IBPHOENIX, 2007).

O projeto Vulcan surgiu do Firebird HEAD, mais tarde denominado Firebird 1.5, para que fossem realizados testes no desenvolvimento de multi – threading (suporte a SMP) e plataformas de 64 bits, e está sendo implementado para, na junção com o Firebird 2.0, resultar na versão 3.0 do Firebird.

3.6 TIPOS DE DADOS

O *Firebird* suporta três dialetos, que determinam a disponibilidade de tipos e como serão armazenados no banco de dados.

O dialeto 1 é compatível com a versão do *Interbase* 6. O dialeto 2 é somente utilizado para testes na identificação de qualquer incompatibilidade entre os dialetos 1 e 3. Já o dialeto 3 é recomendado para o desenvolvimento, sendo o mais atual, com novas características e suporte a tipos de dados.

Na definição do dialeto de um banco de dados utiliza-se o SET SQL DIALECT x, onde x é o dialeto a ser utilizado.

Os tipos de dados suportados são:

- *Char* e *Varchar*;
- *Date/ Time/ Timestamp*;
- *Integer* , *Smallint* e *Bigint*;
- *Float*;
- *Double precision*;
- *Numeric* e *Decimal*;
- *Array*;
- *Blob*;

Char e *Varchar* são tipos utilizados para armazenar letras, números, caracteres, etc..A diferença entre um e outro é que o *Char* retorna os valores com espaço a direita, já *Varchar* retorna da mesma forma em que foram armazenados. O uso do *Char* é indicado quando o número de campos já é definido (CANTU, 2005).

O tipo *Date* armazena somente datas e o *Time* armazena somente horas, enquanto *Timestamp* armazena tanto datas como horas. Estes tipos são originados do tipo *Date* do *Interbase* pelo dialeto 3.

Integer e *Smallint* são tipos que suportam valores numéricos inteiros com sinal, onde a diferença entre ambos é que o primeiro ocupa 32 bits, enquanto o segundo 16 bits. Já o *Bigint* armazena também valores numéricos, porém guarda inteiros de até 64 bits.

O tipo *Float* guarda números de ponto flutuante, porém na maioria dos casos deve ser evitado, pois é possível que a informação armazenada não seja exatamente a mesma quando recuperada, visto que a metodologia utilizada para armazená-los, IEEE (*Institute of Electrical and Eletronics Engineers*), não garante isso.

O *Double precision*, como o *Float*, também armazena valor em ponto flutuante, porém com uma faixa maior de valores, guardando em 64 bits, contudo possui o mesmo problema do *Float*, o da precisão na recuperação dos valores (CANTU, 2005).

Os tipos *Numeric* e *Decimal* são utilizados para armazenar valores do tipo fracionário, muito importante quando é necessário precisão no retorno dos dados.

Campos do tipo *Array* não são muito utilizados no *Firebird*, pois não suportam alguns componentes de acesso a dados (CANTU, 2005). Esse tipo é equivalente a um vetor, onde é possível armazenar diversos valores do mesmo tipo.

Um campo *Blob* é utilizado para armazenar desde textos a arquivos binários. Sua capacidade de armazenamento é ilimitada, sendo limitada somente por falta de espaço em disco, por exemplo.

“Existem 6 subtipos predefinidos para campos *Blob*:

0-Dados binários

1-Informação textual

2-*BLR-Binary Language Representation*

3-Lista de controle de acesso

4-Não utilizado (reservado para uso futuro)

5-Descrição codificada para os metadados de uma tabela

6-Descrição de uma transação entre múltiplos bancos de dados que terminou de forma irregular.” (CANTU, 2005, p. 77).

Além desses existem os campos calculados, onde não é possível definir que tipo de dados irá retornar, pois são resultados de operações feitas com valores de outros campos. Quem define o tipo de dado é o servidor, baseado nos tipos dos outros campos. Além disso, os campos calculados não são armazenados nos bancos de dados.

3.7 EMPRESAS QUE CONFIAM NO FIREBIRD

A tecnologia do Firebird possui mais de vinte anos de mercado, levando em conta o período de desenvolvimento do Interbase, dando segurança e conhecimento tanto ao desenvolvedor como ao administrador do banco.

“Muitas empresas internacionais e nacionais adotaram o Firebird em seus projetos. Entre as empresas nacionais destaca-se a Embrapa (Empresa Brasileira de Pesquisa Agropecuária), a Clínica do Leite/ESALQ-USP, e centenas de empresas de desenvolvimento de software que adotaram o Firebird como banco de dados de suas aplicações.” (CANTU, 2005, p.9).

A Embrapa utiliza o software AINFO versão 5.0, que é um sistema para gerenciar bibliotecas e recuperar informações, que utiliza o Firebird como sua arquitetura cliente/servidor.

A Caixa Econômica Federal utilizava o software SEFIP, com sua base de dados no Interbase 6.0, mas devido a vários problemas relatados pelas empresas que usavam o sistema, na versão 8.3, ele foi migrado para o Firebird 1.5.

“Os aplicativos empresariais BB armazenam os dados gerados durante sua operação, para que você possa consultá-los depois, em caso de necessidade. O armazenamento é feito através do Sistema Gerenciador de Banco de Dados Firebird.” (BB, 2007).

Além dessas empresas, temos também a Faculdade de Ciências Aplicadas “Sagrado Coração”, Unilinhães, que utiliza dois servidores trabalhando em conjunto, sendo que um deles armazena o Athenas 3000, que usa o servidor Firebird.

