

Interoperabilidade de Sistemas: um estudo de caso do SiGPes – Sistema de Gestão de Pessoal da MB - GptFNSa

Fábio do Rosario Santos*
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, S/N
Barbalho, Salvador Bahia
fabiorosario@ifba.edu.br

Allan Edgard Silva Freitas†
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, S/N
Barbalho, Salvador Bahia
allan@ifba.edu.br

RESUMO

Este trabalho é um estudo de mecanismos de interoperabilidade entre sistemas existentes. Utiliza como estudo de caso o desenvolvimento de um sistema computacional denominado Sistema de Gestão de Pessoal (SiGPes). Sistema que permite integrar informações de sistemas diversos de uso administrativo da Marinha do Brasil (MB). O SiGPes é capaz de abstrair as dificuldades existentes na interação de dados dos diversos sistemas hoje utilizados no Grupamento de Fuzileiros Navais de Salvador (GptFNSa), bem como integrá-los, afim de torna-los mais conexos e gerenciáveis. Além disso, o SiGPes visa ser uma ferramenta de apoio à decisão, pois possibilita o uso de técnicas de Interoperabilidade de Dados para esse sistema, bem como a possibilidade da utilização das técnicas de *Business Intelligence* – BI, decorrentes da integração de dados. Os resultados do estudo de caso indicam a importância do SiGPes para a gestão do pessoal militar, destacando-se os artefatos de visualização de informações oferecidos pelo mesmo, através da plataforma *web*.

1. INTRODUÇÃO

O objetivo desse trabalho foi o estudo de mecanismos de interoperabilidade entre sistemas existentes. Foi utilizado como caso de estudo o desenvolvimento de um sistema computacional denominado de Sistema de Gestão de Pessoal (SiGPes). O sistema em questão, busca prover a integração dos dados e *softwares* relativos ao controle de pessoal, controle de acesso e lançamento dos índices obtidos no Teste de Avaliação Física dos militares do GptFNSa. E, além disso, esse sistema visa ser aproveitado, futuramente, para apoio a decisão nessa instituição com a utilização das técnicas de BI. O SiGPes atende as necessidades da instituição no que diz respeito à Gestão do Pessoal nos processos supracitados, visando tornar os dados gerados pelos sistemas envolvidos mais confiáveis e íntegros.

* Aluno do Curso de Análise e Desenvolvimento de Sistemas.

† Doutor em Ciência da Computação

O GptFNSa é uma Organização Militar – OM, pertencente a instituição Marinha do Brasil (MB). Tem por missão “Realizar ações de defesa de porto e/ou de área marítima restrita e das demais instalações navais e civis de interesse da MB, compatíveis com a sua composição, apoiar e/ou reforçar Grupamentos Operativos da Força de Fuzileiros da Esquadra em operações dentro ou fora da área de jurisdição do Segundo Distrito Naval; realizar operações de segurança interna e operações terrestres de caráter Naval a fim de contribuir para o preparo e a aplicação do Poder Naval na área sob jurisdição do Comando do 2º Distrito Naval que é o seu Comando imediatamente Superior” [8].

Nesse trabalho, serão abordados alguns conceitos, vantagens e desvantagens sobre *web services* e da linguagem XML (*eXtensive Markup Language*) para prover a interoperabilidade de aplicações desenvolvidas em diferentes plataformas de *hardware*, diferentes base de dados e diferentes linguagens de programação (que não é o caso dos sistemas envolvidos no estudo de caso) através da Internet ou de uma rede local.

Durante o desenvolvimento da ferramenta SiGPes foi utilizado a metodologia Processo Unificado (PU), que é fortemente ligada à orientação a objetos. Nesse sentido, serviu como Plataforma de desenvolvimento a Java EE (*Java Platform, Enterprise Edition*), que oferece um conjunto robusto de serviços e tecnologias para o desenvolvimento de aplicações *web*, a saber: JDBC (*Java Database Connectivity*), JavaMail, JSF (*JavaServer Faces 2.2*), dentre outras.

Este trabalho está organizado da seguinte maneira:

- A Seção 2 aborda o conceito de interoperabilidade de sistemas.
- A Seção 3 apresenta os conceitos fundamentais a respeito de *Web Services* e da Linguagem XML.
- Na quarta seção é apresentado o estudo de caso: controle de pessoal, controle de acesso e teste de avaliação física no GptFNSa.
- Na Seção 5 é apresentado o SiGPes, a solução e proposta para o problema exposto.
- Nas seções seguintes são expostas as conclusões, trabalhos futuros e referências do projeto.

2. CONCEITO DE INTEROPERABILIDADE SISTEMAS

No entendimento do governo brasileiro interoperabilidade não é apenas integrar sistemas nem tão pouco integrar redes. Não referencia unicamente troca de dados entre sistemas e não contempla simplesmente definição de tecnologia. É, sobretudo, a soma de todos esses fatores. Ademais, considera a existência de um legado de sistemas, de plataformas de *hardware* e *software* instalados. Parte de princípios que tratam da diversidade de componentes, com a utilização de produtos diversos de fornecedores distintos. A interoperabilidade tem por meta a consideração de todos os fatores para que os sistemas possam atuar cooperativamente, fixando as normas, as políticas e os padrões necessários para consecução desses objetivos. Esse entendimento foi fundamentado em quatro conceitos de diferentes Governos e Organizações, a saber: [20]

- I. “Intercâmbio coerente de informações e serviços entre sistemas. Deve possibilitar a substituição de qualquer componente ou produto usado nos pontos de interligação por outro de especificação similar, sem comprometimento das funcionalidades do sistema.” (Governo do Reino Unido).
- II. “Habilidade de transferir e utilizar informações de maneira uniforme e eficiente entre várias organizações e sistemas de informação.” (Governo da Austrália).
- III. “Habilidade de dois ou mais sistemas (computadores, meios de comunicação, redes, *software* e outros componentes de tecnologia da informação) de interagir e de intercambiar dados de acordo com um método definido, de forma a obter os resultados esperados.” (ISO).
- IV. “Interoperabilidade define se dois componentes de um sistema, desenvolvidos com ferramentas diferentes, de fornecedores diferentes, podem ou não atuar em conjunto.” (Lichun Wang, Instituto Europeu de Informática – CORBA Workshops).

O conceito de interoperabilidade, também, pode ser entendido como a capacidade de um sistema, seja ele informatizado ou não, de se comunicar de maneira mais transparente possível com outro sistema, semelhante ou não [19]. Em consequência dessa comunicação, os sistemas envolvidos podem compartilhar dados ou invocar processos comuns, independentes de sua plataforma, arquitetura, linguagem de programação ou sistema operacional. Para tanto, continuamente, as pessoas devem estar engajadas e com o propósito de assegurar que sistemas, processos e culturas de uma organização sejam gerenciados e direcionados para maximizar oportunidades de troca e reuso de informações, interna e externamente [20].

