

Processo:	342 891
Folha:	553
Func:	605



PROJETO SREI

Sistema de Registro Eletrônico Imobiliário

PA 2.4.3 - Relatório sobre definição do ambiente de execução do SREI

Título	PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução
Versão	Versão 1.0 release 18
Data da liberação	04/02/2012
Classificação	LSI-TEC:Restrito
Autores	Adilson Hira, Moacir Campos, Natanael Menezes, Leandro Moraes, Daniel Savoy
Propriedade	LSI-TEC
Restrições de acesso	LSI-TEC e CNJ

Sumário

1	INTRODUÇÃO.....	3
2	SERVIDOR DE APLICAÇÃO.....	5
2.1	LINGUAGEM PARA O SERVIDOR DE APLICAÇÃO.....	5
2.2	PHP.....	6
2.3	JAVA ENTERPRISE EDITION (JEE).....	7
2.4	DISCUSSÃO ENTRE SERVIDORES JEE E PHP.....	9
2.5	PRINCIPAIS SERVIDORES JEE.....	9
2.5.1	JBoss.....	10
2.5.2	Tomcat.....	11
2.5.3	Glassfish.....	12
2.5.4	Jetty.....	12
2.5.5	Geronimo.....	13
2.6	COMENTÁRIOS E DISCUSSÃO DOS SERVIDORES DE APLICAÇÃO.....	13
3	SGBD.....	15
3.1	POSTGRESQL.....	15
3.2	MYSQL.....	16
3.3	CONSIDERAÇÕES SOBRE O SGBD.....	17
4	SISTEMA DE BACKUP.....	18
4.1	BACKUP.....	18
5	REDUNDÂNCIA E ALTA DISPONIBILIDADE.....	20
5.1	POSTGRESQL.....	21
5.2	MYSQL.....	22
6	SISTEMA OPERACIONAL.....	24
7	DISCUSSÃO.....	27
8	RECOMENDAÇÃO.....	28
	REFERÊNCIAS.....	29

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	2 / 29

1 Introdução

Este documento descreve o levantamento das alternativas técnicas de arcabouços de software para definição do ambiente de execução do Sistema de Registro Eletrônico Imobiliário (SREI) a ser especificado para operação no Estado do Pará.

Este levantamento é direcionado ao uso de Software Livre, o que minimiza custo de licenças de software, e estando aderente às diretrizes do Governo no incentivo a adoção de software livre para o desenvolvimento de tecnologia de informação no setor público.

Particularmente, o Software Livre tem em sua essência alguns componentes estruturais, entre eles:

- uso de padrões abertos;
- licenciamento livre dos softwares;
- formação de comunidades, em especial de usuários e desenvolvedores.

O ambiente de exceção do SREI deverá estar aderente ao modelo de arquitetura 3-camadas, tanto para a sua "Camada de Concentração do SREI" como para a "Camada de Núcleo", de acordo com proposto pela PA2.3 "Alternativas de Arquitetura do Sistema". A arquitetura é composta pelas seguintes camadas:

- Camada de Apresentação (Navegador WEB);
- Camada de Negócio (Servidor de Aplicação);
- Camada de Persistência (Servidor Gerenciador de Banco de Dados).

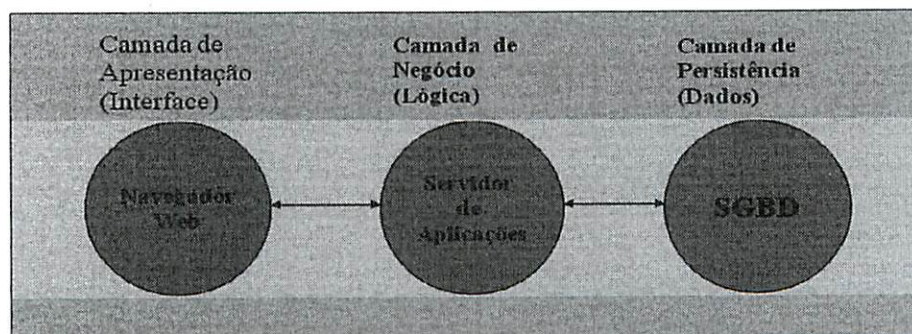


Figura 1: Modelo de arquitetura em 3-Camadas.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	3 / 29

Além do modelo 3-camadas, também são consideradas outras questões importantes do Ambiente de Execução do SREI, tais como: *back-up*, contingência e disponibilidade.