Existem também alguns softwares que utilizam o Firebird como banco de dados. Segue abaixo alguns exemplos de softwares da ViaSoft (VIASOFT, 2007):

- Indústria Pro - indicado para empresas de pequeno e médio porte na informatização de seus serviços.
- Contábil - destinado a escritórios contábeis ou empresas que desejam fazer sua escrituração fiscal.
- Folha - voltado para o controle de remunerações devidas aos empregados da empresa.
- Estoque - possibilita o acompanhamento dos movimentos de estoque.

3.8 VERSÕES DO SERVIDOR

Ao começar a desenvolver com Firebird, surge a necessidade de escolher qual versão do servidor optar. O Firebird possui três versões distintas, *Classic*, *SuperServer* e *Embedded*. A versão *Embedded* é uma inovação e é utilizada em distribuições de programas em CD-ROM. No site Firebase foi divulgada uma pesquisa de utilização de versões, finalizada em setembro de 2005, em vários países (FIREBASE, 2007).

- *SuperServer* – 69%
- *Embedded* – 24%
- *Classic* – 5%
- Indefinido – 2%.

3.8.1 Classic

A versão *Classic* é remanescente da versão 4 do Interbase, e é baseada em processos, ou seja, para cada nova conexão cliente, é disparado um novo processo servidor para atendê-la. A principal característica é que cada processo é totalmente independente, possuindo um cache de memória dedicado. Cada processo utiliza cerca de 2MB de memória.

“[...] Utiliza processos separados do servidor para gerenciar cada conexão com o banco de dados. Se há conexões com um banco, então haverá servidores Firebird rodando, cada um gerenciando uma conexão específica, e cada um com sua própria área de memória e cache.” (CANTU, 2006, p.2).

3.8.2 SuperServer

A versão *SuperServer* traz um conceito novo, em que um único processo gerencia várias conexões clientes simultâneas. Cada conexão é denominada “thread”, ou uma pequena parcela do processo. Esse processo utiliza cerca de 2MB de memória, cada conexão adiciona cerca de 115K ao consumo de memória.

“Utiliza um único processo do servidor, que criará uma *thread* para cada nova conexão feita com um banco de dados, compartilhando o *cache* entre as conexões.” (CANTU, 2006, p.2).

3.8.3 Embedded

Uma das novidades, a partir do Firebird 1.5, foi o surgimento da versão *Embedded*, que é uma variante da versão *Super Server* para plataformas Windows. Ela é simplesmente um servidor Firebird completo em uma única DLL chamada *fbembed.dll*.

“A grande vantagem desse tipo de servidor é facilitar a vida dos desenvolvedores que desejam criar versões de demonstração dos seus softwares, catálogos em cd-rom ou distribuir seus softwares para utilização em ambiente monousuário, sem que haja necessidade do usuário final instalar um servidor Firebird em sua máquina. (CANTU, 2005, p. 241).

Essa versão é limitada ao método de acesso local e permite conexão com um único usuário. Uma aplicação “embutida”, como é conhecida a aplicação que usa a versão *Embedded*, pode ser executada na mesma máquina que estiver rodando um servidor Firebird normal, contudo os dois servidores não podem ter acesso a um banco de dados ao mesmo tempo. Todos os recursos e potencial do Firebird estão presentes na versão *Embedded*, como controle transacional, *stored procedures*, *triggers*, entre outros. É possível criar versões de demonstração de programas com alto grau de complexidade, sem praticamente qualquer mudança do código da aplicação. Esse é um recurso exclusivo do Firebird.

4 CARACTERÍSTICAS

4.1 FERRAMENTAS

Para administrar o banco de dados, o Firebird não possui ainda uma ferramenta gráfica nativa. Juntamente com o servidor Firebird, é disponibilizado um conjunto de ferramentas de linha de comando, como:

- fbmgr – interface Shell com o serviço do Super Server no ambiente Linux.
- instsvc – interface com o serviço Super Server em plataformas Windows para instalação, iniciação e desligamento do servidor.
- gbak – utilizado para gerar cópias de segurança e restaurar bancos de dados.
- gsec – ferramenta de manutenção de senhas e usuários.
- gfix – ferramenta para reconfigurar propriedades, fazer pequenos reparos e executar tarefas de limpeza no banco de dados.
- gstat – ferramenta para extração de relatórios estatísticos sobre índices e dados.
- fb_lock_print – para recuperação de estatísticas de bloqueio que o servidor Firebird mantém para controle das alterações feitas por múltiplas transações.
- isql – interactive SQL, incorpora ferramentas e técnicas para uso de SQL na manutenção de objetos dos bancos de dados, gerenciamento de transações, exibição de metadados e gerenciamento de scripts de definição de bancos de dados.

Além das ferramentas de linha de comando que acompanham o servidor, a comunidade Firebird tem a sua disposição inúmeras interfaces gráficas, muitas delas com acesso gratuito. Entre as mais conhecidas estão:

- Database Workbench (comercial)
- IBExpert (comercial), que oferece uma versão gratuita limitada (Lite)
- IBOConsole (gratuita)
- MDOConsole (gratuita), desenvolvida no Brasil, sob o comando do brasileiro Henrique Meira.

- EMS SQL Database Management Solutions (possui uma versão gratuita totalmente funcional).