Para que se torne possível à integração entre sistemas heterogêneos e distribuídos, é preciso dispor da característica de interoperabilidade entre eles. Nesse sentido, os *web services* proporcionam às aplicações, integração de informações que se precisa atualmente, a fim de gerar aos sistemas todas as vantagens encontradas nesse tipo de arquitetura, tais como, maior escalabilidade, desempenho, interoperabilidade (foco deste estudo), segurança e rapidez em suas utilizações [1]. O entusiasmo em torno dos *web services* é baseada principalmente na promessa de interoperabilidade e por poderem ser desenvolvidos usando qualquer Linguagem de Programação que suporte conexão *web*. A arquitetura dos *web services* se baseia no envio de mensagens no padrão XML (*eXtensive Markup Language*) em um formato SOAP (*Simple Object Access Protocol*),

que serão abordados, arquitetura e padrões, mais adiante neste trabalho.

2.1. Dimensão de Integração

A integração entre dois sistemas diferentes de *software* pode se dar em diversos graus e modalidades, conforme a necessidade e certos aspectos de compatibilidade [23]. As modalidades de integração que interessam e são buscadas neste estudo são: a de intercâmbio de dados, que pode ser síncrona ou assíncrona; e a integração de procedimentos.

Intercambiar dados de maneira assíncrona já é uma atividade bastante corriqueira em alguns domínios de computação e isto geralmente se dá através de trocas de arquivos. Os formatos desses arquivos podem ser compreendidos por dois programas de computador diferentes. Por exemplo, processadores de texto mesmo distintos dão suporte a formatos de arquivos que programas distintos são capazes de editar. Entretanto, o intercâmbio de dados de maneira síncrona entre sistemas diferentes constitui um desafio maior. Não apenas dois programas de computador devem ser capazes de compreender um formato comum para representação de dados, mas precisam de um protocolo para realizar a troca destes dados dinamicamente, como por exemplo, padrões baseados em XML [23].

Apesar das vantagens do intercâmbio de dados, o foco deste trabalho é de uma integração de processos. Sendo assim, é desejável que certo módulo de *software* não apenas troque dados com outro módulo, mas utilize seus serviços como parte de sua tarefa, ou então trabalhe colaborando em tarefa resultantes da interação de dados de ambos. Isto significa que os dois ou mais módulos precisam trabalhar em conjunto, como se pertencessem ao mesmo programa.

3. CONCEITOS FUNDAMENTAIS DE WEB SERVICES

Os *web services* são uma consequência natural da evolução e utilização da Internet. Sobretudo, tais serviços observam aspectos de permissão a interações sistemáticas aplicação-a-aplicação, além da integração da infraestrutura de rede de computadores existente na *web*. Alguns consideram essa utilização massificada, como um processo que produz a evolução desse meio de comunicação entre pessoas [4]. E também, por a Internet ser uma grande rede de computadores, naturalmente, levou à possibilidade de se escrever aplicações e disponibilizá-las ao público em grande escala.

A *web*, inicialmente, baseava-se em páginas estáticas providas de informações interligadas, que poderiam ser consultadas através de navegadores *web*, conhecidos pela palavra em inglês *browser*. Só havia alterações no conteúdo dessas páginas quando a pessoa que matinha a página as realizava [1].

Em seguida, a *web* passou por uma evolução que tornou as páginas disponibilizadas capazes de interagir e acionar programas produtores de informações dinâmicas, provenientes de banco de dados e outras fontes. Então, foi possibilitada aos usuários desse conteúdo a capacidade de inserir, alterar e excluir informações. Portanto, o navegador *web* tornou-se o cliente universal da Internet [12] para ser usado por seres humanos, como um ambiente onde se constrói *interface* com o usuário. Consequentemente, e como abordaremos mais adiante neste trabalho, os *web services* permitem a interação entre programas de forma automática via plataforma *web*.

Em decorrência da inexistência de padrões, diversos parceiros de negócio passaram a interagir na Internet através de soluções proprietárias. Entretanto, essas soluções produzem dependências de seus fabricantes, sendo apenas licenciadas para aqueles que as utilizam, mas com direitos exclusivos para o produtor [25]¹. Portanto, desse modelo proprietário, criou-se uma demanda reprimida de negócios que poderiam ser realizados na *web*.

As soluções proprietárias permitem apenas algumas configurações, sem, no entanto, possibilitar inclusão de novas funcionalidades sem o consentimento da empresa proprietária. Essa demanda pode ser definida, de modo simples, como a necessidade apresentada por negócios comuns e empresas de todos os tamanhos em trocar informações entre sistemas diferentes que pudessem se comunicar através de padrões simples e públicos. Em resposta a uma demanda supracitada é que se propõe a tecnologia de *web services* e a Linguagem XML. Segundo [25]², XML pode ser considerada onipresente na Internet, simples, genérica, barata e de fácil uso.

A tecnologia *web services* pode ser apresentada sob duas visões capazes de refletir os pontos de vista técnico e conceitual. Levando-se em consideração o ponto de vista técnico dessa tecnologia, os *web services* constituem-se em *software* de baixo acoplamento, reutilizáveis, com componentes feitos para serem facilmente acessados pela Internet. Do ponto de vista conceitual, a utilização da tecnologia supracitada representa um modo para integrar tarefas que compõem um processo de negócio através da Internet, em uma cadeia de valor na qual procedimentos estão interligados e são interdependentes para atingir um resultado concreto final [25]³.

Basicamente, um *web service* é uma aplicação desenvolvida na plataforma *web* que dispõe e publica uma API (*Application Program Interface*) nessa plataforma. A API em questão possui a capacidade de suportar a comunicação programa-para-programa, o que permite às aplicações dessa tecnologia se comunicar usando XML através da própria *web* [12]. Os *web services* são simples, permitindo que serviços possam ser acessados pela Internet totalmente encapsulados, cabendo ao cliente saber apenas que serviço deseja e como empregá-lo. Além disso, essa tecnologia é caracterizada pelo baixo acoplamento entre as partes e fundamentada em XML, HTTP e outros padrões já estabelecidos na Internet, o que a torna extremamente atraente [1].