Dessa forma, serão considerados neste trabalho os seguintes tipos de arcabouços de execução de software:

- Servidor de aplicação (App);
- Servidor Gerenciador de Banco de Dados (SGBD);
- Sistema de *backup* e de armazenamento;
- Redundância e alta disponibilidade;
- Sistema operacional.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	4 / 29

2 Servidor de Aplicação

Um servidor de aplicação (App) ou em inglês, *application server* (AS), é um servidor que disponibiliza um ambiente para a instalação e execução de certas aplicações. Os servidores de aplicação também são conhecidos como software de *middleware*.

2.1 Linguagem para o Servidor de Aplicação

Nestes últimos anos, estão ocorrendo intensos debates sobre quais seriam as melhores linguagens de programação para desenvolvimento Web. Atualmente, há duas opções mais utilizadas de linguagens para servidores de aplicação baseadas em software Livre: Java/JEE e o PHP:

- Java/J2EE ou JEE) - Linguagem do estado da arte no desenvolvimento de software, com a implementação de conceitos como orientação a objetos, controle, segurança, persistência de dados, arquitetura n camadas (n-tier),
- PHP - Linguagem que se caracteriza pela simplicidade e a velocidade no desenvolvimento web, e ainda utilizando conceitos simplificados de orientação a objetos, arquitetura livre e simples, pouca dependência.

Uma das questões importantes a serem consideradas na escolha do Servidor de Aplicação é sua aderência à orientação a objetos, que é um paradigma atual de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos.

A análise e projeto orientados a objetos têm como meta identificar o melhor conjunto de objetos para descrever um sistema de software. O funcionamento deste sistema se dá através do relacionamento e troca de mensagens entre estes objetos.

Na programação orientada a objetos, implementa-se um conjunto de classes que definem os objetos presentes no sistema de software. Cada classe determina o comportamento (definido nos métodos) e estados possíveis (atributos) de seus objetos, assim como o relacionamento com outros objetos.

Entre as vantagens da Orientação a Objetos podem ser citados:

- Maior facilidade para reutilização de código e por consequência do projeto;

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	5 / 29

- Possibilidade do desenvolvedor trabalhar em um nível mais elevado de abstração;
- Utilização de um único padrão conceitual durante todo o processo de criação de software;
- Maior facilidade de comunicação com os usuários do sistema e com outros profissionais de informática;
- Maior adequação à arquitetura cliente/servidor e 3-camadas;
- Ciclo de vida mais longo para os sistemas (desenvolvimento, manutenção e melhorias contínuas para o software);
- Desenvolvimento acelerado de sistemas;
- Possibilidade de se construir sistema muito mais complexos, e o re-uso de padrões e componentes de software.

Mas, entre as desvantagens podem ser citadas:

- Complexidade no aprendizado para desenvolvedores de linguagens estruturadas;
- Maior esforço na modelagem de um sistema OO do que estruturado (porém haveria menor esforço de codificação, o que torna-se uma vantagem);
- Dependência de funcionalidades no caso da herança, implementações espalhadas em classes diferentes.

2.2 PHP

O PHP (um acrônimo recursivo para "PHP: *Hypertext Preprocessor*", originalmente *Personal Home Page*.) é uma linguagem interpretada livre e muito utilizada para gerar conteúdo dinâmico na *World Wide Web*.

A grande desvantagem do PHP sempre foi a falta de aderência à orientação a objetos, mais estruturada na versão 4, sendo parcialmente resolvido a partir da versão 5. Porém, não ainda apresenta uma forte orientação a objetos, como o Java.

Algumas desvantagens da linguagem PHP são:

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	6 / 29

- Não possui sobrecarga de métodos, ou seja, não pode ter um método com mesmo nome e com parâmetros diferentes;
- Não existe obrigação de utilizar a orientação a objetos no código, podendo se trabalhar ainda no paradigma da forma procedural;
- Não existe tratamento de exceções nos métodos nativos;
- Linguagem puramente interpretada, os erros no PHP aparecem apenas em tempo de execução. No Java, por ser uma plataforma híbrida, onde primeiramente o código é compilado em bytecodes, muitos erros podem já ser detectados em tempo de compilação;
- Sendo mais flexível, mais rapidamente um desenvolvedor pode aprender a linguagem e já sair desenvolvendo, embora não utilize padrões específicos se não se aprofundar no aprendizado. Comparativamente, o Java tem uma curva de aprendizado é um pouco maior que o PHP, tendo que seguir determinados padrões (como a orientação a objetos).

Além disso, pelo PHP apresentar esta maior flexibilidade em termos de codificação, muitas vezes acelera o processo de desenvolvimento para pequenas e médias aplicações, de complexidade mais baixa, desde que não haja grandes integrações, além do que a futura manutenção não seja tão crítica. A linguagem Java acaba tendo maior aceitação do que o PHP quando se trata do fator manutenibilidade. Num ciclo de vida de projetos longo, os projetos com PHP têm maiores esforços para realização de manutenção e melhorias.