Uma lista completa de ferramentas gráficas disponíveis no mercado pode ser obtida no site da IBPhoenix³. A concorrência entre os fornecedores de ferramentas tem sido outro fator interessante no trajeto de desenvolvimento do servidor Firebird.

4.2 COMPONENTES

O usuário do Firebird tem a seu dispor uma quantidade enorme de componentes de acesso ao banco de dados, tanto na tecnologia de acesso direto através da API do banco, como na de acesso através de uma camada intermediária, entre a aplicação e a API do banco, tendo compatibilidade com diversas linguagens de programação, como Java, Delphi, C, C++, Visual Basic, C#, entre outras.

“No geral, recomendo sempre o uso de componentes que forneçam acesso nativo ao banco, mas há situações nas quais isso não é viável, ou mesmo desejado...” (CANTU, 2005, p.93).

Alguns componentes de acesso ao Firebird:

- IB Objects (IBO) – Considerado o pacote mais completo. É o que oferece o maior número de componentes.
- dbExpress – É uma camada muito fina entre a aplicação e a API do banco de dados. Possui poucos componentes, em comparação com outros pacotes oferecidos.
- Interbase Express (IBX) – Devido ao fato de ser desenvolvido pela Borland, sua compatibilidade com as futuras versões do Firebird não pode ser garantida.
- JDBC – Conexão com aplicações Java utilizando o drive JDBC 4.0 disponível para download gratuito no site da Sun.

³ www.ibphoenix.com

- Zeos – É um projeto open source que existe há bastante tempo, suportando apenas o MySQL e o PostgreSQL, sendo posteriormente adicionado acesso a outros bancos, entre eles o Firebird, possuindo arquitetura baseada no JDBC do Java.
- Mercury DataBase Objects (MDO) – É um projeto open source, desenvolvido por brasileiros.
- SIBProvider (OLEDB) – É um driver OLEDB nacional (comercial) para o Firebird. Pode ser utilizado por qualquer linguagem que suporte ADO ou OLEDB, incluindo Delphi, Visual Basic, Crystal Reports, ColdFusion e ASP.

Uma pesquisa realizada pela FireBase (FIREBASE, 2007), maior comunidade brasileira sobre Firebird, aponta os componentes de acesso mais utilizados pelos desenvolvedores brasileiros:

- dbExpress – 31%
- BDE – 16%
- IBX – 15%
- IBO – 13%
- Zeos – 7%
- Java – 3%
- UIB – 2%
- Outros – 13%

4.3 PRINCIPAIS RECURSOS

4.3.1 Sistema Multi-Generacional

Um dos recursos mais vantajosos e comentados do Firebird é o uso de um sistema otimista de concorrência no acesso ao banco de dados. Na maioria dos SGBDs, se tem um bloqueio de páginas inteiras, já no Firebird, se guarda várias versões dos registros durante uma transação, mantendo dessa forma uma visão segura dos

dados, mesmo que alguma informação já tenha sido alterada após o início da transação.

Como exemplo, um sistema em que várias conexões estão ocorrendo no banco de dados ao mesmo tempo, uns estão visualizando relatórios para impressão, enquanto outros estão inserindo ou alterando os registros. Entre o *preview* de um relatório e sua impressão definitiva, um usuário altera ou insere registros que influem no relatório visualizado. Em outro SGBD, haveria diferença entre o que foi visualizado e o que realmente foi impresso. No Firebird, devido a diversos tipos de isolamento de transação, se tem uma imagem dos dados inalterada pelo tempo que quiser, sem impedir que outros usuários continuem acessando ou alterando os registros.

4.3.2 Replicação

O Firebird não possui uma ferramenta nativa para replicação⁴ do banco de dados, porém existem vários componentes de terceiros:

- IBReplicator – adquirido pela IBPhoenix.
- ReplicadorBR – desenvolvido por brasileiros, open source.
- FBReplicator – ferramenta gratuita.

4.3.3 Sombreamento

Para se precaver contra a perda de dados, ou corrupção do banco de dados em caso de falha no disco rígido, na rede ou de um usuário que pode excluir acidentalmente o banco de dados, o Firebird tem o recurso de sombreamento⁵, que pode recuperar o banco de dados nesses e em outros casos.

⁴ “Mecanismo que permite a transferência contínua de dados, provocando redundância intencional, porém controlada, com objetivo de melhorar a performance das consultas em clientes conectados a servidores que recebem as cópias.” (GULLIVER, 2007).

⁵ “Sombreamento é uma cópia física idêntica do banco de dados. Quando um sombreamento é definido, as atualizações feitas no banco de dados são escritas simultaneamente no seu sombreamento. Assim, o sombreamento sempre reflete o estado atual do banco de dados” (ARAÚJO, 2006, p.12).

Vantagens do sombreamento:

- Recuperação rápida
- Alocação de disco rígido pode ser controlada
- Não utiliza um processo separado
- Necessita de pouca ou nenhuma manutenção

4.3.4 Suporte a Diversos Protocolos de Conexão

Impossível imaginar, atualmente, empresas que não utilizam redes em seus ambientes de trabalho, pois as redes de computadores estão no centro da competição empresarial moderna. Partindo dessa ótica, um banco de dados que promete ser uma ótima opção para o desenvolvimento de softwares cliente/servidor deve dar suporte aos principais protocolos de conexão. O Firebird está neste contexto, dando suporte aos seguintes protocolos:

- Local – O protocolo Local existe hoje em duas versões, e é disponível apenas no ambiente Windows. O Firebird, a partir da versão 2.0, trouxe uma nova implementação do protocolo Local, chamada agora de XNet, onde a memória é compartilhada de forma sincronizada pelos processos. O XNet pode ser usada para conexão com um servidor Firebird *Classic*, e com serviços não interativos e sessões de terminal (*terminal services*).

