



Fundação Educacional do Município de Assis
IMESA - Instituto Municipal de Ensino Superior de Assis

ANDERSON WILLIAM SILVA ELIZIÁRIO

FRAMEWORK GERADOR DE SQL PARA REPLICAÇÃO DE DADOS

**Assis
2014**



Fundação Educacional do Município de Assis
IMESA - Instituto Municipal de Ensino Superior de Assis

ANDERSON WILLIAM SILVA ELIZIÁRIO

FRAMEWORK GERADOR DE SQL PARA REPLICAÇÃO DE DADOS

Trabalho de conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do curso de Análise e Desenvolvimento de Sistemas.

Orientado: Anderson William Silva Eliziário

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto.

**Assis
2014**

FICHA CATALOGRÁFICA

ELIZIÁRIO, Anderson W. S.

Framework Gerador de SQL para Replicação de Dados / Anderson William Silva Eliziário. Fundação Educacional do Município de Assis – FEMA – Assis,2014.

43p.

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Framework. 2. Gerador 3. Replicação.

CDD: 001.61
Biblioteca da FEMA

FRAMEWORK GERADOR DE SQL PARA REPLICAÇÃO DE DADOS

ANDERSON WILLIAM SILVA ELIZIÁRIO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Analisador (1): Prof. Esp. Guilherme de Cleva Farto

**Assis
2014**

DEDICATÓRIA

Dedico este trabalho à minha esposa, minha filha, ao meu orientador e a todos que me ajudaram direta e indiretamente nestes três anos.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me sustentado e me ajudado durante esta fase da minha vida, pela forma, coragem para conseguir enfrentar todas as dificuldades ao longo destes três anos muito importantes.

Ao prof. Dr. Alex Sandro Romeo de Souza Poletto, pela compreensão e auxílio na orientação, que foi indispensável para a conclusão deste trabalho.

A minha esposa Nayara pela compreensão no decorrer destes anos de curso e por sempre estar ao meu lado me apoiando.

Aos meus pais Nivaldo e Claudia pelo incentivo para não desistir dos meus sonhos.

Aos meus amigos, em especial ao Eduardo Sodré pelo apoio para a realização do curso e pela ajuda durante a o mesmo, e aos outros que me ajudaram de alguma forma compartilhando seus conhecimentos e ajuda.

Agradeço a todos que direta ou indiretamente me ajudaram para que este trabalho se concretizasse.

" Sem a curiosidade que me move, que me inquieta, que me insere na busca, não aprendo nem ensino".

(Paulo Reglus Neves Freire)

RESUMO

Este trabalho tem por objetivo o desenvolvimento de um framework gerador de SQL para replicação de dados. O mesmo precisará de um sistema onde o framework irá se integrar de forma que fará funcionalidades diferentes, dando qualidade ao software que estiver integrado e dando agilidade no desenvolvimento do mesmo. O sistema irá gerar logs, onde o framework gerador de SQL para replicação de dados tem como objetivo através dos logs, gerar SQLs e assim após um processo, fazendo as replicações para os demais servidores construindo comunicações entre eles e atualizando as informações do banco de dados.

Palavras-chaves: Framework; Gerador; Integrado.

ABSTRACT

The aim of the present work is the development of a SQL framework generator for data replication. It will need a system where this framework will integrate itself, achieving different functions, giving a better quality and agility to the integrated software and its development process. This system will create logs where the SQL framework generator for data replication has for objective, using the logs, to create SQLs and then, after a process, replicating to other servers, building the communication between them, updating the data stored in the database.

Keywords: Framework, Generator, Integrated.

LISTA DE ILUSTRAÇÕES

Figura 1 – Estrutura da Replicação	27
Figura 2 – Tabela SqlAtualizações	28
Figura 3 – Tabela SqlServidor	29
Figura 4 – Tabela SqlExecuta	30
Figura 5 – Método DAO	31
Figura 6 – Código de Gerar SQL INSERT	33
Figura 7 – Método de Grava SQL	34
Figura 8 – Estrutura Analítica do Projeto (WBS)	35
Figura 9 – Sequenciamento das Atividades Definidas	36
Figura 10 – Caso de Uso	38
Figura 11 – Diagrama de Classe	39

LISTA DE TABELAS

Tabela 1 – Cronograma de Atividades	36
Tabela 2 – Estimativas de Custos	37
Tabela 3 – Lista de Eventos	37

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVOS.....	14
1.2 JUSTIFICATIVAS	15
1.3 MOTIVAÇÃO	15
2 REVISÃO DA LEITURA	16
2.1 LINGUAGEM DE PROGRAMAÇÃO	16
2.2 ORIENTAÇÃO A OBJETIVOS	16
2.3 JAVA.....	17
2.4 BANCO DE DADOS	18
2.5 MYSQL	18
3 FERRAMENTAS E TECNOLOGIAS	20
3.1 ECLIPSE.....	20
3.2 ASTAH COMMUNITY.....	20
3.3 NAVICAT	21
3.4 FRAMEWORK.....	22
3.4.1 Tipos de Framework	22
3.4.1.1 Caixa-Branca.....	23
3.4.1.2 Caixa-Preta.....	23
3.4.1.3 Caixa-Cinza	23
3.5 FRAMEWORK UTILIZADOS.....	23
3.5.1 Hibernate	23
3.5.2 Vantagens do Hibernate	24
3.5.3 API Reflection	24
4 PROPOSTA DE TRABALHO	25
4.1 INTRODUÇÃO.....	25
4.2 CATEGORIZAÇÃO DO FRAMEWORK RELICADOR	25

4.3 FUNCIONAMENTO GERAL DO FRAMEWORK REPLICADOR	26
5 ESTUDO DE CASO	28
5.1 TABELA SQLATUALIZAÇÕES	28
5.2 TABELA SQLSERVIDOR	29
5.3 TABELA SQLEXECUTA.....	30
6 PLANEJAMENTO DO PROJETO.....	35
6.1 ESTRUTURAS ANALÍTICAS DO PROJETO (WBS)	35
6.2 SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS	36
6.3 CRONOGRAMA DE ATIVIDADES.....	36
6.4 ESTIMATIVA DE CUSTOS	37
6.5 LISTA DE EVENTOS.....	37
6.6 CASO DE USO.....	37
6.7 DIAGRAMA DE CLASSE	38
7 CONCLUSÕES	40
REFERÊNCIAS	41

1 INTRODUÇÃO

Pensando em empresas que possuem filiais, e que precisam ter um sistema que faça a integração dos dados, chegou-se à conclusão de que será necessário o desenvolvimento de um *framework*. Este *framework* terá como função, replicar dados entre "n" Bancos de Dados distribuídos em diferentes locais, proporcionando ao programador redução de tempo, uma vez que utilizado este *framework* pretende-se desenvolver o sistema em pouco tempo.

O *framework* será utilizado nos sistemas exigindo a criação de três tabelas. As *tabelas*, serão necessárias para armazenar as transações do sistema que está usando o *framework*, sendo elas, gravar, inserir, editar; junto a mesma, será disponibilizada uma classe utilitária para os programadores usarem, de tal forma que o sistema já estará gerando os *logs*.

O segundo desafio, será o desenvolvimento de um aplicativo "*replicador*"(motor), cuja função em conjunto com o *framework*, realizara as "*replicações*"(atualizações), de todas as instruções SQL que estiverem armazenadas nas tabelas que foram criadas. As "*replicações*" dos dados serão realizadas de tempo a tempo, sendo as mesmas, configuradas pelo programador. Para o desenvolvimento desse *framework*, bem como para o "*Replicador de Dados*" será utilizada a linguagem Java.