3.1. Modelo de Computação Distribuída Web

A complexidade e o crescimento da Internet fizeram com que vários modelos e propostas para a comunicação entre aplicações fossem apresentados como opção para computação distribuída. Segundo [5], o modelo de aplicação baseada na interação humana, navegador *web* e servidor, constitui-se no modelo de aplicação distribuída de maior adoção na Internet, superando modelos tradicionais de programação baseados em CORBA (*Common Object Request Broker Architecture*), DCOM (*Distributed Component Object Model*) e RPC (*Remote Procedure Call*), principalmente por ser mais flexível [5]. Além da flexibilidade, [1] relaciona outras características que fazem com que esse modelo seja amplamente adotado:

1. Interação simples entre os clientes (navegadores *web*) e servidores que trocam mensagem do tipo MIME (*Multipurpose Internet Mail Extensions*), sendo que a semântica da mensagem pode ser modificada através do uso de cabeçalhos;
2. Destino da mensagem é especificado indiretamente com o uso de URL (*Universal Resource Loader*), propiciando a implementação do balanceamento de carga e controle de sessão, entre outras características necessárias a aplicações distribuídas;
3. Simplicidade para o acréscimo de recursos e novos membros no uso da aplicação, tanto de cliente quanto de servidores, sendo necessário apenas um registro de DNS (*Domain Naming System*), o que é natural no contexto da Internet;
4. Padrões bem estabelecidos e formados, de domínio público e amplamente testados como HTTP (*Hypertext Transfer Protocol*) e HTML (*HyperText Markup Language*);
5. Clientes de baixa complexidade e disponíveis na grande maioria das plataformas computacionais: navegadores *web* (*browsers*);
6. Todas as ações ocorrem de forma descentralizada, sem a necessidade de uma coordenação central, o que propicia um elevado grau de interoperabilidade, estabilidade e flexibilidade de conexão para o uso de aplicações.

Diante do exposto, dois pontos principais precisam ser enfatizados: o baixo acoplamento entre clientes (navegadores *web*)/servidores, em que os clientes endereçam uma URL do servidor e esse quando disponível, oferece o serviço desejado; e a interação humana com a *interface* do sistema disponibilizada, visto que os sistemas no modelo *web* são desenvolvidos para que as pessoas interajam com os mesmos e obtenham o que desejam deles. Portanto, essas duas premissas em cooperação com o modelo de aplicação distribuída na Internet são responsáveis por sua grande aceitação no âmbito da computação distribuída.

Entretanto, as diferenças entre cliente e servidor nesse modelo são amplamente “empacotadas” e compatibilizadas sob três padrões de camadas básicas de *software*: TCP/IP (*Transmission Control Protocol/Internet Protocol*), protocolo de rede; HTTP, protocolo de apresentação; e HTML, linguagem de hipertexto, que é utilizada para descrever informações apresentadas no lado cliente.

A Figura 1 indica como esses padrões se estabeleceram. Do lado cliente, temos as 3 (três) camadas que servem para que a informação seja acessada. No lado do servidor, temos a informação produzida em HTML, depois transportada pelo HTTP e finalmente colocada na rede pelo TCP/IP, que a divide em pacotes e entrega para a rede. No outro lado o navegador *web* contém a mesma pilha de protocolos, recebendo a informação e apresentado-a em HTML [1].

É importante salientar, que o protocolo de apresentação, no caso da utilização de *web services*, é substituído de HTML para XML. O XML é uma linguagem capaz de descrever diversos tipos de dados e tem o propósito de facilitar o compartilhamento de informações.

3.2. Padrões adotados em Web Services

Uma definição técnica de *web services* poderia ser como um serviço disponibilizado na Internet, descrito via WSDL (*Web Service*

¹ Apud. [1].

² Apud. [1].

³ Apud. [1].

Definition Language), registrado via UDDI (*Universal Definition Discovery Interface*), acessado utilizando SOAP e com os dados transmitidos sendo representados em XML (*eXtensive Markup Language*) [22]. Para facilitar o uso e a aplicabilidade de *web services*, os padrões e tecnologias supracitados foram adotados por empresas de grande relevância da área de tecnologia como a IBM, Microsoft, Borland, Oracle, Sun, dentre outras, com a finalidade de planejar, desenvolver e decidir juntas a tecnologia de *web services*. Abaixo passamos a expor esses padrões, segundo [6].

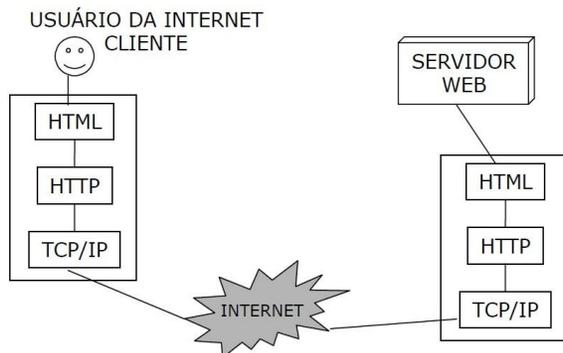


Figura 1. Ambiente padrão de Internet.

3.2.1. XML (*eXtensive Markup Language*)

A linguagem de marcação XML divide o documento em partes lógicas e hierárquicas, obedecendo a um conjunto de regras. É, amplamente, utilizada para trabalhar em conjunto com diversas tecnologias, seja no padrão de gravação de um arquivo ou mesmo no transporte de informações. Em parceria com a tecnologia de *web services*, fornece estrutura para o documento de regras e serviços (WSDL), pacote de dados e ocorrências.

3.2.2. WSDL (*Web Service Definition Language*)

O WSDL tem o objetivo de fornecer informações sobre a definição de um *web service*, utilizando-se do padrão XML para sua criação. Tal definição consiste em métodos, parâmetros e serviços fornecidos pelo serviço *web*.

3.2.3. SOAP (*Simple Object Access Protocol*)

O protocolo utilizado para troca de informações através de objetos criados em diversas linguagens de programação, como Delphi, Java, C++, C#, VB.NET, entre outras, é conhecido como SOAP. Utiliza-se da porta 80 de um servidor HTTP para o tráfego das informações, o que facilita o trabalho com servidores protegidos por Firewall, pois o tráfego é permitido mesmo a referida porta sendo fortemente monitorada.

3.2.4. UDDI (*Universal Definition Discovery Interface*)

O UDDI na realidade é um grande catálogo de *web services*, que pode ser comparado com os serviços de busca da Internet, com alguns parâmetros diferenciados, permitindo, assim, localizar e mapear serviços semelhantes. Diferentemente de SOAP e WSDL, que são capitaneadas e mantidas pelo W3C (*Worldwide Web Consortium*), a especificação UDDI é independente e mantida por um grupo de empresas do mundo dos negócios [14]⁴.

⁴ Apud. [1].

Como se pode notar, *web services* são utilizados empregando padrões de protocolos da Internet e XML. Portanto, formam uma arquitetura construída sobre uma infraestrutura padrão. Essa infraestrutura proporciona a construção de aplicações distribuídas como serviços que podem ser publicados, localizados e acessados pela Internet por qualquer cliente que se adeque a estes padrões já estabelecidos [15]⁵. Na Figura 2 é ilustrada a arquitetura de *web services* e os blocos do núcleo de construção, empregando conjuntamente os principais padrões.

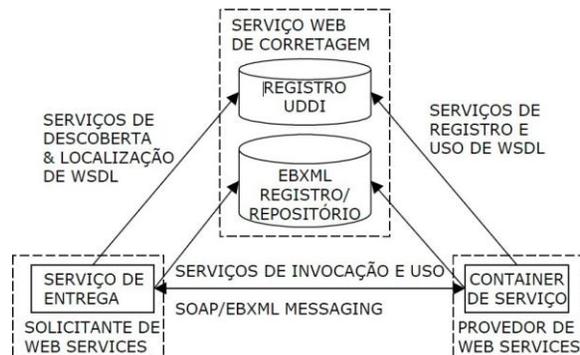


Figura 2. Arquitetura de Web Services e os blocos do núcleo de construção.