Exemplo de uma *string* para conexão com o banco de dados:

C:\banco\dados.fdb

- TCP/IP (*Transmission Control Protocol / Internet Protocol*) – é o protocolo de rede mais utilizado, pois é o protocolo nativo da internet. Cada computador de uma rede possui a sua identificação (simplesmente chamada de IP), que é única. O servidor Firebird pode ser acessado pela aplicação através do IP ou nome do computador.

Exemplo de uma *string* para conexão com o banco de dados:

Windows – 192.168.1.1:c:\banco\dados.fdb ou fbserver:c:\banco\dados.fdb

Linux – 192.168.1.1:/banco/dados.fdb ou fbserver:/banco/dados.fdb

Não é necessário mapear um drive na estação para acessar o arquivo de banco de dados. O acesso ao arquivo físico é feito pelo processo do servidor Firebird, que está rodando na máquina servidora. O usuário associado ao processo do servidor Firebird deve ter direito de acesso aos arquivos do banco de dados. A porta padrão de conexão do Firebird é a 3050.

- NetBEUI (*Net Bios Extended User Interface*) – é hoje um protocolo em decadência, visto que não pode ser usado em redes com tamanho razoável, pois não se comunica com outras redes, por vários fatores, entre eles o de não ser roteável, causando ainda muito tráfego na rede. Nos sistemas operacionais Windows, está presente até o Windows 2000, o XP não vem com a instalação padrão.

Exemplo de uma *string* para conexão com o banco de dados:

\\fbserver\c:\banco\dados.fdb

“[...] para conexão local, é mais interessante usar o protocolo XNet, e para as outras, o recomendado é o TCP/IP” (CANTU, 2006, p.67).

4.3.5 UDF

O Firebird possui suporte a UDF's⁶, que podem ser criadas em qualquer linguagem de programação que gere bibliotecas compartilhadas, conhecidas como dll⁷ no Windows e so⁸ no Linux. Podem ser utilizadas em comandos SQL (com select), ou em Stored Procedures e Triggers, aumentando o poder do banco de dados, pois uma única biblioteca pode conter inúmeras funções (UDF's).

⁶ “User Defined Functions, ou UDFs, é a forma que o Firebird usa para que rotinas e comandos SQL executados no servidor chamem rotinas externas, podendo passar e recuperar parâmetros.” (CANTU, 2006, p.219).

⁷ *Dinamic Link Library*

⁸ *Shared Objects*

Segundo CANTU (2005), a criação de uma biblioteca de função é geralmente uma tarefa simples, e devem ser seguidos os seguintes passos:

- Codificar a biblioteca;
- Compilar o código para gerar a .dll ou .so;
- Transportar a biblioteca para o diretório padrão do Firebird, na pasta UDF e
- Declarar as funções em seu respectivo banco de dados.

```
// Função para remover caracteres que não sejam letras e números
function Limpa (sz, szRes: PChar): PChar;
begin
    result:= szRes;
    while (sz^ <> # 0) do
    begin
        if sz^ in ['A'..'Z', 'a'..'z', '0'..'9'] then
        begin
            szRes^:= sz^;
            inc (szRes);
        end;
        inc (sz);
    end;
    szRes^:= #0;
end;
```

Figura 4.1 – Exemplo de declaração de uma UDF no banco de dados
Fonte: CANTU (2005, p.255)

4.3.6 Instalação Reduzida

Outro recurso muito atraente no Firebird, principalmente do ponto de vista do desenvolvedor de software, é o tamanho do instalador (menos de 5mb na versão 2.0.1.12855-1-win32), que torna o seu envio em conjunto com a instalação do sistema extremamente simples, mesmo que o sistema tenha que ser baixado de algum servidor na internet. A instalação do *client* necessita de apenas 3 arquivos e fica com menos de 1mb, já a instalação completa do *server* fica em 13mb. Para se ter uma idéia do tamanho reduzido do Firebird, o SQL Server, SGBD comercial da Microsoft, tem um instalador de 118mb e sua instalação ocupa 199mb de espaço em disco.

“[..], o Firebird é um dos menores (em termos de Kb) servidores de banco de dados que existem, ocupando pouquíssimo espaço em disco. Somando isso ao fato de ser um banco de dados open source e portanto poder ser distribuído gratuitamente, muitos desenvolvedores de software pretendem enviar o Firebird junto com seus produtos e gostariam de ter uma instalação única e personalizada de todo o pacote.” (CANTU, 2005, p.271).

4.3.7 Tamanho do Banco de Dados

O tamanho máximo de um arquivo de banco de dados Firebird, teoricamente é de 2GB no Windows 95/98 e de 4GB no Windows NT/2000. O número máximo de arquivos por banco de dados Firebird é 65.536, sendo este o limite do Firebird, podendo ainda ser limitado pelo sistema operacional (ARAÚJO, 2006).

O Firebird é limitado a capacidade de processamento do sistema operacional, podendo atingir a ordem de Terabytes e manusear essa quantidade de informação com tranquilidade.