Java é uma linguagem de programação orientada a objetos, multiplataforma, e roda em qualquer sistema operacional aonde se tenha um interpretador instalado. Esse Interpretador denominado máquina virtual JAVA (JVM), é um programa que converte o código Java em comandos que o sistema operacional possa executar.

Hoje a tecnologia Java proporciona inúmeras funcionalidades que fazem frente a qualquer outra tecnologia, e uma das coisas mais fascinantes e produtivas do Java é a *Reflection*, uma característica intrínseca desta tecnologia e que existe em poucas outras como existe nessa. Ela está presente no Java desde a versão 1.1. A *Reflection* permite um programa Java examinar ou fazer a introspecção nele mesmo,

ou seja, olhar e examinar suas propriedades e estrutura. Com isso, você pode, por exemplo, obter o nome de todos os membros de uma classe, como atributos e métodos, bem como executar um método usando a *Introspection*. É deste modo que os aplicativos de ambiente de desenvolvimento (as IDEs) conseguem examinar e exibir a estrutura e conteúdo das classes e beans, conforme Destro (2002).

Será utilizado no desenvolvimento do "*framework*" o Sistema Gerenciador de Bancos *MySQL*. Segundo Pisa (2012), o **MySQL** é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

Além do MySQL, será utilizado o Hibernate. Segundo Cruz, o Hibernate é um framework para o mapeamento *objeto_relacional* escrito na linguagem Java, mas também disponível em .Net com o nome NHibernate. Este programa facilita o mapeamento dos atributos entre uma base de dados relacional tradicional, com um modelo objeto de uma aplicação, mediante o uso de arquivos (XML), para estabelecer esta relação. Hibernate é um software livre, de código aberto, distribuído com a licença LGPL (Lesser General Public License).

Um *framework*, segundo França (2000), é uma estrutura de classes inter-relacionadas, que constitui uma implementação inacabada, para um conjunto de aplicações de um domínio. Além de permitir a reutilização de um conjunto de classes, um *framework* também minimiza o esforço de desenvolvimento de novas aplicações, pois já contém a definição de arquitetura gerada a partir dele, bem como tem predefinido o fluxo de controle da aplicação.

1.1 OBJETIVO

O desenvolvimento do "*framework gerador de SQL para replicação de dados*" auxiliará nos sistemas de empresas, principalmente para aquelas que possuem filiais e precisa de um sistema que faça as atualizações entre as mesmas, sendo elas, o controle de vendas, estoque, cadastro de clientes login de funcionários entre outras realizações relacionadas ao sistema. Além das informações já citadas, outra função, é agilizar o processo de desenvolvimento de software.

1.2 JUSTIFICATIVA

A necessidade do desenvolvimento deste "*framework*" se dá por atender as empresas que necessitam de sistemas que faça as atualizações de suas transações necessárias, como cadastro de cliente, baixa de produtos em estoque, etc., com uma maior eficiência. Para os desenvolvedores de software, ira poupar mais trabalho na hora de acrescentar novos atributos, já que não será necessário saber a qual classe acrescentar, e nem mesmo criar uma nova classe, sendo que o próprio "*framework*", se encarregara disso.

1.3 MOTIVAÇÃO

Pensando na redução de tempo no desenvolvimento de sistemas complexos, cujas funções serão controlar dados das empresas, sendo eles, vendas, cadastros de cliente, fornecedores, produtos, etc. A principal motivação para o desenvolvimento desse *framework*, deu-se pelo fato de ser uma tecnologia que auxiliará nas tarefas já mencionadas, fazendo as transações de dados entre as empresas e suas filiais.

O objetivo do capítulo 1 é mostrar com relação ao trabalho proposto, um pouco de como será realizado o framework gerador de SQL para replicação de dados, falando também da motivação para criação do mesmo e qual o objetivo deste trabalho ou seja o que levou para ser criado um framework com essas características.

2 REVISÃO DA LITERATURA

Este capítulo estará especificando algumas definições sobre questões relacionadas com o desenvolvimento do framework gerador de SQL para replicação de dados:

2.1 LINGUAGEM DE PROGRAMAÇÃO

Uma linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias. Uma das principais metas das linguagens de programação é permitir que programadores tenham uma maior produtividade, permitindo expressar suas intenções mais facilmente do que quando comparado com a linguagem que um computador entende nativamente (código de máquina). Assim, linguagens de programação são projetadas para adotar uma sintaxe de nível mais alto, que pode ser mais facilmente entendida por programadores e engenheiro de software possam escrever programas mais organizados e com maior rapidez, segundo Vasch (2011).

2.2 ORIENTAÇÃO A OBJETOS

A proposta da orientação a Objetos é representar o mais possível as situações do mundo real nos sistemas computacionais. Nós entendemos o mundo como um todo composto por vários que interagem uns com os outros. Da mesma maneira, a Orientação a Objeto consiste em considerar os sistemas computacionais não como uma coleção estruturada de processos, mas sim como uma coleção de objetos que interagem entre si.

Os programas Orientados a objetos são programas estruturados em módulos que agrupam um estado e operações sobre este estado. Apresentam ênfase em reutilização de código. Um dos grandes diferenciais da programação orientada a objeto em relação a outros paradigmas de programação que também permitem a definição de estruturas e operações. Sobre essas estruturas estão os conceitos de herança e mecanismo, através dos quais definições existentes podem ser facilmente estendidas. Juntamente com a herança deve enfatizar a importância do polimorfismo, que permite selecionar funcionalidades que um programa irá utilizar de forma dinâmica, durante sua execução. “Orientação a Objeto consiste em considerar os sistemas computacionais como uma coleção de objetos que interagem de maneira organizada” (Farinelli, 2007).

2.3 JAVA

Java é uma linguagem de programação orientada a objeto desenvolvida pela Sun Microsystems. Modelada depois de C++, a linguagem Java foi projetada para ser pequena, simples e portátil a todas as plataformas e sistemas operacionais, tanto o código fonte como os binários. Esta portabilidade é obtida pelo fato de linguagem ser interpretada, ou seja, o compilador gera um código independente de máquina chamado *byte-code*. No momento da execução este *byte-code* é interpretado por uma máquina virtual JAVA (JVM), que é um programa que converte o código Java em comandos que o sistema operacional possa executar. Para portar Java para uma arquitetura hardware/s específica, basta instalar a máquina virtual (interpretador). Além de ser integrada à Internet, Java também é uma excelente linguagem para desenvolvimento de aplicações em geral. Dá suporte ao desenvolvimento de software em larga escala, segundo Java.

2.4 BANCO DE DADOS

Os bancos de dados são softwares que foram desenvolvidos com a funcionalidade principal de armazenar dados específicos de onde serão obtidas informações para a execução de determinada ação.

Os sistemas de bancos de dados são projetados para administrar grandes volumes de informações sobre uma determinada aplicação, provendo um ambiente que seja adequado e eficiente para o armazenamento e a recuperação das mesmas (Siberschatz; Korth; Sudarshan, 2006).

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, de acordo com (Date, 2003).

2.5 MySQL

Segundo o Oficina da Net (2010) o MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo.