3.3. Arquitetura de Web Services

Uma aplicação que utiliza *web services* pode ser comparada como uma corrente, em que cada elo representa uma funcionalidade da aplicação. Um *link* entre elos diferentes representa processos compartilhados via *web services*, onde ocorre a conexão entre um serviço e um consumidor deste serviço, o que significa que um *web service* pode ser o consumidor de outro *web service*. Assim, o primeiro elo da corrente representa a *interface* da aplicação, por meio da qual requisições a *web services* poderão ser realizadas. Essa representação não implica em forte acoplamento entre os serviços, pois eles podem ser utilizados por aplicações diversas [17]⁶.

A arquitetura de *web services* tem como base a interação de três entidades: provedor do serviço (*service provider*), cliente do serviço (*service requestor*) e servidor de registro (*service registry*). Geralmente, as interações tem a finalidade de publicar, buscar e executar operações. As operações, os componentes envolvidos e suas interações são ilustrados na Figura 3.

O provedor do serviço (*service provider*) representa a plataforma que hospeda o *web service*, permitindo que os clientes acessem o serviço [22]. O cliente do serviço (*service requestor*) é a aplicação que está procurando, invocando ou iniciando uma interação com o *web service*. O cliente do serviço pode ser uma pessoa acessando através de um *browser* ou uma aplicação realizando uma invocação aos métodos descritos na *interface* do *web service*. O *service registry* representa os servidores de registro e busca de *web services* baseados em arquivos de descrição de serviços que foram publicados pelos *service providers*. Os clientes (*service requestor*) buscam por serviços nos servidores de registro e recuperam informações referentes à *interface* de comunicação para os *web services* durante a fase de desenvolvimento ou durante a execução do cliente, denominados *static binding* e *dynamic binding*, respectivamente [22].

⁵ Apud. [1].

⁶ Apud. [24].

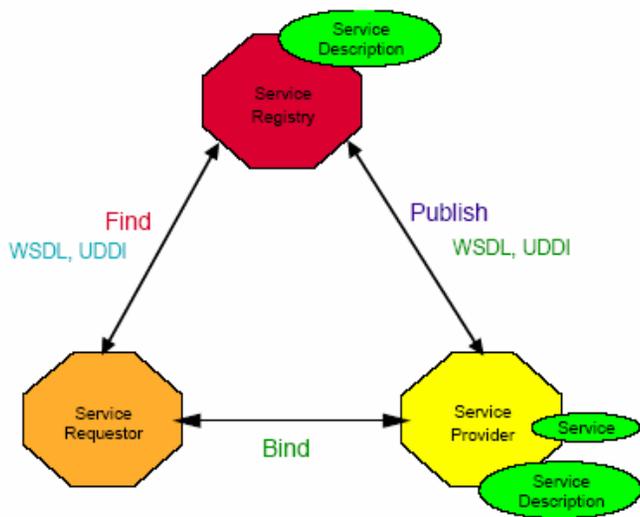


Figura 3. Arquitetura de web services, operações e interação das entidades.

3.4. Linguagem XML

XML é um formato de texto simples, muito flexível derivado do SGML (*Standar Generalized Markup Language*) – ISO 8879. Originalmente concebido para enfrentar os desafios da publicação eletrônica em grande escala, o XML também está desempenhando um papel cada vez mais importante na troca de uma ampla variedade de dados, principalmente, na web [27]. Entretanto, no contexto dos web services, a XML, não é apenas utilizada como um formato para troca de mensagens, mas também a forma através da qual os serviços são definidos [24]⁷. Com a utilização de XML, qualquer número de elementos que associam significado as informações podem ser definidos [23]. A composição da linguagem XML é a que se segue [9]:

- Declaração: o uso da declaração da tag é importante, pois mostra que se trata de um documento XML e a versão em que foi escrita, podendo também declarar outros atributos que podem ser importantes na leitura do documento;
- Elementos: elementos iniciam-se com '<' e terminam com '>'. Outra característica da linguagem XML é que, assim como os sistemas da família Unix, o XML diferencia letras maiúsculas de minúsculas;
- Comentários: podem conter qualquer dado ou informação, sendo que o uso de comentários auxilia no entendimento do documento por usuários humanos, principalmente para sua transmissão. O comentário inicia com '<!--' e termina com '->';
- Hierarquia: é baseada na semântica ou na estrutura lógica de documentos. A Figura 4 mostra a representação gráfica e seu correspondente em XML de uma hierarquia. Toda hierarquia possui uma raiz, que é o início da hierarquia e geralmente uma abstração;

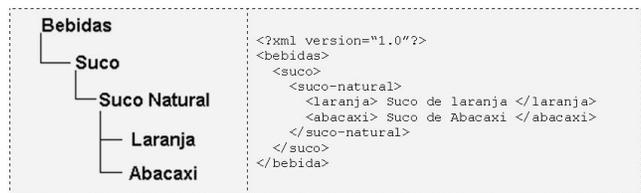


Figura 4. Exemplo de uma hierarquia em XML.

- Atributos: tags XML podem conter atributos, porém diferentes de elementos não possuem sub-atributos ou outros elementos. Na declaração de atributos devem ser dados seu nome e valor, podendo ter mais de um atributo a uma única tag, como pode ser visto na Figura 5:

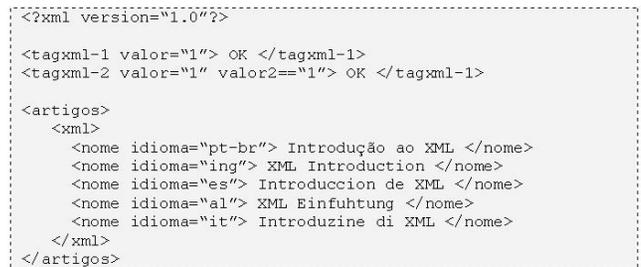


Figura 5. Atributo em um documento XML.

- Elementos vazios: na linguagem XML os elementos vazios possuem uma sintaxe modificada. Os elementos vazios são utilizados para marcar uma determinada ação, onde são representados delineados por '<' no início e '>' no final, como no exemplo '<vazio/>'.

Em sua grande maioria, os princípios em se baseia a XML são resultados da experiência anterior de grande sucesso, o HTML. O HTML se mostrou adequado para diferentes plataformas e sistemas que compõem a Internet. Entretanto, XML amplia os horizontes, lançando-se como linguagem base não apenas para a descrição de páginas web, mas para qualquer conteúdo registrado e trocado entre aplicações [23]. XML possui recursos de metalinguagem, que permitem a criação de linguagens para domínios específicos de aplicação. No caso de web services, a sintaxe de XML utilizada, especifica como os dados são genericamente representados, define como e com que qualidade de serviços os dados são transmitidos, e detalhes de como os serviços são publicados e descobertos. Implementações de web services decodificam os conteúdos das mensagens em XML para interagir com várias aplicações e domínios de softwares que usam estes serviços [3]⁸.