Aconteceu recentemente em Moscou (Rússia), a *Softool'06*, entre os dias 26 e 29 de Setembro de 2006, onde o público presente pode acompanhar a desmistificação de que o Firebird é um SGBD para pequenos bancos de dados e poucas conexões.

“Um bom exemplo foi mostrado na *Softool'06*, onde o *Avarda* (ERP russo) estava rodando com um servidor Firebird 2.0 *Classic* e um número médio de 100 conexões simultâneas, acessando um banco de dados de 120GB com 700 milhões de registros” (BUBLITZ, 2007, p. 50).

Os arquivos de banco de dados crescem dinamicamente, conforme a necessidade, não sendo necessário um planejamento minucioso do tamanho que o banco de dados pode alcançar para pré-alocar esse espaço. Havendo espaço no HD, o Firebird o utilizará automaticamente sempre que se fizer necessário. Índices, procedures, triggers, enfim, todos os objetos do banco de dados ficam armazenados em um único arquivo, com a extensão .fdb. Podemos ter, em teoria, bases com alguns Terabytes de informação.

4.3.8 Backup On-line

Para realizar backup no Firebird, não há necessidade de se parar o servidor, pois um processo captura a imagem da base de dados no momento em que ele inicia. Desta forma os usuários podem continuar realizando conexões normalmente com o servidor, num processo ininterrupto, enquanto o backup é executado.

“O Firebird realiza backup sem a necessidade de acesso exclusivo ao banco. O backup contém uma imagem consistente do banco de dados no momento em que o processo de backup foi iniciado, ou seja, não importa que o backup demore uma hora, ele vai conter a imagem exata do banco de dados no momento em que ele foi iniciado.” (CANTU, 2005, p.5).

Para gerar e/ou restaurar um backup no Firebird, existe o utilitário de linha de comando chamado gbak (gbak.exe), que se encontra na pasta de instalação do Firebird, dentro da pasta bin.

```
gbak -user SYSDBA -pas masterkey 192.168.1.10:c:\dados\banco.fdb
c:\backups\backup.gbk
```

Figura 4.2 - Exemplo de geração de backup
Fonte: www.firebase.com.br

Na figura 4.2 será criado um arquivo de backup chamado backup.gbk, na pasta backups no drive c: do computador onde o gbak é executado. O banco de dados está em outro computador, determinado pelo IP 192.168.1.10 e pelo path c:\dados\banco.fdb. O comando deve ser digitado em uma linha única, não podendo haver quebra de linha, como acima.

```
gbak -user SYSDBA -pas masterkey -r -p 4096 c:\backups\backup.gbk
192.168.1.10:c:\dados\banco.fdb
```

Figura 4.3 - Exemplo de restauração de backup
Fonte: www.firebase.com.br

Na figura 4.3 será restaurado o banco c:\dados\banco.fdb no servidor indicado pelo IP 192.168.1.10. O novo banco de dados terá as páginas com o tamanho de 4k (4096 bytes) e caso ele já exista, será sobrescrito (-r).

4.4 VANTAGENS

4.4.1 Suporte

Um diferencial no uso do Firebird é a comunidade crescente que participa do desenvolvimento do projeto, seja trabalhando no código fonte, ou sugerindo novas implementações e relatando falhas e bugs. O Brasil possui uma das maiores comunidades de Firebird no mundo, e tem vários especialistas no país.

“É fácil encontrar comunidades on-line na internet, dedicadas ao suporte dos bancos de dados open source. Geralmente as dúvidas são respondidas nas listas ou fóruns de discussão em questão de horas ou mesmo minutos. A adoção cada vez maior dos BDs open source está provocando o aparecimento de empresas especializadas em consultoria para esses bancos, uma opção para aqueles que precisam da segurança de um suporte especializado ou contratado.” (CANTU, 2006. p.21).

4.4.2 Portabilidade

Ao escolher um banco de dados, talvez o sistema operacional em que está baseado o servidor, já esteja definido. Por outro lado, o banco de dados deve propiciar ao administrador uma transição para o maior número possível de sistemas operacionais.

“[...] é conveniente que o banco acompanhe as mudanças ou atualizações dos recursos de TI permitindo transições suaves.” (SANTOS, 2007, p.56). Nesse contexto o Firebird dá garantias necessárias ao administrador, estando disponível para os seguintes sistemas operacionais:

- Windows;
- Linux;
- HPUNIX;
- MacOS;
- Solaris;
- SinixZ;

- FreeBSD;
- AIX;
- Darwin e
- Windows CE.

“A portabilidade do banco de dados entre múltiplas plataformas o torna também indicado para aplicações que operam com atualizações de bases off line e que tenham necessidade de integração centralizada.” (SILVA, 2006, p. 53).

A cada ano, o site LinuxQuestions.org, realiza uma pesquisa com usuários de Linux, para saber qual banco de dados eles utilizam. Em 2006 (LINUXQUESTIONS, 2006), o resultado foi:

- MySQL – 61,68%
- PostgreSQL – 15,89%
- Firebird – 11,84%
- SQLite – 4,88%
- Oracle – 3,32%
- Outros – 2,39%

4.4.3 Open Source

O Firebird é um produto open source⁹, visto que os sistemas que são desenvolvidos com ele não precisam pagar qualquer quantia a Fundação que o desenvolve, nem mesmo permitir que os usuários que adquirem a licença do sistema possam vendê-lo, sem o seu consentimento.