Características:

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, Python, Perl, PHP, ASP e Ruby);
- Excelente desempenho e estabilidade;

- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um Software Livre com base na GPL;
- Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid...
- Suporta controle transacional;
- Suporta Triggers;
- Suporta Cursors (Non-Scrollable e Non-Updatable);
- Suporta Stored Procedures e Functions;
- Replicação facilmente configurável;
- Interfaces gráficas (MySQL Toolkit) de fácil utilização cedidos pela MySQL Inc.

O capítulo 2, está falando um pouco com relação as tecnologias que foram necessárias para a criação do framework gerador de SQL para replicação de dados. São tecnologias que compõem o framework assim como; a linguagem utilizada, o banco de dados, todos trazendo a qualidade e um bom funcionamento do mesmo.

3 FERRAMENTAS E TECNOLOGIAS

Neste capítulo, será especificado sobre as ferramentas e tecnologias utilizadas para o desenvolvimento do *framework*:

3.1 ECLIPSE

Segundo Silva (2011) o Eclipse é uma das ferramentas mais populares no mercado de desenvolvimento de software. Devido a sua extrema leveza, torna-se uma ferramenta ideal para trabalhar com a linguagem Java, já que a mesma emprega muitos recursos da máquina.

Destacam-se algumas funcionalidades no Eclipse, para o desenvolvimento em Java, que são:

- Auto-complete;
- Organização de *imports*;
- Auto-formatação de código;
- Mapeamento de referências;
- Compilação imediata;
- Geração de trechos de códigos;
- Incorporação de plugins: TO-DO, TASKS, etc.

O Eclipse é uma ferramenta de filosofia de código aberto (*Open-Source*).

3.2 ASTAH COMMUNITY

Segundo Souza (2009) a ferramenta Astah Community é uma boa ferramenta de modelagem UML gratuita. É desenvolvida na plataforma Java, o que garante sua portabilidade para qualquer plataforma que possui uma máquina virtual Java. Por ser uma versão Community possui algumas limitações, por ter também uma versão profissional disponível e paga, mas os recursos presentes na versão Community podem suprir a necessidade de grande parte dos artefatos necessários no dia-a-dia.

São muitas as funcionalidades dessa ferramenta. Dentre os recursos da ferramenta estão:

- Suporte a UML 2.1
- Diagramas de Classe, Caso de Uso, Sequência, Atividade, Comunicação, Máquina de Estado, Componentes, Implantação, Estrutura de Composição, Objetos e Pacotes.
- Ajustes de alinhamento e tamanho dos diagramas
- Impressão dos diagramas (com a marca d'água da ferramenta)
- Exportação das imagens dos diagramas (com a marca d'água da ferramenta)

3.3 NAVICAT

O Navicat for MySQL é uma poderosa ferramenta para a administração e desenvolvimento de servidores de banco de dados MySQL. É conhecida mundialmente, bastante famosa inclusive, utilizada por empresas conceituadas como os gigantes Google e Apple. É caracterizada por ser um aplicativo sofisticado voltado a desenvolvedores profissionais e, ao mesmo tempo, de fácil aprendizagem aos iniciantes.

O programa suporta qualquer versão do MySQL, incluindo os recursos mais recentes como *trigger*, procedimentos armazenados, funções, eventos, visualização, gerenciamento de usuário etc. Sua concebida interface gráfica possibilita a criação,

organização, acesso e compartilhamento de informações de forma fácil, rápida e segura.

Através dele se pode conectar usuários a servidores MySQL locais ou remotos, com uma série de ferramentas úteis à disposição, tais como sincronização de dados e estruturas, importação e exportação de dados, cópias de segurança e relatórios para facilitar o processo de manutenção dos dados, segundo equipe Baixaki.

3.4 FRAMEWORKS

Um *framework* consiste em um conjunto de classes que se relacionam e representam uma solução incompleta. "Um *framework* é o esqueleto de uma aplicação que pode ser customizado por um desenvolvedor da aplicação", de acordo com Fayad e Johnson (1999).

Um *framework* deve servir de base para a implementação da aplicação. Os objetos devem ser estendidos e implementados de acordo com o domínio de problema particular, segundo Assis (2003). "*Frameworks* fornecem um mecanismo para obter a reutilização e são bem apropriados para domínios onde várias aplicações similares são construídas várias vezes, partindo-se apenas de ideias", Fiorini (2001).

3.4.1 TIPOS DE FRAMEWORKS

Segundo Fayed e Johnson (2000) *framework* são mais do que projeto ou arquitetura, tratam-se de uma aplicação semi-completa contendo componentes estáticos e dinâmicos que podem ser adaptados para produzir aplicações específicas do usuário. Ainda, consideram existir, de acordo com a forma de reuso, três tipos de *frameworks*: caixa branca, caixa preta e caixa cinza.

3.4.1.1 CAIXA-BRANCA

Os *frameworks* de Caixa Branca são os *frameworks* que se caracterizam pela presente existência da orientação a objetos e ligação através do reuso que se dá por meio da herança, MALDONADO (2009).

3.4.1.2 CAIXA-PRETA

Os *frameworks* caixa-preta são baseados em componentes de software. A extensão da arquitetura é feita a partir de interfaces definidas para componentes. Os recursos existentes são reutilizados por meio de: definição de um componente adequado a uma interface específica e integração de componentes em um *framework* que utiliza padrões de projeto como o Strategy.

3.4.1.3 CAIXA CINZA

Já o framework caixa cinza consiste da mescla entre os frameworks caixa branca e caixa preta sendo que o reuso é obtido por herança, por ligação dinâmica e por interfaces de definição.

3.5 FRAMEWORKS UTILIZADOS

3.5.1 HIBERNATE

O Hibernate é um framework *open source* de mapeamento objeto/relacional desenvolvido em Java, ou seja, ele transforma objetos definidos pelo desenvolvedor

em dados tabulares de uma base de dados, portanto com ele o programador se livra de escrever uma grande quantidade de código de acesso ao banco de dados e de SQL, segundo (Ortoncelli, Lazarin 2009).

3.5.2 VANTAGENS DO HIBERNATE

A utilização de código SQL dentro de uma aplicação agrava o problema da independência de plataforma de banco de dados e complica, em muito, o trabalho de mapeamento entre classes e banco de dados relacional. O Hibernate abstrai o código SQL da nossa aplicação e permite escolher o tipo de banco de dados enquanto o programa está rodando, permitindo mudar sua base sem alterar nada no seu código Java.

Além disso, ele permite criar suas tabelas do banco de dados de um jeito bem simples, não se fazendo necessário todo um design de tabelas antes de desenvolver seu projeto que pode ser muito bem utilizado em projetos pequenos. O Hibernate não apresenta apenas a função de realizar o mapeamento objeto relacional. Também disponibiliza um poderoso mecanismo de consulta de dados, permitindo uma redução considerável no tempo de desenvolvimento da aplicação, segundo (Ortoncelli, Lazarin 2009).

3.5.3 API REFLECTION

A reflexão é uma ferramenta poderosa, que oferece aos programadores a capacidade de analisar as propriedades das classes e dos objetos, manipulando-as diretamente. Reflexão é definida pela própria Sun (ORACLE, 2011) como sendo a capacidade de um programa de examinar ou modificar o comportamento de aplicações rodando na Java Virtual machine em tempo de execução, de acordo com Linden (2011).