Empregando Unicode para codificar seus caracteres, XML é projetada de modo a conter sua auto-descrição de dados representados neutra. Portanto, pode armazenar como um simples documento de dados altamente complexos, de modo legível e processável. Com essa flexibilidade, XML tornou-se o padrão para o transporte de dados estruturados, conteúdos e formato para dados que representam documentos em meio eletrônico.

De acordo com [15]⁹, no núcleo da tecnologia de web services, XML desempenha o papel vital de ser facilmente transportável, compatível e comum com a camada de rede pela qual é transmitida

⁷ Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2005. “Orientação: Prof. Dr. Sérgio Creso Coelho da Silva Pinto, Ciências Exatas e Tecnologias”.

⁸ Apud. [13].

⁹ Apud. [1].

em todos os formatos de comunicação. A Figura 6 demonstra graficamente como XML posiciona-se na base dos três principais padrões adotados para constituir a tecnologia de *web services* (SOAP, WSDL e UDDI) descritos na subseção 3.2 e baseados sobre a linguagem XML.

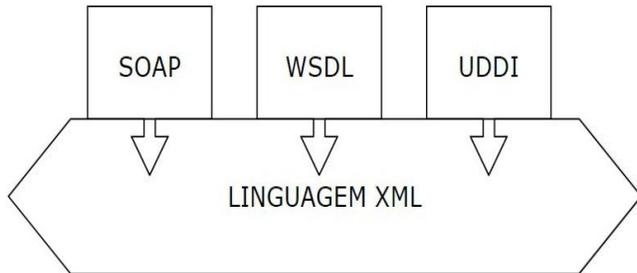


Figura 6. Linguagem XML como base de *Web Services*.

3.5. Vantagens de Web Services

A principal vantagem de um *web service* está na comunicação e na distribuição de informação de forma dinâmica. As tecnologias nela incluídas (HTTP, SOAP, WSDL, UDDI e XML) são amplamente difundidas e trazem total independência das linguagens de sistemas, permitindo uma evolução mais suave e econômica dos modelos computacionais [11]. O que se deseja com *web services* é encaixar sua arquitetura em qualquer implementação onde a aplicação de qualquer outra tecnologia seja cara ou quase impossível de ser utilizada. Entretanto, existem outras vantagens para o uso de *web services*: 1) integrar sistemas legados; 2) rápido desenvolvimento e diminuição dos custos; e 3) integrar-se com parceiros comerciais externos [21]¹⁰.

3.5.1. Integração de Sistemas Legados

A interconexão entre sistemas legados (estabelecidos, seguros, operacionais) foi onde se concentrou os esforços dos primeiros adeptos da tecnologia de *web services*. [21]¹¹ Do ponto de vista gerencial, um sistema de *software* mais antigo que vem executando seu processamento perfeitamente, é mais confiável que qualquer sistema novo, por mais ultrapassada que seja a tecnologia utilizada no primeiro. Os clientes preferem produtos que funcionem corretamente e de maneira estável por muito tempo, sendo a tecnologia utilizada para esse fim irrelevante. Por conseguinte, a substituição desse produto por outro que não tenha as mesmas garantias, mas que seja mais moderno é de difícil aceitação. Nesse sentido, *web services*, a fim de conservar toda a equidade que determinado cliente construiu na base de código do produto adquirido, são escritos de uma maneira que exige pouca ou nenhuma mudança na base do código legado.

3.5.2. Rápido desenvolvimento e diminuição dos custos

Um dos principais focos do desenvolvimento de *software* é o conceito da reutilização de código, uma ideia simples e intuitiva. Entretanto, essa reutilização apresenta um grande problema no que se refere ao empacotamento do código: para que esse seja reutilizado, precisa ser criado de maneira a facilitar que um novo programa o encontre e faça uso dele.

A fim de solucionar o problema em questão, *web services* se baseiam na criação de um serviço e de uma descrição formal do mesmo. O documento WSDL contém informações suficientes sobre o serviço para permitir que um cliente seja gerado. Em vez de iniciar do zero, o programador de um cliente de *web service* inicia com um cliente gerado que possa acessá-lo corretamente. Podendo, então, aperfeiçoá-lo para interagir com os outros sistemas no lado cliente. A utilização desse método tornam significativos aspectos como a economia de tempo e a eficiência do aprendizado.

3.5.3. Melhor integração com parceiros comerciais externos

Atualmente, a maioria das empresas é montada sobre uma base de parceiros comerciais que outrora eram classificados como simples fornecedores. Contudo, hoje, se auxiliam na redução de custos, a fim de obterem resultados satisfatórios para todos os envolvidos. Nesse sentido, os *web services* são ideais para esse tipo de interação, sendo possível interconectar sistemas diferentes de diversas empresas sem fazer alterações significativas em qualquer um deles. Permitindo, no entanto, a troca e coleta de dados relevantes para os parceiros comerciais.

3.6. Desvantagens de Web Services

3.6.1. Riscos

Os *web services* apesar de resolverem muito bem certos problemas, carregam os seus próprios, decorrentes das suas especificações e inerentes às tecnologias em que se baseiam. O conhecimento desses problemas é importante para que se possa planejar e desenvolver tendo-os como base. [21]¹² aponta alguns desses problemas:

- I) Disponibilidade - Por usarem a mesma infraestrutura dos sítios *web*, os *web services* não estarão totalmente disponíveis. Caso haja a necessidade de um sistema com 100% de disponibilidade, deve-se, normalmente, criar mecanismos que tentem repetir a transação ou terminá-las elegantemente quando isso não ocorre.
- II) Adequação às necessidades - Sempre existirão necessidades especializadas. Alguns consumidores podem precisar do único pequeno recurso que ninguém mais precisa. Os *web services* são vistos como a tecnologia em que uma única solução serve para muitos clientes.
- III) Interfaces imutáveis - No investimento de criação de um *web service* para os clientes, deve ser evitada a mudança de qualquer um dos métodos que é fornecido e os parâmetros que os clientes esperam. Podem-se criar novos métodos e acrescentá-los ao serviço, mas, se mudar os métodos existentes, os programas dos clientes falharão. Isso é fácil de fazer até que se descubra que um dos seus métodos existentes está retornando respostas erradas e não pode ser reparado porque a metodologia é fundamentalmente falha. Embora esse tipo de problema ocorra em todos os sistemas, isso é especialmente verdade nos *web services*. Pode-se não saber quem está usando o serviço e, conseqüentemente, não existe uma maneira de informar esses usuários da mudança.
- IV) Garantia de execução - Toda a ideia por trás de ter um programa de computador em vez de fazer o trabalho manual-

¹⁰ Apud. [19]

¹¹ Apud. [19]

¹² Apud. [19]

mente é que o programa possa rodar desassistido. O HTTP não é um protocolo seguro, uma vez que não garante entrega ou resposta. Se for preciso esse tipo de garantia, é necessário escrever um código para tentar novas requisições ou enviar as requisições através de um intermediário que realize novas tentativas.