Muitos serviços de hospedagem apresentam seus custos mais baixos para um servidor PHP (como o Apache com o plug-in para PHP, por exemplo) devido à menor complexidade. O que facilita aos usuários aderirem a linguagem PHP no caso de pequenos sistemas.

2.3 Java Enterprise Edition (JEE)

O Java Enterprise Edition (JEE) é baseado na linguagem Java, que é umas das linguagens mais utilizadas no mercado, sendo que grande parte do motivo deste

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	7 / 29

sucesso corresponde ao princípio de multiplataforma (*write once, run everywhere*), com o foco na portabilidade da linguagem sobre diversas plataformas. Sendo compilada em bytecodes e interpretada na Máquina Virtual Java, ou Java Virtual Machine (JVM), a linguagem permite a construção de programas que rodem nas mais diversas plataformas, como celulares, internet, diversos sistemas operacionais e outros, sem muita necessidade de se precisar fazer ajustes no código.

Além do disso, quanto ao fator escalabilidade, o Java tem como uma vantagem sobre o PHP, por ser uma linguagem muito mais robusta. Assim, Java é mais adequado para sistemas maiores, que vão possuir um ciclo de vida e manutenção mais longo. Com a boa estruturação da linguagem, torna-se mais fácil realizar manutenções posteriores. Além disto, uma grande vantagem da linguagem Java é a uso de orientação a objetos bem estruturada, com todos os conceitos da Orientação a Objetos (OO) incorporados, como herança, interface e sobrecarga de métodos.

A utilização do JEE permite uma boa estrutura de suporte aos *Web Services* e de seus servidores web disponíveis, compõe importantes fatores para melhorar a escalabilidade da linguagem.

A plataforma JEE disponibiliza padrões para os containers Web e EJB. O Tomcat é um exemplo de container de software livre, onde os Módulos Web podem ser publicados. Estes módulos Web são implementados através de servlets e JSP. Alógica de negócio através de EJBs, como detalhado a seguir:

- Camada Web: consiste de componentes *servlets* e páginas JSP com o objetivo de capturar requisições e processar respostas para a camada do cliente:
 - JSP (*Java Server Pages*)- é um componente que permite que código Java e certas ações pré-definidas sejam intercalados com conteúdo de marcação estático (*tags*), resultando na página sendo compilada e executada no servidor para ser entregue num documento HTML ou XML;
 - Servlet - é um componente do lado servidor que gera dados HTML e XML para a camada de apresentação de um aplicativo Web.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	8 / 29

- o Camada Negócios ou Enterprise Java Beans (EJB) : os componentes contém toda a lógica da aplicação e representam o modelo de negócio.

O JEE possui alguns frameworks MVC (*Model-View-Controles*) disponíveis que auxiliam ainda mais nesta tarefa, tornando a manutenibilidade mais fácil, melhor previsibilidade, maior produtividade de desenvolvimento e segurança. Alguns arcabouços para o MVC são o *Struts* e o *Java Server Faces (JSF)*. O JSF é um MVC mais moderno e com mais recursos de produtividade para interfaces. Existe a especificação do arcabouço *Java Seam* que permite melhor integração entre o EJB e o JSF.

O JEE possui um Serviço de Autenticação e Autorização. O JAAS (*Java Authentication and Authorization Service*) é uma API que permite às aplicações escritas na plataforma JEE usem serviços de controle de autenticação e autorização.

Em relação à persistência dos dados e objetos no SGDB, o JEE tem a especificação JPA (*Java Persistence API*). O Hibernate é uma implementação da JPA que faz o mapeamento dos objetos nos bancos de dados relacionais. Historicamente, o Hibernate surgiu antes da especificação JPA. A Sun se baseou no Hibernate para elaborar esta especificação.

2.4 Discussão entre Servidores JEE e PHP

Comparativamente, os servidores JEE, pelas suas características de maior robustez e escalabilidade, manutenibilidade, ciclo de vida de software mais longo, além de permitir aplicações mais complexas, seria o mais adequado para as características do Projeto do SREI, em detrimento do PHP. Assim na seção a seguir são avaliadas somente as opções de servidor de aplicação baseado na tecnologia JEE.

2.5 Principais servidores JEE

Os principais servidores JEE analisados são:

- Jboss;
- Tomcat;

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	9 / 29

- Glassfish;
- Jetty;
- Geronimo.