4 PROPOSTA DE TRABALHO

4.1 INTRODUÇÃO

O objetivo deste trabalho é o desenvolvimento de um *Framework Gerador de SQL para Replicação de Dados*. O mesmo, auxiliará sistemas de empresas que possuem filiais, uma vez que realizará as transações entre as mesmas de forma objetiva, possibilitando um controle dessas informações entre elas.

Os sistemas que forem utilizar a *Framework Gerador de SQL para Replicação de Dados* serão fornecidos para a utilização da mesma, três tabelas onde serão carregadas todas as instruções SQLs realizadas no sistema. O *Framework Gerador de SQL para Replicação de Dados* será uma *thread*, de forma que funcionará em um fluxo único de controle sequencial dentro de um programa, onde o mesmo irá verificar se à SQLs para a realização de replicação entre as empresas. Uma vez que houver uma falha na conexão com a internet, o *framework replicador* continuará tentando enviar as SQLs até que volte a conexão.

4.2 CATEGORIZAÇÃO DO FRAMEWORK REPLICADOR

Dentre as classificações básicas de *frameworks* pode-se dizer que o objetivo deste trabalho é um *framework* de domínio do tipo caixa-branca. É caixa-branca no sentido em que customizações das ações básicas oferecidas pelo *framework* podem ser feitas através da herança como a classe DAO, pela presente existência da orientação a objetos e também por não existir uma definição de interfaces, já que isso é definido em *framework* caixa preta e o *framework gerador de SQL para replicação de dados* não terá interação com o usuário.

4.3 FUNCIONAMENTO GERAL DO *FRAMEWORK GERADOR DE SQL PARA REPLICAÇÃO DE DADOS*

O *framework gerador de SQL para replicação de dados* será dividido em duas partes, sendo uma o *framework gerador de SQL* e a outra o *replicador de dados*. O *framework gerador de SQL* irá pegar os logs gerados pelos sistemas onde estarão integrados, fazendo assim a construção das SQLs. Após a construção das SQLs, o *framework* irá cadastrar na tabela do bando de dados *SqlAtualizações* fazendo com que todos os dados para a replicação estejam guardados, esse processo ocorrerá em cada sistema sendo a empresa principal e também nas filiais.

Após o *framework gerador de SQL* gravar as SQLs na tabela *SqlAtualizações* o *replicador* irá percorrer a tabela *SqlAtualizações* para verificar se as SQLs cadastradas pelo *framework gerador de SQL* e verificar se ainda não estão “finalizadas”, assim também ocorrerá com a tabela *SqlServidor*, onde estarão cadastrados todos os servidores relacionado a empresa cujo sistema será onde irão estar integrado o *framework gerador de SQL para replicação de dados*. Os servidores por sua vez serão cadastrados manualmente direto no banco de dados pelo proprietário do *framework gerador de SQL para replicação de dados*.

Depois de todo o processo realizado, o replicador de dados organizará na tabela do banco a *SqlExecuta*, cada SQL para cada servidor. Após a organização das SQLs e dos servidores cadastrados na tabela *SqlServidor*, o replicador de dados irá se conectar com os demais servidores para a atualização e replicará da tabela *SqlExecuta* de tempo em tempo sendo o mesmo determinado pelo proprietário do *framework*, de forma continua como uma *thread*, assim de um servidor para o outro mantendo a comunicação e atualização entre os “n” bancos de dados. Na Figura 1 é ilustrado como funcionarão as replicações entre os servidores.

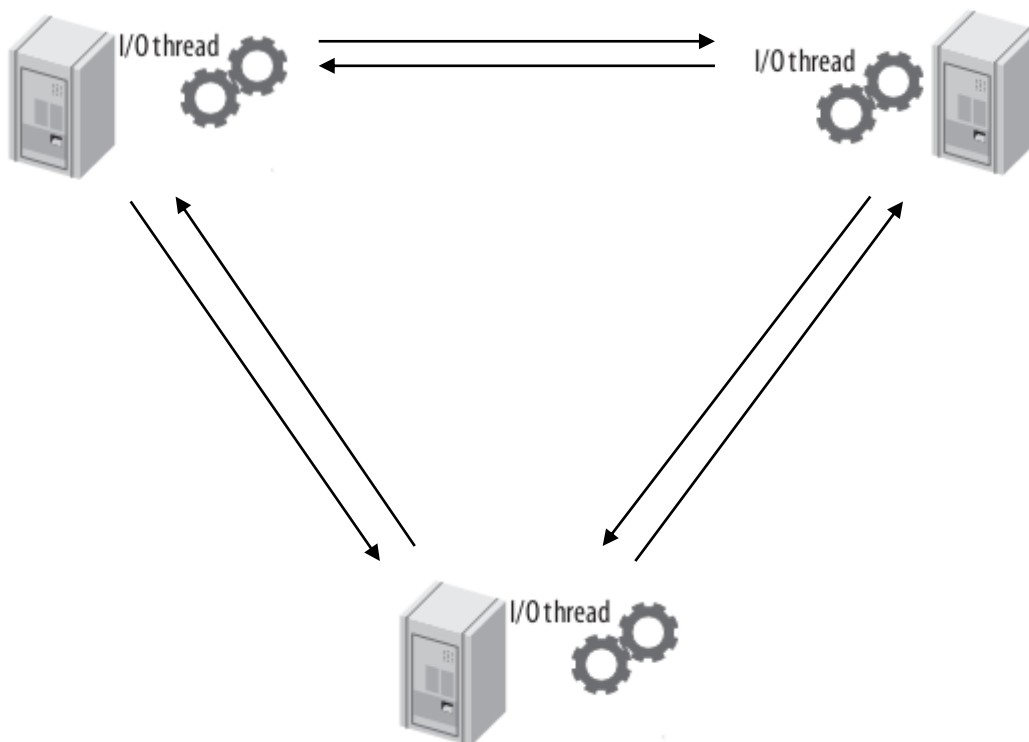


Figura 1 – Estrutura da Replicação

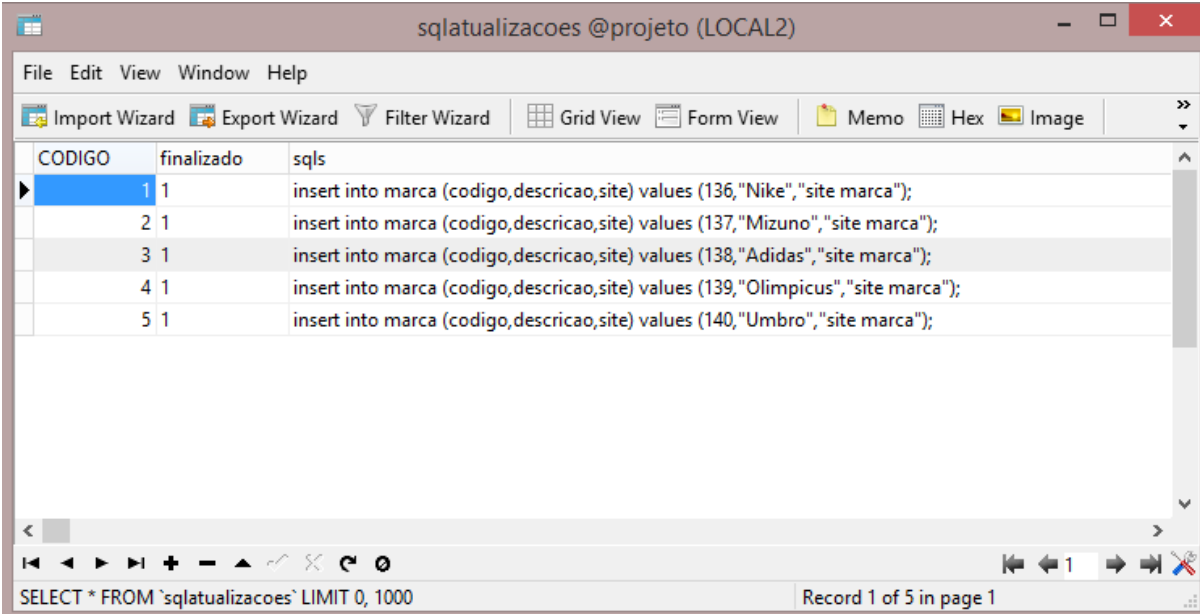
De acordo com a figura 1, o funcionamento do framework gerador de SQL para replicação de dados, está ilustrando três bancos de dados diferentes sendo assim como se houvesse uma empresa local sendo ela a principal e as outras as filiais. O funcionamento através das threads possibilita o envio dos dados de uma para a outra de forma que cada empresa terá instalado o framework assim atualizando as informações geradas em cada uma delas umas às outras.