3.6.2. Problemas de Desempenho

Os *web services* se baseiam no HTTP, que foi projetado para permitir que um servidor manipule centenas de requisições rapidamente [21]¹³. Para tanto, ele não mantém uma conexão longa com informações de estado entre os clientes e o servidor. Em vez disso, inicia uma conexão nova com o servidor e a mantém apenas pelo tempo em que precisar transferir os dados. Ao final da transferência, a conexão é terminada e o servidor é liberado para processar uma requisição de outra aplicação. Isso torna *web services* muito transacionais por natureza. Embora a transação de comunicação HTTP permita que o lado servidor manipule muitos clientes, significa também, muito tempo desperdiçado criando e terminando conexões, para clientes que precisam realizar um grande número de chamadas entre o cliente e o servidor [19].

Outro aspecto de desempenho que precisa ser considerado com os *web services* é a conversão de mensagens XML durante o processo de comunicação. Sendo que toda a comunicação acontece como mensagens XML codificadas em um formato especial (um envelope SOAP), a conversão supracitada leva tempo. A depender da complexidade e quantidade dos dados pode se tornar uma séria penalidade. Ainda que esse tipo de penalidade de tempo ocorra com outras arquiteturas, ela pode ser especialmente ruim para sistemas baseados em *web services*, porque os dados são transferidos como texto XML. Esse *overhead* não é tão grave quando se considera uma *string* ou um número simples a ser codificado. Entretanto, os dados binários como imagens, costumam ser muito maiores e levam muito mais tempo para serem convertidos [19].

4. ESTUDO DE CASO: CONTROLE DE PESSOAL, CONTROLE DE ACESSO E TESTE DE AVALIAÇÃO FÍSICA

O Grupamento de Fuzileiros Navais de Salvador é uma Organização Militar (OM) da Marinha do Brasil, situada na Avenida da França, 1300, Comércio, Salvador-BA. Apesar de não ter como atividade fim a Tecnologia da Informação, possui um Centro de Processamento de Dados (CPD) que é responsável por todo assunto relacionado à informática da instituição.

O CPD é responsável pelo suporte e manutenção de computadores, rede e nós últimos meses, por conta de ineficiências dos sistemas de informação utilizados, também no desenvolvimento de sistemas que atendam as necessidades específicas do Grupamento. Contudo, sem alterar ou substituir por completo os *softwares* existentes, mas aprimorá-los com ferramentas e métodos mais modernos de desenvolvimento de sistemas de informação.

Os sistemas que foram utilizados no estudo de caso deste artigo são: o Sistema de Cadastro de Teste de Avaliação Física, o Sistema de Controle de Pessoal e o Sistema de Controle de Acesso.

O primeiro sistema que faz parte do estudo de caso é Sistema de Cadastro de Teste Avaliação Física, desenvolvido com a ferramen-

ta *Microsoft Office Access*. Nele são lançados todos os índices obtidos por cada militar, nas diversas modalidades que o compõem: abdominal, corrida, flexão de barra ou solo, natação e permanência dentro d'água.

Cada modalidade do Teste de Avaliação Física (TAF) possui pontuação que varia de 0 a 100 pontos, com exceção da permanência que varia de 0 a 50 pontos, perfazendo um total máximo de 450 pontos. O TAF é realizado anualmente, sendo de responsabilidade de pessoal designado aplicar os testes. Ao final de cada modalidade, o militar que foi avaliado rubrica em relatório próprio, a fim de confirmar os índices obtidos. Esse sistema possui uma relação de militares que é atualizada pelos militares da Seção de Operações (S3), mas especificamente os relacionados aos Esportes, tomando como base a relação geral de militares disponibilizada pela Seção de Pessoal (S1).

O Sistema de Controle de Acesso é o segundo sistema pertencente ao estudo de caso. Utiliza o *software* BioConfigurador Acesso – Configurador para Equipamentos de Acesso criado pela Biometrus S.A. para administração dos equipamentos de acesso fabricados por ela [2]. Este sistema é administrado pela Seção de Inteligência (S2), que é responsável pelo cadastro das digitais dos militares do GptFNSa no *software* de controle de acesso, vinculando-as à sua matrícula. Assim como o sistema do TAF, para atualização dos dados dos militares, se baseia na relação geral de militares disponibilizada pelo S1.

O Sistema de Controle de Acesso propicia a geração de relatórios diversos sobre o acesso do pessoal: gerais ou específico para algum militar. Bem como, a alteração e exclusão de dados dos militares cadastrados, além de permitir que todas as informações dos equipamentos de acesso sejam persistidas em uma base de dados do *Microsoft Office Access*.

Por fim, o terceiro sistema contido no estudo de caso é o Sistema de Controle de Pessoal. É neste sistema que a relação geral de militares utilizados pelos outros sistemas participantes do estudo de caso é atualizada. O Encarregado de Movimentação de Pessoal é o responsável por manter a relação geral de militares atualizadas e disponível.

A Seção de Pessoal na apresentação de militares efetua o cadastro do pessoal, com todos dados pessoais e inerentes à carreira do militar, tais como última data de promoção de graduação e antiguidade dentro da respectiva graduação para praças e posto para oficiais, além da situação funcional de cada militar. Após a atualização da relação, ela é disponibiliza através da página *web* do Grupamento, para que todos possam acessá-la e dela possam fazer seu uso específico.

No cenário supracitado dos sistemas que integram o estudo de caso, deve ser destacada a diminuta interligação entre as relações de militares do sistema de controle de pessoal e o do sistema de teste de avaliação física, bem como a inexistência no sistema de controle de acesso. Além disso, as relações são atualizadas manual e separadamente. Consequentemente, o controle de presença efetiva do pessoal é muito demorado, e os sistemas envolvidos estão geralmente desatualizados.

Outro ponto importante é o que torna o processo de cadastro dos índices obtidos no TAF demorado: a centralização da responsabilidade de fazer os lançamentos no sistema em apenas um militar ou estação de trabalho. Partindo dos problemas expostos, o SIGPes é a ferramenta proposta para resolvê-los com mecanismos de interoperabilidade entre os sistemas descritos nessa seção.