2.5.1 JBoss

O JBoss é um servidor de aplicação de código fonte aberto baseado na plataforma JEE implementada completamente na linguagem de programação Java. Como é baseada em Máquina Virtual Java, JBoss pode ser usado em qualquer Sistema Operacional que suporte Java. Os desenvolvedores responsáveis estão agora empregados por uma empresa de serviços chamada "JBoss Inc." fundada por Marc Fleury, o criador da primeira versão do JBoss. O projeto é custeado por uma rede mundial de colaboradores. Em Abril de 2006 foi anunciada sua aquisição pela Red Hat. A seguir são apresentadas as suas principais características:

- Um dos primeiros App, conhecido por estabilidade e segurança em produção;
- Arquitetura modular;
- Fácil configuração;
- Pioneiro em várias especificações do JEE (EJB3, SEAM, Web beans);
- Capacidade de *clustering* (aglomerado de computadores) e *failover* (recuperação do aglomerado no caso de falha de um dos nós de processamento);
- Maduro, robusto, escalável e confiável;
- Possui uma versão paga com suporte;
- Interface gráfica de configuração limitado na sua versão free. A versão paga possui uma interface melhor na sua versão paga;
- JBoss AS 6 não é totalmente aderente na especificação do JEE 6 (JEE compliant), que é a especificação mais recente.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	10 / 29

2.5.2 Tomcat

O Tomcat é um servidor web Java, mais especificamente, um container WEB (suporte para Servlets e JSP). O Tomcat possui algumas características próprias de um servidor de aplicação, porém não pode ser considerado um servidor de aplicação JEE completo, por não preencher todos os requisitos necessários. Por exemplo, o Tomcat não tem suporte a EJB. Desenvolvido pela Apache Software Foundation, é distribuído como software livre dentro do conceituado projeto Apache Jakarta, sendo oficialmente endossado pela Sun como a implementação de referência para as tecnologias Java Servlet e JavaServer Pages (JSP). Ele cobre parte da especificação J2EE, como tecnologias como servlet e JSP, e tecnologias de apoio relacionadas como Realms, JNDI Resources e JDBC DataSources.

Ele tem a capacidade de atuar também como servidor web, tendo nativamente integrado a um servidor web HTTP.

O servidor inclui ferramentas para configuração e gerenciamento, o que também pode ser feito editando-se manualmente arquivos de configuração formatados em XML.

A seguir são apresentadas as suas principais características do Tomcat:

- Servidor de aplicação (App) Leve;
- Utilizados internamente em vários outros App (JBOSS, Geronimo);
- Suporte a *clustering* e *failover*;
- Largamente usado em produção e em desenvolvimento;
- Bem testado e robusto;
- Não suporta EJB, JMS e coisas fora do lado web;
- Excelente para Servlets 3.0 e JSF 2 e Web Services.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	11 / 29

2.5.3 Glassfish

Em computação, GlassFish é um servidor de aplicação que foi desenvolvido pela Sun Microsystems para a plataforma Java Enterprise Edition (Java EE). Havia uma versão comercial denominada *Sun Java System Application Server*. Com a aquisição da SUN pela Oracle, o App está sendo distribuído pela própria Oracle, que atualmente continua a sua distribuição de forma free e gratuita.

GlassFish está atualmente na versão 2.1, sendo um dos poucos projetos de servidor de aplicação que possui suporte total às novas especificações J2EE 6.

Como características do Glassfish, podem ser citados:

- App relativamente novo (a partir de 2005);
- Suporte a *clustering e failover*;
- Aderente a especificação JEE 6 (Fully JEE 6 compliant);
- Fácil deploy (implantação de software);
- Ótima ferramenta administrativa;
- Robusto, estável, bem documentado e suportado;
- Recebe frequentes atualizações de segurança e performance.

2.5.4 Jetty

O Jetty é um servidor HTTP e *WEB Container* 100% escrito em Java. É o grande concorrente do Tomcat, que ficou famoso por ter sido utilizado como o servlet container do JBoss.

A grande vantagem do Jetty com relação ao Tomcat é a sua fácil configuração. O Jetty é instanciado junto com a aplicação em tempo de execução, o que é vantajoso considerando a necessidade de pessoal técnico de suporte de menor especialização. O problema desta característica é que no caso de necessidade de existências várias instâncias de aplicações, teremos várias instâncias do Jetty. Ele também foi o pioneiro a usar I/O (entrada/saída) assíncrono para aguentar uma carga maior de usuários simultâneos sem depender da antiga estratégia thread-per-connection,

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	12 / 29

assim suporte um maior número de conexões simultâneos. É o servidor Java muito difundido atualmente atualmente.

2.5.5 Geronimo

O Apache Geronimo é um servidor de aplicação criado pela Apache Software Foundation compatível com a especificação J2EE. Ele é distribuído como software livre sob a licença Apache. O Geronimo apresenta como características:

- Suportado por IBM;
- Bom desempenho;
- Estabilidade;
- Suporte a especificação anterior do JEE 5;
- Poucos cases de uso do Geronimo.