5 ESTUDO DE CASO

Neste capítulo, estará descrevendo a forma de funcionamento do *Framework Gerador de SQL para Replicação de Dados* quando estiver instalado em algum sistema:

5.1 TABELA SQLATUALIZAÇÕES

Na tabela `sqlatualizações`, ficarão armazenadas todas as SQLs geradas pelo framework gerador de SQL, o mesmo irá cadastrar através dos logs gerados pelo sistema onde o framework estará integrado, assim todas as movimentações relacionadas no sistema irão estar na tabela `sqlatualizações` de modo que ficarão salvas aguardando a verificação do replicador de dados. Na Figura 2 está sendo ilustrado a tabela `sqlatualizações`.



CODIGO	finalizado	sqls
1	1	insert into marca (codigo,descricao,site) values (136,"Nike","site marca");
2	1	insert into marca (codigo,descricao,site) values (137,"Mizuno","site marca");
3	1	insert into marca (codigo,descricao,site) values (138,"Adidas","site marca");
4	1	insert into marca (codigo,descricao,site) values (139,"Olimpicus","site marca");
5	1	insert into marca (codigo,descricao,site) values (140,"Umbro","site marca");

Figura 2 – Tabela `SqlAtualizações`

Na tabela sqlatualizações, todas as SQLs que estiverem gravadas esperados á sua execução, são todas geradas pelo framework que faz o tratamento dos logs que uma vez foi gerado pelo sistema e são todas organizadas nesta tabela contendo informações na coluna “finalizado” se a SQL já foi finalizada ou não sendo representada por “0”, não finalizada e por “1” finalizada.

5.2 TABELA SQLSERVIDOR

Na tabela sqlservidor ficarão cadastrados todos os servidores que estarão disponíveis pela empresa, podendo ser estes servidores filiais que estará sendo necessário as transações para atualização de seus Bancos de Dados. Na Figura 3 está sendo ilustrado a tabela sqlservidor.

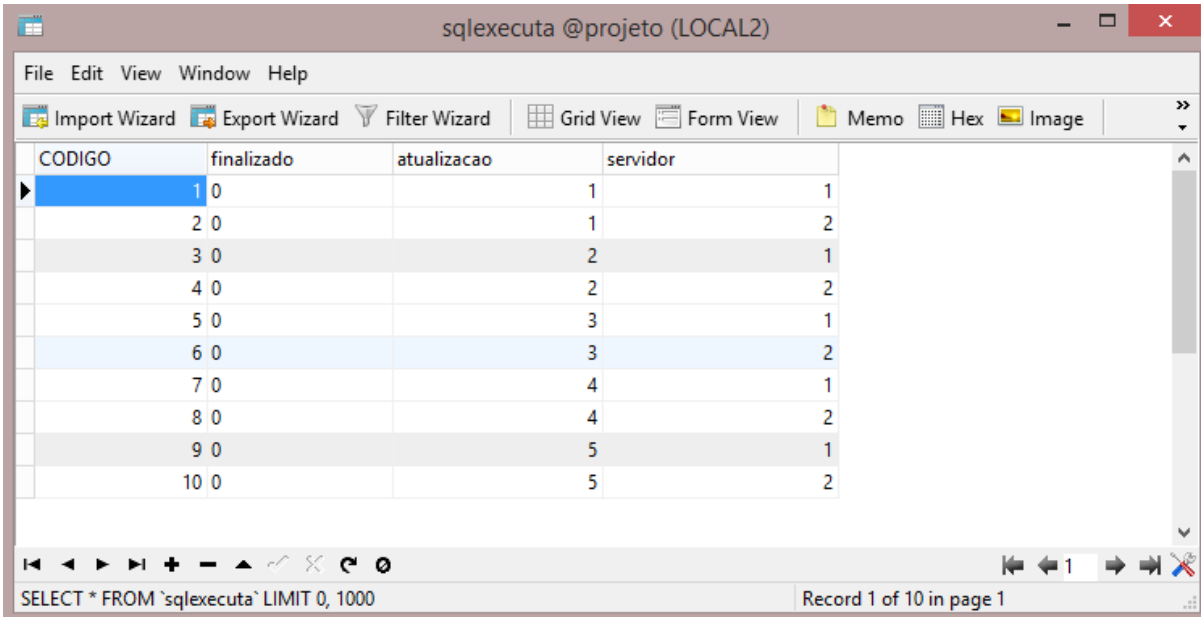
CODIGO	cancelado	host	senha	usuario	servidor
1	N	jdbc:mysql://localhost/servidor	123	root	1
2	N	jdbc:mysql://localhost/servidor1	123	root	2

Figura 3 – Tabela SqlServidor

Na tabela sqlservidor, todos os servidores serão cadastrados diretamente no banco de dados, sendo representado pela coluna “cancelado”, onde será representado por “N” indicando não cancelado e por “S” cancelado. Além da informação anterior serão indicados a conexão, senha, o tipo de usuário e também o código do servidor para melhor organização dentro da tabela sqlexecuta.

5.3 TABELA SQLEXECUTA

Na tabela sqlexecuta, é onde serão realizadas as replicações e enviadas todas as SQLs geradas pelo sistema para os servidores cadastrados. Mas antes dessa replicação, o replicador irá separar cada SQL gerada para cada servidor cadastrado assim, verificando se os servidores não estão cancelados, verificando se as SQLs cadastradas na sqlatualizações não estão finalizadas e após esse processo estará organizando todas dentro da tabela sqlexecuta. Na Figura 4 está sendo ilustrado a tabela sqlexecuta.



CODIGO	finalizado	atualizacao	servidor
1 0			1
2 0			1
3 0			2
4 0			2
5 0			3
6 0			3
7 0			4
8 0			4
9 0			5
10 0			5

Figura 4 – Tabela SqlExecuta

Na tabela sqlexecuta cada SQL cadastrada irá ser organizada pelo seu código e assim sendo separada a mesma para cada servidor distribuindo assim as informações e fornecendo as atualizações necessárias para manter as bases de dados iguais.