¹³ Apud. [19]

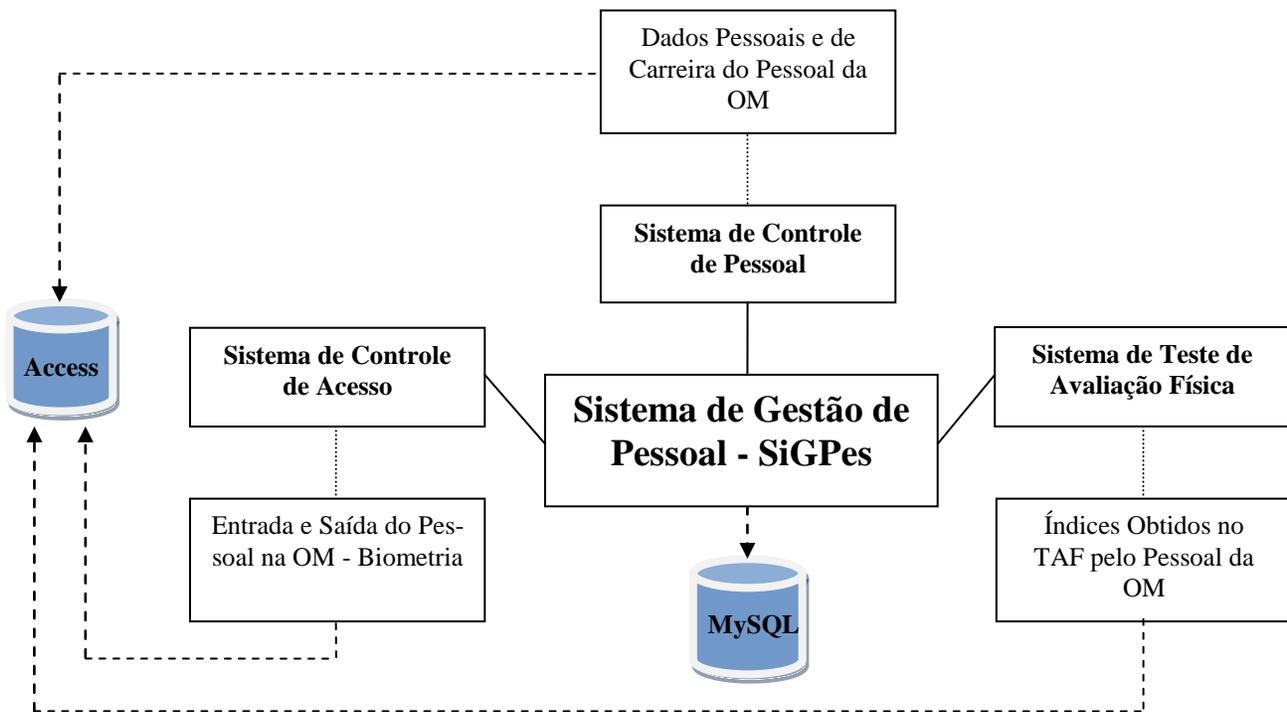


Figura 7. Sistemas e dos dados necessários para integração.

Diante da situação exposta, o SiGPes tem o objetivo de fazer a interação entre os diversos sistemas utilizados no GptFNSa. Como está demonstrado na Figura 7, o objetivo da solução proposta é a coleta dos dados gerados pelos sistemas hoje em uso. Esses dados são persistidos no *Microsoft Office Access*, mas poderiam ser em outro Banco de Dados.

Por conseguinte, esses dados serão processados e persistidos num SGBD (Sistema Gerenciador de Banco de Dados), no caso em questão, o MySQL. O intuito dessas ações é permitir, no futuro, a tomada de decisão do Comando do GptFNSa mais pautada em informações precisas. Por exemplo, na escolha da melhor maneira de preparar os militares da OM para o teste de avaliação física.

A fim tornar o SiGPes uma ferramenta de apoio a decisão, pode-se utilizar de técnicas modernas, como as técnicas de BI (*Business Intelligence*). BI basicamente refere-se ao processo de coleta, organização, análise, compartilhamento e monitoramento de informações que oferecem suporte a gestão de negócio.

Além dos objetivos supracitados, o SiGPes possibilitará que os sistemas hoje em uso, tenham algumas de suas funcionalidades melhoradas e disponibilizadas através da plataforma *web*. Principalmente, as relacionadas ao cadastro e a consulta dos dados persistidos nos seus respectivos banco de dados. As funcionalidades do SiGPes serão disponibilizadas de forma independente, com autenticação e autorização de usuários com suas respectivas permissões. Fato este, que atualmente não é oferecido pelos sistemas desse estudo de caso em uso na Instituição.

O Grupamento de Fuzileiros Navais de Salvador dispõe para o funcionamento do aplicativo de um servidor de aplicações que serve tanto para o desenvolvimento como para o funcionamento da aplicação. Contudo, o servidor *web* da instituição é independente e é através dele que os diversos setores têm acesso às aplicações.

5. A FERRAMENTA SIGPES

O SiGPes foi desenvolvido levando-se em consideração a importância das atividades desempenhadas pela Seção de Pessoal, Seção de Inteligência e Seção de Operações, com relação ao controle de pessoal, controle de acesso e lançamento dos índices obtidos no teste de avaliação física, respectivamente. Além do fato, de que essas atividades, quando realizadas de maneira ineficiente, impactam negativamente na gestão dos aspectos pessoais e de carreira dos militares que servem nesta OM, bem como torna dispendioso e lento o cumprimento dos prazos estipulados pelo alto escalão da Marinha do Brasil.

Diante do exposto, foi identificada a necessidade da utilização de um sistema de gerenciamento mas coeso que permita a interoperabilidade dos sistemas legados atualmente em funcionamento no GptFNSa. Com o intuito de prover-lhes de funcionalidades que por si só não são capazes de disponibilizar. Esta é a proposta do Sistema de Gestão de Pessoal (SiGPes), que consiste em uma importante ferramenta para auxiliar o gerenciamento e controle dos aspectos pessoais e de carreira do pessoal militar da MB.

O SiGPes está dividido em módulos, correspondentes a cada sistema legado existentes: controle de pessoal, controle de acesso e teste de avaliação física. Sendo possível adicionar-lhe outros módulos e sistemas. O SiGPes além de disponibilizar através da plataforma *web* funcionalidades inerentes as atividades atualmente exercidas pelas Seções supracitadas, proporciona uma mudança gradativa nos sistemas envolvidos. Essa abordagem permite a utilização das novas funcionalidades em conjunto com aquelas disponibilizadas pelos sistemas legados.

As funcionalidades de cada sistema envolvido e que essa ferramenta dá suporte, são disponibilizadas ao usuário através de um *login* e senha de acesso, com suas devidas permissões conferidas

pelo Administrador da ferramenta. A partir do acesso autorizado e autenticado o usuário poderá:

1. No caso do módulo de controle de pessoal incluir, alterar ou excluir dados dos militares da OM, bem como ter acesso a consultas, relatório e gráficos;
2. No módulo de controle de acesso, além das operações e inclusão, alteração e exclusão de dados, é possível consultar, por exemplo, uma relação de militares que foram apresentados para servir mais ainda não consta no controle de acesso, sem a necessidade de uma participação formal da Seção de Pessoal;
3. Já no módulo do TAF, além das funcionalidades básicas de inclusão, alteração e exclusão dos índices, também será possível ter acesso a relatórios estatísticos e gráficos dos percentuais alcançados em cada modalidade pelos militares.