“Há diversas soluções de armazenamento de dados disponíveis no mercado, e as de código livre vêm ganhando destaque devido ao número cada vez maior de recursos oferecidos.” (SANTOS, 2007, p. 52).

⁹ “Segundo a Free Software Foundation, [...] Open Source é aquele cujo código fonte é de domínio público, podendo assim qualquer pessoa usá-lo, melhorá-lo, adaptá-lo ou corrigir qualquer erro.” (WEBTUGA.COM, 2007).

4.5 LIMITAÇÕES

4.5.1 Segurança

A maior preocupação dos administradores de banco de dados, desde o surgimento do ambiente de rede corporativa, é a segurança dos dados.

“Um dos aspectos mais importantes em gerenciar um banco de dados é garantir a segurança dos dados. Você deve assegurar-se de que todos que precisam realmente acessar os dados possam fazê-lo, mas que o acesso seja adequado às reais necessidades e que ninguém tenha acesso não apropriado.” (REBECCA, 2002, p.39).

A questão da segurança em nível de banco de dados no Firebird, apesar das mudanças realizadas na versão 2.0, ainda merece atenção redobrada por parte dos desenvolvedores, para contornar certas limitações, visto que na época do surgimento do Interbase muito pouco se falava em questões de segurança da informação.

O Firebird arquiva tudo que está relacionado à segurança (usuários e senhas), em um banco de dados central, independente de quantos outros bancos de dados existam no computador. Na versão 1.0 esse banco é denominado isc4.gdb, fortemente identificado ainda ao Interbase 6, já na versão 1.5 foi denominado Security.fdb, quebrando essa relação, e na versão 2.0 se chama Security2.fdb.

“O Firebird 2.0 traz diversas melhorias na questão segurança, minimizando a possibilidade de ataques ao servidor. A preparação do terreno está feita, e podemos esperar grandes novidades nessa área para o Firebird 3.0[..].” (CANTU, 2006, p.50).

4.5.1.1 Usuário SYSDBA

Todo servidor Firebird possui um administrador de banco de dados (DBA) denominado SYSDBA, que é um usuário que está imune às regras de segurança dos bancos de dados. Sua senha, por default, é masterkey, e é o único usuário, que

pode alterar o banco de dados de segurança. Uma medida de segurança indicada é mudar a senha padrão, por ser de conhecimento público. Apesar de vários avanços no Firebird 2.0, principalmente na área de segurança, o projeto manteve o usuário SYSDBA com todos os seus privilégios.

“Antes que o leitor pergunte, não... infelizmente o SYSDBA continua podendo abrir qualquer banco de dados – melhorias nesse sentido estarão presentes no Firebird 2.5 ou 3.0.” (CANTU, 2006, p.33).

Até a versão 1.5, qualquer usuário com privilégio suficiente podia se conectar ao banco de dados de segurança remotamente e manipular as informações contidas nas tabelas. Por exemplo, um computador da rede, que tivesse privilégios de administrador, e o servidor Firebird instalado, podia acessar facilmente o computador servidor de banco de dados, e manipular todo o banco tranquilamente, como se fosse seu administrador. Na versão 2.0, o Firebird não permite essas conexões remotas, sendo feita exclusivamente através do servidor, via API de serviços, fortalecendo as barreiras contra intrusos que conhecem essa brecha na segurança do banco.

4.5.1.2 Criptografia

A criptografia¹⁰ tem a finalidade de proteger a “mensagem” de pessoas não autorizadas, pois mesmo que elas obtenham meios de acessá-la, não terão como decodificá-la, pois não possuem a chave que a “abre”.

Em um banco de dados, é de suma importância que somente pessoas autorizadas possam ter acesso ao arquivo de banco de dados físico, e nesse ponto o Firebird, até a versão 2.0, disponibiliza essa segurança apenas nas senhas de usuários.

¹⁰ “A criptografia é a codificação e o embaralhamento de mensagens para impedir o acesso não autorizado ou o entendimento dos dados transmitidos.” (LAUDON, 2001, p.349).

“Um dos pontos fracos do Firebird é não possuir dispositivo para criptografar dados. Os únicos dados onde se aplica criptografia são as senhas de usuários.” (MECENAS, 2006, p.19).

4.5.1.3 Ataques de Força Bruta

Quando surgiu o Interbase, ataques de força bruta¹¹ não eram considerados um motivo de preocupação, visto que o poder de processamento das máquinas era muito baixo, mas atualmente com desktops de alguns *Gigahertz* a preço acessível, o risco é real.

O Firebird 2.0 tenta se proteger monitorando o número de tentativas de conexão que um usuário ou IP requisitam num determinado espaço de tempo. Se detectar que está em andamento um ataque de força bruta, o Firebird passa a recusar os pedidos de conexão desse usuário ou IP por alguns segundos. Esse mecanismo de defesa não existe nas versões anteriores.

4.5.2 Número Máximo de Clientes ou Conexões Simultâneas

Embora essa não seja uma limitação do Firebird, esse tópico foi incluído, pois muito se fala nesse ponto como uma limitação do Firebird, sendo uma inverdade. O Firebird, como outros SGBDs, depende em muito do sistema operacional escolhido, e a limitação do número de conexões simultâneas está totalmente ligada a ele.