A instalação do Framework Gerador de SQL para Replicação de Dados, será fácil pois uma vez que fornecido o arquivo de extensão JAR, onde estão as tabelas para acrescentar no banco de dados do sistema. Para que os cadastros das SQLs sejam efetuados de maneira correta o sistema terá que instanciar o DAO genérico do framework onde o mesmo será chamado pelo sistema após um comando de tela do mesmo sendo ele INSERT, UPDATE ou DELETE, e o framework fará com que seja realizado todo o processo de gerar a SQL para cadastrar na tabela SqlAtualizações através do Hibernate. O Replicador será instalado e uma vez executado para a realização das replicações para atualizar os servidores e se houver a necessidade de interromper o replicador, é só parar o processo.

Quando o sistema na qual será integrado o framework gerador de SQL para replicação de dados realizar um cadastro, o DAO do sistema irá instanciar o DAO genérico do framework que tem todos os métodos com cadastrar, alterar e excluir. A Figura 5 está ilustrando o método de cadastrar do DAO.

```
public abstract class DAO<E> {  
  
    @SuppressWarnings( "unchecked")  
    private Class clazz;  
  
    @SuppressWarnings("unchecked")  
    public DAO(Class clazz){  
        this.clazz = clazz;  
    }  
  
    public void cadastrar(E objeto) throws Exception {  
        Session session = null;  
        Transaction transaction = null;  
        try {  
            session = HibernateUtil.getSession();  
            transaction = session.beginTransaction();  
            GeraSql<E> log = new GeraSql<E>();  
            objeto = log.retornaCodigo(objeto);  
        }  
    }  
}
```



```
        session.save(objeto);
        transaction.commit();
        session.flush();
    } catch (Exception exe) {
        System.out.println(exe);
        if (transaction != null && !transaction.wasCommitted() &&
!transaction.wasRolledBack()) {
            transaction.rollback();
        }
        throw exe;
    } finally {
        if (session != null && session.isOpen()) {
            session.close();
        }
    }
}
```

Figura 5 – Método DAO

Na classe GeraSql, é onde as SQLs são geradas de acordo com a classe e os atributos. O Reflection é utilizado para saber os dados do objeto preenchido e saber os atributos dos objetos. A Figura 6 está ilustrando a forma de montar a SQL insert.

```

String sql = "insert into "+nomeClasse+" (";

    for (int i = 0; i < claz.getDeclaredFields().length; i++) {
        Field f = claz.getDeclaredFields()[i];
        if(ReflectionUtil.temAnotacaoNoCampo(f, Transient.class)) {
            continue;
        }
        sql += f.getName()+",";
    }
    sql = sql.substring(0,sql.length()-1);
    sql = sql +")";

    sql = sql + " values (";

    for (int i = 0; i < claz.getDeclaredFields().length; i++) {
        Field f = claz.getDeclaredFields()[i];
        if(ReflectionUtil.temAnotacaoNoCampo(f, Transient.class)) {
            continue;
        }
        sql += pegaTipoAtributo(f, objeto)+",";
    }

    sql = sql.substring(0,sql.length()-1);
    sql = sql + ")";
    System.out.println(sql);
    gravaSql(sql);

    return objeto;
}