2.6 Comentários e Discussão dos Servidores de Aplicação

Segue um comparativo entre os principais Servidores JEE que foram apresentados nas seções anteriores.

Em relação ao suporte a especificação JEE, o Glassfish é totalmente aderente ao JEE6, enquanto o Jboss não é totalmente e o Geronimo é JEE5.

A desvantagens do Geronimo é não ter muitos "cases" de sucesso se comparado aos demais servidores JEE. Entretanto, o suporte da IBM tem agregado algumas das funcionalidades do WebSphere, servidor de aplicação comercial da IBM.

Para o projeto SREI, o Tomcat ou Jetty, que só possuem o Container WEB, seriam mais adequados para aplicações mais leves, enquanto os os servidores JEE completos, como o JBoss, Glassfish e Geronimo, seriam adequados para aplicações complexas, distribuídas, com maiores requisitos de controle e persistência.

Entre o Tomcat e o Jetty, devido à possibilidade de existirem várias instâncias de aplicação de Camada de Núcleo do SREI do Estado do Pará, cada um destas instâncias necessitaria de uma instância do Jetty para cada cartório, sendo o uso

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	13 / 29

deste App não adequado e vantajoso. Para tornar o SREI uma aplicação mais leve, a melhor escolha seria o Tomcat.

E no caso de ser mais adequado uso de servidor JEE, caso o SREI se torne uma aplicação complexa, sendo conservador, a recomendação é o uso do JBoss e Glassfish, por serem produtos mais disseminados que o Geronimo. O Glassfish tem a vantagem de ser aderente ao JEE 6.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	14 / 29

3 SGBD

Este levantamento se baseia nas duas principais soluções e mais utilizadas como Servidor Gerenciador de Banco de Dados (SGBD) em software livre: PostgreSQL e o MySQL.

3.1 PostgreSQL

PostgreSQL é um sistema gerenciador de banco de dados objeto relacional (SGBDOR) desenvolvido como projeto de código aberto mais avançado, contando com recursos como:

- Consultas complexas;
- Chaves estrangeiras;
- Integridade transacional;
- Controle de concorrência multi-versão;
- Suporte ao modelo híbrido objeto-relacional;
- Gatilhos;
- Visões (Consultas dinâmicas);
- Linguagem procedural em várias linguagens (PL/pgSQL, PL/Python, PL/Java, PL/Perl) para procedimentos armazenados;
- Indexação por texto;
- Possui uma extensão denominado PostGIS, que tem excelente de objetos e padrões de georreferenciamento;
- Há a EnterpriseDB, que é o principal fornecedor e suporte de produtos e serviços empresariais baseados em PostgreSQL.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	15 / 29

3.2 MySQL

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,. Suas características principais são:

- 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++, C#, Visual Basic, 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;
- Possui um suporte a objetos de dados e padrões de georreferenciamento, entretanto comparativamente muito menos que a extensão do PostGIS do PostgreSQL;
- Georreferenciamento: tem um suporte a Georreferenciamento, mas ainda inferior a PostgreSQL, através da sua extensão PostGIS.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	16 / 29

3.3 Considerações sobre o SGBD

Algumas considerações sobre os SGBDs:

- O PostgreSQL é mais escalável e aproveita melhor o número de núcleos de processamento (*cores*) do servidor;
- O PostgreSQL possui Async API rápido;
- O PostgreSQL destaca-se em grande concorrência e em grande quantidade de dados;
- O PostgreSQL destaca-se em grandes quantidades de operações de escrita;
- O PostgreSQL é reconhecido por ser robusto e ter um rigoroso controle de integridade de dados;
- O PostgreSQL suporta CHECK Constraint. O MySQL ainda não;
- Valores Default – Suporta qualquer função marcada como IMMUTABLE ou STABLE. No MySQL, somente a função NOW() é aceita em uma única coluna do tipo TIMESTAMP;
- O PostgreSQL suporta *stored function*, enquanto que o MySQL suporta *stored procedure*, que são funcionalidades equivalentes;
- PostgreSQL e MySQL possuem uma funcionalidade nativa de *built-in replication* (replicação e sincronização de dados entre máquinas). No PostgreSQL é recente, somente a partir da versão 9. No MySQL já é antiga;
- O PostgreSQL tem maior e melhor suporte a padrões de georreferenciamento em relação ao MySQL;

Os servidores de PostgreSQL e MySQL são muito semelhantes em características, devido aos fatores de robustez, integridade e suporte a georreferenciamento. A escolha pesa a favor do PostgreSQL.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	17 / 29

4 Sistema de *backup*

Nesta seção, são tratados as opções de ferramentas e software para o *backup* do Sistema SREI, que fisicamente poderão ser feitos em disco (local ou em *storage*).