”O número máximo de clientes conectados é de 1024 (protocolo TCP/IP), embora se deva trabalhar com um número máximo de 150 clientes concorrentes, no modelo *SuperServer*, em aplicações interativas consideradas normais. [...] existem aplicações que suportam entre 300 e 350 usuários concorrentes, como as existentes na livraria municipal de Praga. Teoricamente esse limite poderia chegar a 400 usuários [...]” (MECENAS, 2006, p.15).

¹¹ “Ataques de força bruta são aqueles onde um código malicioso tenta conectar um servidor usando como senha uma lista de palavras previamente definidas, ou mesmo formando essas palavras através de algoritmos específicos, a fim de adivinhar as senhas utilizadas pelos usuários.” (CANTU, 2006, p.39).

Na versão *Classic* é necessário lembrar que cada conexão inicia um novo processo que consome em torno de 2MB de memória, e se tivermos 100 conexões simultâneas, teremos um consumo de 200MB de memória, estando claramente limitado o número de conexões a disponibilidade do *hardware*.

4.5.3 Interface para Administração do Banco

Ainda não possui ferramentas nativas de interface gráfica, tendo apenas componentes de linha de comando, mas existe um projeto de interface gráfica, de código aberto, em andamento, denominado FlameRobin¹², com versões para Windows, Mac OS, FreeBSD, Solaris e Linux (Debian, Ubuntu, Mandriva) e que no futuro deve fazer parte do pacote de instalação do Firebird, suprimindo essa carência.

Essa é uma limitação que é contornada naturalmente pelo desenvolvedor, pois inúmeras ferramentas de terceiros, gratuitas ou não, específicas para o Firebird, podem ser encontradas com facilidade na internet.

“Uma das ferramentas mais completas é o IBExpert[..], que possui uma versão Free (com algumas limitações) e uma versão shareware completa[..].” (CANTU, 2005, p.8).

¹² Maiores informações no site www.flamerobin.org (em inglês).

5. ESTUDO DE CASO

Ao realizarmos a pesquisa para o trabalho, monitoramos um banco de dados montado com o Firebird versão *SuperServer* 2.01, com mais de 170.000 registros, uma média de 350 BAU's¹³ por dia e até 15 conexões concorrentes. Esse banco de dados encontra-se em funcionamento no Hospital Geral de Linhares (HGL), originalmente foi criado no ano 2000, com o interbase 6.0 (open source), sendo um arquivo .gdb.

Em Janeiro do ano corrente passei¹⁴ a administrar essa base de dados, que constantemente nos causava transtornos, devido à demora de conexão das estações clientes com o servidor. Esse *delay* era em torno de 20 a 25 segundos, o que gerava descontentamento nos colaboradores, que tentavam abrir várias vezes o sistema, ocasionando travamento na máquina cliente. Em Fevereiro desse ano, depois de consultas na internet com profissionais que já utilizavam o Firebird resolvemos substituir o Interbase 6.0 pelo Firebird *SuperServer* 1.5.3. O arquivo de dados continuou sendo .gdb, mesmo assim o desempenho foi surpreendente, com as aplicações clientes acessando o banco instantaneamente, solucionando o problema de *delay* entre a solicitação de abertura do sistema e o retorno ao colaborador.

Em maio, incentivado pelos estudos feitos para o trabalho, resolvemos trocar a versão do Firebird. Atualmente estamos utilizando a versão *SuperServer* 2.01, e o arquivo de dados foi mudado para .fdb, num processo simples, embora feito em linha de comando. Hoje nosso arquivo de dados é totalmente Firebird, embora tenha sido criado no Interbase.

¹³ Boletim de Atendimento de Urgência.

¹⁴ Gustavo André de Freitas é funcionário público, lotado no Hospital Geral de Linhares, no setor de TI.

6. CONCLUSÃO

A escolha de um SGBD leva o profissional de tecnologia da informação a analisar diversos fatores como a empresa na qual será utilizado, a necessidade de processamento, o hardware utilizado, o sistema operacional já instalado, a familiaridade do desenvolvedor do sistema, o conhecimento dos profissionais que estarão envolvidos com a manutenção do sistema, entre vários outros fatores.

“Escolher que banco utilizar como qualquer outro produto de informática é uma questão que requer alguma avaliação.” (MECENAS, 2006, p.20).

E surge também a questão que levanta suspeita em algumas empresas, o software livre, open source, gerando desconfiança e medo de não se ter por trás deste uma equipe especializada.

“Talvez o maior fator limitante na adoção do Firebird [...] frente aos bancos comerciais não seja a falta de um ou outro recurso, mas sim o medo que algumas empresas ainda tem na adoção de um software livre. Felizmente, com a popularização do Linux e de outros softwares abertos em grandes empresas e corporações, assim como o investimento de gigantes como a IBM em alguns desses softwares, esse medo tende a deixar de existir em um futuro próximo.” (CANTU, 2004, p.22).

6.1 VALE A PENA UTILIZAR?

“O Firebird hoje é uma alternativa forte, gratuita e com código aberto, testada, em pleno desenvolvimento e que conta com uma imensa base de usuários espalhados por todo o mundo. Obviamente ainda há projetos que necessitam do poder de um Oracle, mas o Firebird tem seu espaço garantido em projetos de pequeno e médio porte.” (CANTU, 2004, p.23).