```

Figura 6 – Código de Gerar SQL INSERT

Para grava a SQL gerada o método gravaSql, instancia a classe SqlAtualizações e SET falso para a coluna finalizado e também usa o SqlAtualizaçõesDAO para cadastrar no banco. A Figura 7 está ilustrando o método gravaSql.

```
public void gravaSql(String sql1){
    SqlAtualizacoes sql = new SqlAtualizacoes();
    sql.setFinalizado(false);
    sql.setSql(sql1);
    try {
        new SqlAtualizacoesDAO().cadastrar(sql);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 7 – Método de Grava SQL

Com a instalação do framework gerador de SQL para replicação de dados, o sistema tem a tranquilidade quanto a replicação uma vez que fará o trabalho de armazenamento de tudo o que está sendo realizado no mesmo pelo motivo em que toda a realização do sistema seja ele INSERT, UPDATE ou DELETE, por mais que seja interrompido a conexão com a internet ficarão salvos na tabela SqlAtualizações para que ao retornar essa conexão o replicador termine de enviar todas para os outros servidores.

Neste capítulo pode-se observar a forma em que o framework gerador de SQL para replicação de dados funciona, algumas tabelas que estão ilustrando o funcionamento do mesmo no banco de dados explicando a relação de cada uma delas para o funcionamento e quais suas funções. Este capítulo também demonstra através de códigos usados para a criação do framework um pouco de como foi realizado algumas tarefas na qual o framework realiza.

6 PLANEJAMENTO DO PROJETO

6.1 ESTRUTURAS ANALÍTICAS DO PROJETO (WBS)

Para o desenvolvimento do FRAMEWORK GERADOR DE SQL PARA REPLICAÇÃO DE DADOS elaborou-se a seguinte Estrutura Analítica do Projeto:

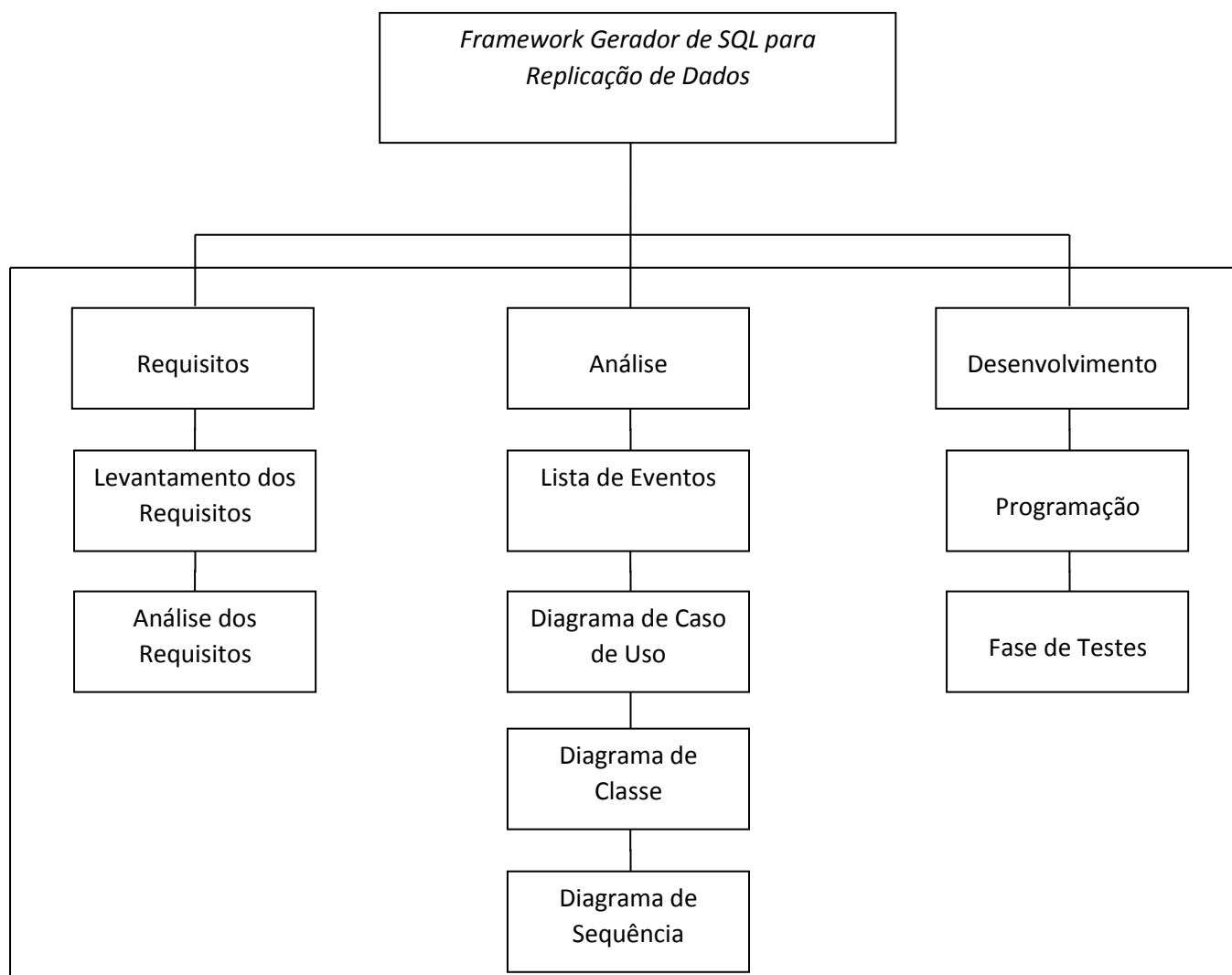


Figura 8 – Estrutura Analítica do Projeto (WBS)

6.2 SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS

O Sequenciamento de Atividades do projeto envolve estabelecer uma relação lógica das atividades a serem desenvolvidas ao longo do projeto FRAMEWORK GERADOR DE SQL PARA REPLICAÇÃO DE DADOS:

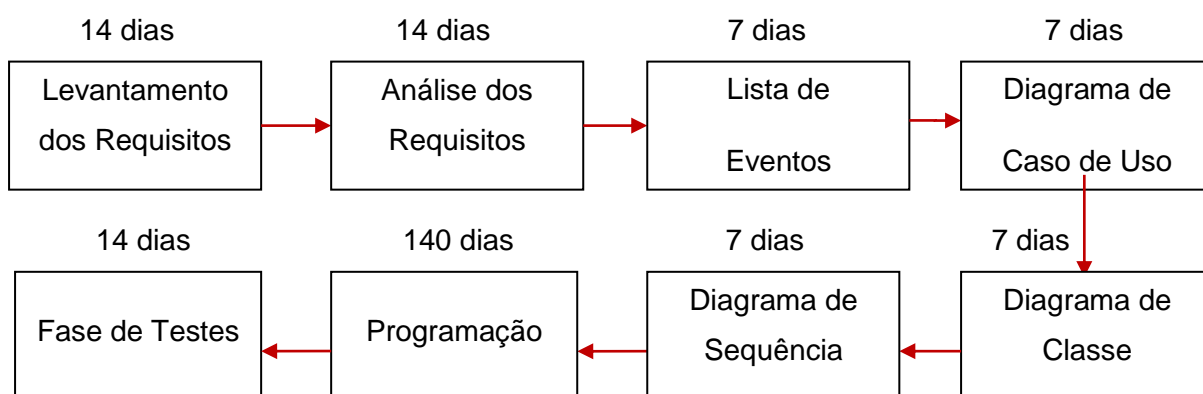


Figura 9 – Sequenciamento das Atividades Definidas

6.3 CRONOGRAMA DE ATIVIDADES

Mês	Janeiro				Fevereiro				Março				Abril				Maio				Junho				Julho				Agosto							
Semana	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Levantamento dos Requisitos	X	X																																		
Análise de Requisitos			X	X																																
Lista de Eventos							X																													
Diagrama UC							X																													
Diagrama de Classe								X																												
Diagrama de Sequência											X																									
Programação									X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				

O caso de uso está ilustrando o funcionamento do replicador, uma vez que executado o replicador funcionará como uma thread enviando e atualizando assim as bases de dados entre os servidores e também a finalização do mesmo caso necessário. A Figura 10 está ilustrando o Caso de Uso.

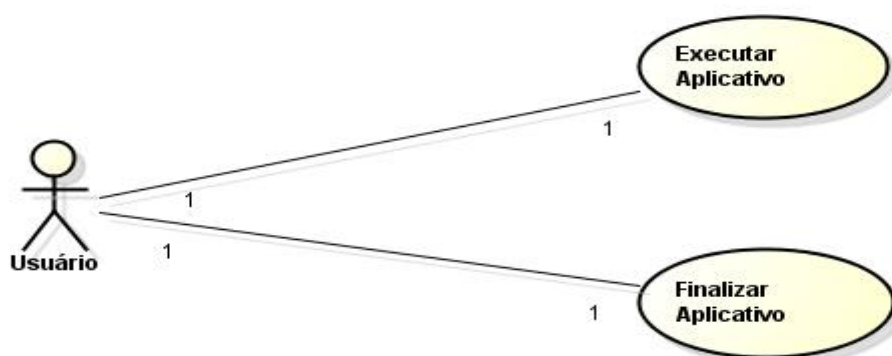


Figura 10 – Caso de Uso

6.7 DIAGRAMA DE CLASSE

O diagrama de classe está ilustrando o funcionamento das tabelas relacionadas ao framework. A tabela `SqlAtualizações`, será a que receberá todas as SQLs geradas pelo framework após receber os logs gerados pelo sistema. Já a tabela `SqlServidor` representa a tabela onde serão cadastrados todos os servidores para a realização das replicações. A tabela `Sqlexecuta` receberá através do replicador todas as SQLs e todas os servidores que foram cadastrados e assim formando este tipo de ligação. Na Figura 11, está ilustrando o Diagrama de classe.

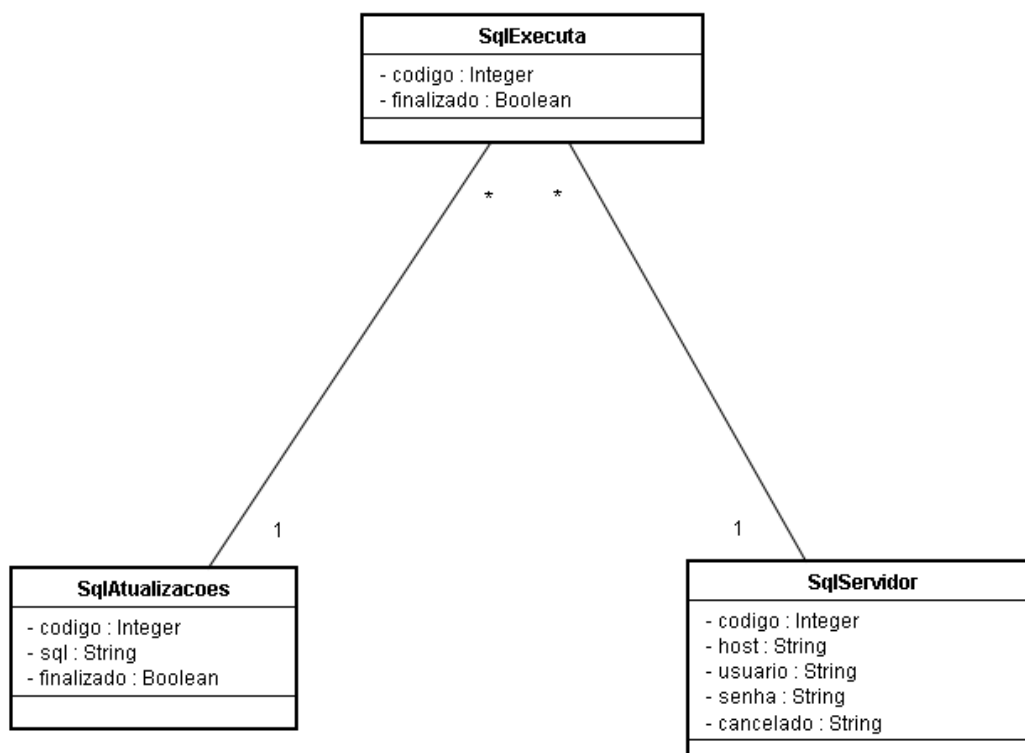


Figura 11 - Diagrama de Classe

O capítulo 6, tem como objetivo mostrar como foi estruturado o projeto framework gerador de SQL para replicação de dados mostrando assim, o planejamento do mesmo com relação ao tempo total para criação do framework, as estimativas de custo ou seja este capítulo informa a parte inicial do projeto.

7 CONCLUSÕES

Após a grande jornada de trabalho sobre esta documentação, venho reconhecer que é importantíssimo tal trabalho, já que sem o mesmo não estaria compreendendo sua importância. Todo conhecimento que adquiri não somente pelas pesquisas, estudos mais sim também através da pratica por certo estão comigo pelo resto de minha vida.

O desenvolvimento do Framework Gerador de SQL para Replicação de Dados trouxe grande conhecimento, aja vista que não tinha experiência e com isso pude conhecer um pouco da linguagem Java e um pouco de suas inúmeras funcionalidades como Reflection, fora que a prática mostrou grandes obstáculos e muitas dificuldades.

Em relação ao Framework desenvolvido acredito que ainda mais empresas precisaram de um framework dessa qualidade e característica, visto que não é natural um framework deste tipo, já que outros tipos de replicações são realizados pelos bancos de dados e não por um framework.