4.1 Backup

Uma das opções para *backup* é a ferramenta de rsync, que é um software específico para realização de cópias, de sincronização de arquivos e diretórios (pastas), que junto com script de automação, funciona de maneira a acelerar as transferências ao copiar as diferenças entre dois conjuntos de arquivos, ao invés de copiar todo o conjunto de arquivos todas as vezes. Ele consegue identificar quais arquivos foram alterados e o administrador pode optar por copiar somente os arquivos novos ou modificados. A cópia de todos os arquivos tornaria o processo de *backup* demorado de maneira desnecessária. Esse recurso pode apresentar grande vantagem, principalmente para *backups* diários de grande quantidade de arquivos. Outro grande benefício do rsync é a preservação as informações sobre permissões e propriedade dos arquivos e diretórios, inclusive de links simbólicos.

Particularmente, o *backup* poder ser realizado nativamente, tanto no PostgreSQL ou MySql.

O PostgreSQL disponibiliza seguintes formas de *backup*:

- A ferramenta pg-dump para *backup* lógico em arquivo script.
- Para um agrupamento de banco de dados em PostgreSQL, o *backup* é realizado pelo pg-dumpal.
- No caso de um processo on-line (banco de dados em funcionamento), pode ser usada a ferramenta "pg_start_backup".

O MySql disponibiliza duas formas básicas de *backup*:

- *Backup* Lógico: As informações são representadas na forma de estrutura lógica das tabelas database de dados (CREATE TABLE) e o conteúdo é

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	18 / 29

representado da forma de INSERT. O *backup* é lento, pois o servidor deve acessar as tabelas e as informações e transformá-las na forma lógica. O resultado é maior comparativamente do que na forma física. A ferramenta *mysqldump* permite realizar este tipo de *backup*;

- **Backup Físico:** Consiste na cópia completa dos diretórios e arquivos, é um método mais rápido que o *backup* lógico e tem resultado mais compacto. As ferramentas *mysqlhotcopy*, *innobackup* e *ibbackup* permitem realizar este tipo de *backup*.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	19 / 29

5 Redundância e Alta Disponibilidade

A redundância e alta disponibilidade no banco de dados podem ser implementadas de várias formas, tais como balanceamento de carga, clusterização, replicação de dados síncrona e assíncrona, com Master/Slave, ou falha no servidor primário, um servidor secundário possa rapidamente assumir o trabalho (alta disponibilidade).

Servidores de banco de dados podem trabalhar juntos para permitir que caso haja alguma falha no servidor primário, um servidor secundário possa rapidamente assumir o trabalho (alta disponibilidade). Também é possível que vários servidores forneçam as mesmas informações (balanceamento de carga).

Este problema de sincronização é o grande obstáculo para que os servidores possam trabalhar em conjunto. Não há uma solução que resolva todos os problemas, mas há várias soluções que abordam este problema de forma diferente.

Algumas soluções lidam com o problema permitindo que somente um servidor modifique os dados. Servidor que pode modificar os dados é denominado servidor de leitura e escrita, master ou servidor primário. Servidor que recebe as modificações feitas no servidor primário é denominado de servidor *standby*, secundário ou escravo. Servidor secundário que não pode receber conexões até serem promovidos a servidor primário são chamados de servidores *warm* e o servidor secundário que pode receber conexões somente para leitura são chamados de servidores *hot*.

Algumas soluções são síncronas, ou seja, um dado alterado em alguma transação é considerado confirmado somente quando todos os servidores obtiveram sucesso na atualização desta informação. Em contraponto, algumas soluções são assíncronas, ou seja, há um certo período de tempo entre a alteração do dado e sua propagação para os outros servidores. O desempenho do servidor de banco de dados diminui conforme se aumenta o sincronismo entre os servidores, dessa forma, deve-se tomar um certo cuidado para a escolha da técnica de replicação a ser utilizada.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	20 / 29

A seguir, serão apresentadas diferentes soluções para a replicação de dados e alta disponibilidade para o PostgreSQL e o MySQL.