Essa opinião do Cantu foi no ano de 2004 e na época a versão do Firebird era a 1.5. Com o lançamento do Firebird 2.0 em novembro de 2006, o cenário mudou, pois o Firebird evoluiu, tendo sua tecnologia amplamente testada pela grande comunidade mundial que auxilia no desenvolvimento através de sugestões e solicitações de melhorias e reportando *bugs*, encontrando seu espaço entre os desenvolvedores que procuram um SGBD seguro, robusto, e que possa ser distribuído gratuitamente para seus clientes, não onerando o custo de seus sistemas.

“O Firebird possui várias funcionalidades interessantes, já que é um banco com uma longa história, e também se destaca pela facilidade de configuração e manutenção, sendo um banco com grande autonomia.” (SANTOS, 2007, p.57).

Para pequenos e médios projetos, o Firebird surge como uma alternativa altamente viável, e é certo que ele continuará a se desenvolver, evoluindo a cada dia, pois já se fala numa versão 3.0, que seria a união da versão 2.0 com o tão comentado projeto Vulcan podendo assim num futuro bem próximo, ser uma alternativa viável também para grandes projetos.

7 REFERÊNCIAS

- 1 ARAÚJO, Carlos Alberto Pedroso. **Programação Cliente/Servidor com Firebird**. Florianópolis: Visual Books, 2006.
- 2 CANTU, Carlos Henrique. **Firebird Essencial**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2005.
- 3 CANTU, Carlos Henrique. **Firebird 2.0 – o banco de dados do novo milênio**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2006.
- 4 DATE, C. J.. **Introdução a Sistemas de Banco de Dados**. 8.ed. Rio de Janeiro: Elsevier, 2005.
- 5 ELSMARI, Ramez; NAVATHE, Shamrant B.. **Sistemas de Bancos de Dados**. São Paulo: Addison Wesley, 2005.
- 6 LAUDON, Kenneth C.; LAUDON, Jane Price. **Gerenciamento de Sistemas de Informação**. 3 ed. Rio de Janeiro, 2001.
- 7 MEDEIROS, Marcelo. **Banco de Dados para Sistemas de Informação**. Florianópolis: Visual Books, 2006.
- 8 REBECCA, M. Riordan. **Microsoft SQL Server 2000 – Programando Passo a Passo**. São Paulo: Pearson Education, 2002.
- 9 Revistas:
 - BUBLITZ, Jorge Luis. Novidades do Firebird 2.0. **ClubeDelphi**, Rio de Janeiro, ano 6, n. 81, 2006.
 - CANTU, Carlos Henrique. A escolha do banco de dados. **DB FreeMagazine**, Rio de Janeiro, ano 2, n. 8, 2006.
 - CANTU, Carlos Henrique. Firebird, essa é a minha escolha, e a sua?. **SQL magazine**, Rio de Janeiro, ano 2, n. 15, 2004.
 - MECENAS, Ivan. Firebird. **SQL magazine**, Rio de Janeiro, ano 3, n. 37, 2006.
 - SANTOS, Paulo Sérgio M. dos. Comparando Firebird, MySQL, PostgreSQL e SQLite. **SQL magazine**, Rio de Janeiro, ano 4, n. 41, 2007.
 - SILVA, Hélio. Firebird poupa a máquina. **Info EXAME Coleção 2006 – banco de dados**, São Paulo, edição 27, 2006.

SILVA, Willian. IBM DB2 UDB Express C-Edition. **SQL magazine**, Rio de Janeiro, ano 3, n. 37, 2006.

10 Sites:

BB. Servidor de Banco de Dados Firebird para Windows. Disponível em:
<<http://office.bb.com.br/office/plugins/frame6.html>>.
Acesso em: 08 Mai. 2007.

GULLIVER, Alvaro. **Fundamentos de dados distribuídos.** Disponível em:
<<http://www.devmedia.com.br/articles/viewcomp.asp?comp=5017&hl=replicação>>.
Acesso em: 20 Mai. 2007.

FIREBASE. Qual a versão mais utilizada do Firebird 1.5? Disponível em:
<http://www.firebase.com.br/fb/imgdocs/FB_Survey_2005.pdf>.
Acesso em: 08 Mai 2007.

FIREBASE. Qual é seu principal método de acesso ao FB.
Disponível em: <<http://www.firebase.com.br/fb/polls.php?resultado=14>>.
Acesso em: 20 Mai. 2007.

FREITAS, Alfredo Américo de. Padrão ANSI SQL-92.
Disponível em: <<http://www.firebird.com.br>>.
Acesso em: 08 Mai. 2007.

IBPHOENIX. Projeto Vulcan. Disponível em:
<<http://www.ibphoenix.com/downloads/VulcanOverview.pdf>>.
Acesso em: 08 Mai. 2007.

LINUXQUESTIONS. Qual o seu banco de dados preferido?
Disponível em:
<<http://www.linuxquestions.org/questions/showthread.php?t=514950>>.
Acesso em: 05 Mai. 2007.

UNDER-LINUX.ORG. O que é SorceForge.net. Disponível em:
< <http://under-linux.org/6322-o-que-e-sourceforgenet.html>>.
Acesso em: 20 Mai. 2007.

VIASOFT. Empresa que trabalha com Firebird.
Disponível em: <<http://www.viasoft.com.br>>.
Acesso em: 09 Mai. 2007.

WEBTUGA. Software Livre e Open Source. Disponível em:
<<http://www.webtuga.com/2007/03/21/software-livre-e-open-source.php>>.
Acesso em: 20 Mai. 2007.