Tenho planos futuros para o framework desenvolvido pois já conheci empresas que precisaram de um framework deste tipo para fazer a atualização entre os bancos de dados. A criação de interfaces gráficas para gerenciar o replicador em busca de modificar formas de trabalhar com o mesmo e também realizar uma parametrização das configurações das replicações.

REFERÊNCIAS

Apostila de Java, Disponível em <<http://www.cm-braganca.pt/document/448112/504883.pdf>>. Acessado em 14/02/2014.

ASSIS, S. R.; SUZANO, R. Framework: Conceitos e Aplicações. Científico, Salvador, v.2, Jun./Dez. 2003.

BAUER, Christian; KING, Gavin. Hibernate in action. Greenwich: Manning Publications, 2005.

BAUER, Christian, KING, Gavin. Java Persistence com Hibernate. Rio de Janeiro, Editora Ciência Moderna Ltda, 2007. 844p.

CRUZ, Elifranio Alves. Hibernate Conceitos. Slideshare, Disponível em <<http://www.slideshare.net/elifranio/hibernate-conceitos>> Acessado em 25/10/2013.

DATE, Christopher J.; Introdução a Sistemas de Banco de Dados. Tradução de Daniel Vieira. 8. ed. Rio de Janeiro: Editora Elsevier, 2003.

DESTRO, Daniel. Java e Reflection. Grupo de Usuários Java, GUJ. Disponível em <<http://www.guj.com.br/articles/10>> Acessado em 15/10/2013.

Ensino e Inovação, Caelum, <http://www.caelum.com.br>. Acesso em 12/10/2013.

FARINELLII, Fernanda. Conceito Básicos de Programação Orientada a Objetos, Disponível em <http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1662272077_POO.pdf> Acessado em 18/02/2014.

FAYAD, M. E.; JOHNSON, R. E. Domain-specific application frameworks: Frameworks experience by industry. First ed. John Wiley & Sons, 2000. Disponível em <<http://aberto.univem.edu.br/bitstream/handle/11077/438/An%C3%A1lise%20e%20Projeto%20de%20um%20Framework%20Baseado%20em%20uma%20Linguagem%2>

Ode%20Padr%C3%B5es%20no%20Dom%C3%ADnio%20de%20Software%20Educacional.pdf?sequence=1>. Acessado em 10/03/2014.

FAYAD, W.E.; SCHMIDT, D.C.; JOHNSON, R.E. Building Application. Frameworks: object-oriented foundations of framework design. New York, USA : John Wiley & Sons, 1999.

FIORINI, S. T. Arquitetura para Reutilização de Processos de Software. 2001. Tese (Doutorado em Informática) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

FRANÇA, L.; STAA, A. Geradores de Artefatos: Implementação e Instanciação de frameworks. Simpósio Brasileiro de Engenharia de Software, 15. SBC, 2001. PP 302-315

Linden, Ricardo. MundoJ a revista para quem gosta de desenvolver software número 46, Ano VIII, Marc/Abr 2011

MALDONADO, José Carlos; BRAGA, Rosa Terezinha Vaccare, Germano, Fernão Stella Rodrigues; MASIERO, Paulo Cesar. Padrões e framework de Software. São Paulo: Universidade de São Paulo, 37 p. (Notas Didáticas). Disponível em: <http://www.cesumar.br/epcc2009/anais/lincoln_fernandes_paulino_santos.pdf>. Acessado em 09/03/2014.

OFICINA DA NET. MySQL- o que é?. Disponível em <http://www.oficinadanet.com.br/artigo/2227/mysql_-_o_que_e>. Acessado em 01/03/2014.

ORTONCELLI, André Roberto; LAZARIN, Carlos Albert Jóia. Persistência de Dados com Java e Hibernate Annotations. Disponível em: <http://www.atualizado.com.br/semfronteiras/index.php?option=com_docman&task=doc_download&gid=8&&Itemid=66>. Acessado em 05/03/2014.

PAGE-JONES, Meilir., FECCHIO, Mario M. O que todo Programador deveria saber sobre Projeto Orientado a Objeto. 1. ed. Sao Paulo: Makron Books, 1997.

PISA, Pedro. O que é e como usar o MySQL?.TechTudo, Artigos. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>> Acessado em 20/10/2013.

Significados.com.br. O que é Java. Disponível em <<http://www.significados.com.br/java/>>. Acessado em 27/10/2013.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistemas de Bancos de Dados. 5. ed. Tradução de Daniel Vieira. Rio de Janeiro: Editora Elsevier, 2006.

SILVA, Eduardo Nicolini Sodre da. Desenvolvimento do framework Java-Fácil. 2011. 50p. Trabalho de Conclusão de Curso – Fundação Educacional do município de Assis – FEMA/Instituto Municipal de Ensino Superior de Assis – IMESA.

SOUZA, Luiz Gustavo S. de. JudeCommunity/ Astah Community. Disponível em <<http://luizgustavoss.wordpress.com/tag/astah-community/>>. Acessado em 13/03/2014.

TUTORIAL. Guia Prático de utilização da ferramenta Astah Community. Disponível em <<http://www-pet-si.inf.ufsm.br/images/consultoriodesoftware/Astah.pdf>>. Acesso em 12/03/2014.

VASCH, Alberto. História das Linguagens de Programação. Disponível em <<http://pt.slideshare.net/AlbertoVach/lingugens-de-programao>>. Acessado em 20/07/2014.

<http://www.dca.fee.unicamp.br/projects/sapiens/Reports/ExplorAnot2/componentFramework.htm>. Acessado em 10/03/2014.

<http://www.baixaki.com.br/download/navicat-for-mysql.htm>. Acessado em 13/03/2014.