5.1 PostgreSQL

O PostgreSQL possui as seguintes técnicas de armazenamento e alta disponibilidade:

- Disco compartilhado: Esta técnica elimina a diminuição de performance da sincronização por ter somente uma única cópia do banco de dados. Utiliza-se somente um disco que é compartilhado entre vários servidores. Se o servidor primário falhar, o servidor secundário poderá acessar o disco e iniciar a base de dados, mesmo que seja necessário um processo de recuperação de informação. Isto permite uma rápida recuperação e não há perda de dados;
- Replicação do sistema de arquivos: Técnica em que todas as mudanças no sistema de arquivos do servidor primário será espelhada para o sistema de arquivo residente em outro computador. DRBD é uma solução popular, em Linux, de replicação de sistema de arquivos, como se fosse um RAID baseado em Rede;
- Replicação multi-master síncrona: PGCluster é um conjunto de modificações para o código fonte do PostgreSQL que permite a montagem de um sistema de replicação síncrono multi-master, garantindo replicação consistente em Cluster bem como balanceamento de carga para bancos de dados baseados em PostgreSQL. A replicação síncrona garante que dados sejam replicados sem que haja atraso e a característica de ser multi-master permite que dois ou mais nós de armazenamento possam receber requisições de usuários ao mesmo tempo;
- Replicação utilizando a técnica de *Point-In-Time Recovery* (PITR): Os servidores secundários são mantidos atualizados através da leitura dos registros do arquivo *write-ahead log* (WAL) do servidor primário. Se o servidor primário falhar, o servidor secundário conterá quase todos os dados e poderá rapidamente se tornar o servidor primário. Este é um processo assíncrono.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	21 / 29

Um servidor PITR pode ser implementado utilizando *file-based log shipping*, onde blocos de registros são enviados aos servidores secundários após um período de tempo, ou pode ser implementado utilizando streaming, onde a replicação é feita a cada novo registro inserido no arquivo WAL do servidor primário;

- Replicação baseada em trigger: O servidor primário envia, assincronamente, todas as mudanças dos dados para o servidor secundário. A ferramenta Slony-I é um exemplo desta replicação. Possui uma granulidade por tabela e suporta vários servidores secundários. Por causa da atualização ser feita assincronamente e em lotes, é possível que haja perda de dados;
- *Statement-based replication middleware*: Com esta técnica, um programa intercepta toda query (consulta) SQL e envia para todos os servidores. Cada servidor opera de forma independente. Operações de escritas são enviadas para todos os servidores, enquanto as operações de leitura são enviadas para somente um servidor.. As ferramentas PgPool-II e Sequoia são exemplos deste tipo de replicação;
- Replicação *multi-master* assíncrona - Cada servidor trabalha de forma independente, e periodicamente se comunicam entre si para a atualização e checagem de conflito. Os conflitos podem ser resolvidos pelo usuário ou utilizando a ferramenta Bucardo.

5.2 MySQL

A replicação de dados e alta disponibilidade em MySQL pode ser realizada basicamente pelas seguintes técnicas:

- *Statement-based replication*: Todas as query SQL são propagadas do Servidor Master para os servidores secundários;
- *Row-based replication*: O servidor primário escreve os eventos em arquivo de log indicando como cada tabela individualmente foi alterada, este arquivo é então propagado aos servidores secundários;

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	22 / 29

- **Clustering:** O MySQL possui recursos para implementar o *cluster* nativamente em várias máquinas servidoras, o componente que permite a criação de *clusters* é NDBCluster.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	23 / 29

6 Sistema operacional

Servidores UNIX, e mais recentemente servidores Linux e FreeBSD, possuem a grande preferência da área corporativa empresarial devido a grande robustez, segurança, desempenho e a conhecida resistência às altas taxas de carga.

Entretanto, existe uma restrição ao LINUX, quando se pretende executar plataformas criadas pela Microsoft. Neste caso, a melhor opção é a utilização de servidores baseados no Windows.

Como o ambiente de suporte à execução da aplicação SREI para o Estado do Pará será Java, a melhor alternativa é a utilização do sistema operacional LINUX.

A seguir estão apresentadas uma descrição das principais distribuições LINUX e da distribuição FreeBSD.

Linux com distribuição Ubuntu:

- Prós: Ciclo de lançamento fixo e período de suporte; novato-friendly, a riqueza da documentação, tanto oficial como usuário contribuiu. Possui uma versão com suporte;
- Contras: Falta de compatibilidade com Debian; freqüentes mudanças.

Linux com distribuição Fedora

- Prós: Altamente inovadoras, grande número de pacotes de software; observância da filosofia do Software Livre; dão preferência para funcionalidades de empresa, ao invés de usabilidade do desktop;
- Contras: Não possui suporte.

Linux com distribuição Red Hat:

- Prós: Inclui pacotes de software testados e aprovados na distribuição Fedora. Distribuição voltada a ambiente corporativo. Tem suporte pago;

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	24 / 29

- **Contras:** Distribuição mais conservativa, inclui funcionalidades somente após estarem testadas e comprovadamente estáveis o que decorre de possuir menos atualizações de pacotes que a distribuição Fedora.

Linux com distribuição Suse

- **Prós:** Possui uma ferramenta completa e intuitiva para configuração; grande repositório de pacotes de software, infra-estrutura web site excelente e documentação impressa. O Suse na versão Enterprise possui suporte;
- **Contras:** Possui ambiente gráfico e utilitários gráficos pesados, vistos como "inchados e lentos".

Linux com distribuição Debian

- **Prós:** Muito estável, controle de qualidade notável, inclui mais de 20.000 pacotes de software, suporte a mais arquiteturas de processadores do que qualquer outra distribuição Linux;
- **Contras:** Conservador, devido ao seu suporte para diversas arquiteturas de processador, novas tecnologias nem sempre são incluídas; ciclo de liberação lenta (uma versão estável a cada 1-3 anos); discussões nas listas de discussão para desenvolvedores e blogs podem ser incultos, por vezes. Não possui suporte.

Linux com distribuição Slackware

- **Prós:** Considerada altamente estável, limpo e em grande parte livre de bugs, forte aderência aos princípios do UNIX;
- **Contras:** Número limitado de aplicações suportadas; conservador em termos de base de pacote complexo. Não possui suporte. Mais indicado para desktop.

Linux com distribuição Gentoo

- **Prós:** Excelente software de gerenciamento de infraestrutura, personalização incomparável e opções de ajuste, a documentação online excelente;

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	25 / 29

- **Contras:** Às vezes a instabilidade e risco de avaria, o projeto padece de falta de rumos e brigas freqüentes entre os seus colaboradores. Não possui suporte.

Linux com distribuição CentOS:

- **Prós:** Extremamente bem testada, estável e confiável, livre para download e uso, vem com 5 anos de atualizações de segurança gratuitas; libera rápida e atualizações de segurança.
- **Contras:** Carece dos pacotes de software mais recentes. Por ocasião do lançamento da distribuição, a maioria dos pacotes de software estão desatualizados. Não possui suporte.

Sistema Operacional FreeBSD:

- Ao contrário das distribuições Linux, que são definidos como soluções integradas de software composto do kernel do Linux, e milhares de aplicações de software, o FreeBSD é um sistema operacional totalmente integrado construído a partir de um kernel que é descendente indireto da AT & T UNIX através da Berkeley Software Distribution (BSD);
- **Prós:** Rápido e estável, a disponibilidade de mais de 15.000 aplicativos de software (ou "portas") para a instalação, documentação muito boa;
- **Contras:** Tende a ficar para trás do Linux em termos de suporte para hardware diferentes, a limitada disponibilidade de aplicações comerciais, carece de ferramentas de configuração gráfica.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	26 / 29

7 Discussão

Este trabalho de levantamento e definição do ambiente de execução ainda não é conclusivo e deverá ser aprimorado ao longo do desenvolvimento do projeto SREI. Depende muito das informações e documentações geradas na modelagem do processo informatizado do SREI, da complexidade da arquitetura geral do sistema e de suas funcionalidades ainda pendentes de definição.

Apesar disso, pode-se apresentar algumas conclusões.

Como Servidor de Aplicação JEE, o TOMCAT é preferível se o SREI for um sistema de baixa complexidade ou Glassfish, caso o sistema SREI seja complexo.

Sobre o Banco de dados, a preferência é pelo PostgreSQL, pela sua robustez, integridade e também pela sua extensão PostGis, que suporta a objetos e padrões de georreferenciamento. Além disso, o PostgreSQL é usado em vários órgãos públicos com muito sucesso, como no SERPRO, Banco do Brasil, Metrô de SP e Datasus.

Em relação ao sistema operacional, a preferência é por uma distribuição LINUX com suporte técnico, como o RedHat ou Suse, pelas características do Projeto SREI de criticidade e manutenção.

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	27 / 29

8 Recomendação

Após análise de todas as opções levantadas para o ambiente de execução, recomenda-se as seguintes tecnologias:

- Linguagem de Programação: Java 6 ou mais recente.
- Servidor de Aplicação: Jboss AS 7.1 ou mais recente.
- Sistema Gerenciador de Banco de Dados: PostgreSQL 9.1.2 ou mais recente.
- Sistema Operacional: Red Hat Enterprise Linux 6 ou mais recente
- Sistema de Backup com Redundância

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	28 / 29

Referências

- <http://www.postgresql.org/docs/>
- <http://www.mysql.com/>
- <http://www.oracle.com/technetwork/java/javasee/tech/index.html>
- <http://glassfish.java.net/>
- <http://www.jboss.org/>
- <http://geronimo.apache.org/>
- <http://tomcat.apache.org/>
- <http://jetty.codehaus.org/jetty/>
- <http://www.linuxlinks.com/Distributions/>

Título	Versão	Classificação	Página
PROJETO SREI: PA2.4.3 - Relatório sobre definição do ambiente de execução	v1.0.r.18	LSI-TEC:Restrito	29 / 29