A Figura 8 refere-se a tela de inclusão dos índices obtidos pelos militares no Teste de Avaliação Física em todas as modalidades: corrida, caminhada, flexão na barra, flexão no solo, abdominal, natação e permanência dentro d'água. Apesar de ser apenas uma pequena parte de todo o SiGPes, essa tela demonstra já alguns vantagens da interoperação entre os sistemas envolvidos. O usuário dessa funcionalidade se preocupará apenas em incluir os índices obtidos pelos militares da sua companhia. A responsabilidade de incluir os dados pessoais de cada militar é dos usuários do módulo de controle de pessoal.

O diferencial oferecido pelo SiGPes consiste no uso dos mecanismos de interoperabilidade entre sistemas para coleta de dados nas diversas bases dos sistemas legados, a fim de gerar informação consistente persistidas num SGBD. Além, de propiciar a reestruturação suave dos sistemas envolvidos, possibilitando a conciliação dos sistemas legados com a ferramenta desenvolvida.

5.1. Requisitos Funcionais do SiGPes

São definidos como os principais requisitos funcionais necessários para que seja possível o atendimento de todos os objetivos propostos pela ferramenta SiGPes:

1. O sistema deverá coletar os dados persistidos nas bases de dados dos sistemas legados.
2. O sistema deverá persistir os dados coletados no MySQL.
3. Permitir a inclusão/alteração de índices do TAF.
4. Permitir a inclusão/alteração/exclusão dos dados de pessoal.
5. O sistema deve permitir a inclusão/alteração/exclusão de usuários.
6. O sistema deve permitir a atribuição de papéis para os usuários.
7. O sistema deve possibilitar a obtenção de informações gráficas dos dados.

5.2. Requisitos Não Funcionais do SiGPes

Caracterizam-se como requisitos não funcionais necessários para o bom funcionamento do SiGPes, os itens abaixo:

1. O Sistema deverá ser desenvolvido com tecnologia *Java*, pois é a linguagem de programação homologada pela Marinha do Brasil (MB), e as camadas de apresentação e lógica de aplicação conectadas através de *web services*.
2. O SiGPes deve se basear no padrão arquitetura MVC (*Model-View-Controller*), bem como utilizar o estilo arquitetural em camadas cliente/servidor.
3. Deverá ser utilizado o Sistema Gerenciador de Banco de

The screenshot shows a web browser window with the URL `http://localhost:8085/SiGPes/faces/taf/digitador/cadastroTAF.xhtml`. The page header is 'SISTEMA DE GESTÃO DE PESSOAL'. The main content area is titled 'Lançamento dos Índices do TAF' and contains a table with the following data:

P/G	Espc	Nip	NomeGuerra	Sexo	Id	Cor	Cam	Abd	Bar	Flex	Nat	Perm	Cia	Pontos
CMG	(FN)	77777777	SALGADO	M	45	1700	0	45	8	0	140	Apto	CiaCSv	e
CMG	(FN)	88888888	CORDEIRO	M	39	1600	0	54	10	0	151	Apto	CiaCSv	e
CMG	(FN)	66666666	CRUZ	M	34	1800	0	46	11	0	125	Apto	CiaCSv	e
CMG	(FN)	99999999	MASCARENHAS	M	38	1900	0	35	7	0	230	Apto	CiaCSv	e

Below this table is another section titled 'Militares que não obtiveram índices na modalidade Abdominal' with a table containing the following data:

P/G	Espc	Nip	NomeGuerra	Abdominal
CMG	(FN)	44444444	PEREIRA	0
CF	(FN)	55555555	TEIXEIRA	0
CF	(FN)	33333333	NETO	0
CF	(FN)	11111111	FILHO	0

At the bottom of the page, there are navigation buttons for 'Abdominal', 'Barra', 'Corrida', 'Natação', and 'Idade'.

Figura 8. Cadastro dos índices obtidos no TAF.

Dados Relacional (SGBD) MySQL, homologado pela Marinha do Brasil, para as persistências necessárias do Sistema.

- O Sistema deve utilizar o TCP/IP como protocolo básico de comunicação entre as diversas camadas.

5.3. Desenvolvimento do SiGPes

O Processo Unificado (PU) foi a metodologia de processo de desenvolvimento empregada para desenvolver o SiGPes. Essa metodologia é fortemente ligada à orientação a objetos e utiliza princípios modernos na área de engenharia de *software* como a componentização e revisões. O PU define quatro fases de desenvolvimento, a saber:

- **Concepção:** fase de levantamento genérico do escopo do projeto, sem a pretensão de especificar detalhadamente requisitos. O objetivo dessa fase é ter uma visão inicial do projeto, a fim de determinar sua viabilidade. Nessa fase foi analisado a viabilidade do SiGPes e realizadas entrevistas com os usuários para identificar necessidades e diferentes perfis de acesso.
- **Elaboração:** nessa fase há um detalhamento mais aprofundado do produto, com o intuito de permitir um melhor planejamento da fase de construção. Foram analisados os requisitos e o domínio da aplicação.
- **Construção:** fase na qual é implementada, de modo interativo, uma versão completamente operacional do produto (implementação do código). É, também, nessa fase que se prepara a aplicação para implantação (realização de testes).

- **Transição:** nesta fase o produto é colocado à disposição dos usuários (implantação), após a realização de testes finais.

A metodologia em questão foi escolhida para elaboração deste trabalho, principalmente, por se tratar de um processo ágil, com uma grande variedade de artefatos e atividades opcionais, sem muita burocracia para sua utilização. Por conseguinte, permitiu o desenvolvimento do SiGPes rapidamente, num contexto adaptativo e leve que a metodologia propõe e dentro do tempo reduzido para conclusão deste trabalho.

Igualmente, além de o PU permitir um desenvolvimento interativo e incremental do sistema, é muito flexível e aberto a inclusão de praticas interessantes de outros métodos interativos (*eXtreme Programming – XP*, *Scrum* e outros). Por exemplo: as práticas de XP de desenvolvimento guiados por testes, refatoração e integração; e a sala comum de projeto (*war room*) e da prática de reunião diária do *Scrum*. O PU, ainda, combina as melhores práticas comumente aceitas, como ciclo de vida iterativo e desenvolvimento guiado por risco, em uma descrição de processo coesa e bem documentada [10].

No decorrer do desenvolvimento do SiGPes, alguns documentos foram necessários para a execução do projeto, levando-se em consideração as fases sugeridas pela metodologia PU. Inicialmente, na fase de Concepção, foi criado o Documento de Visão, a fim de se obter uma descrição das necessidades e funcionalidades gerais do sistema, permitindo, assim, capturar todas as perspectivas que o sistema deve abranger.

A seguir, na fase de Elaboração, foi criado o diagrama de caso de uso a partir da análise de requisitos do sistema. A Figura 9 representa o Diagrama de Caso de Uso da aplicação SiGPes. Através

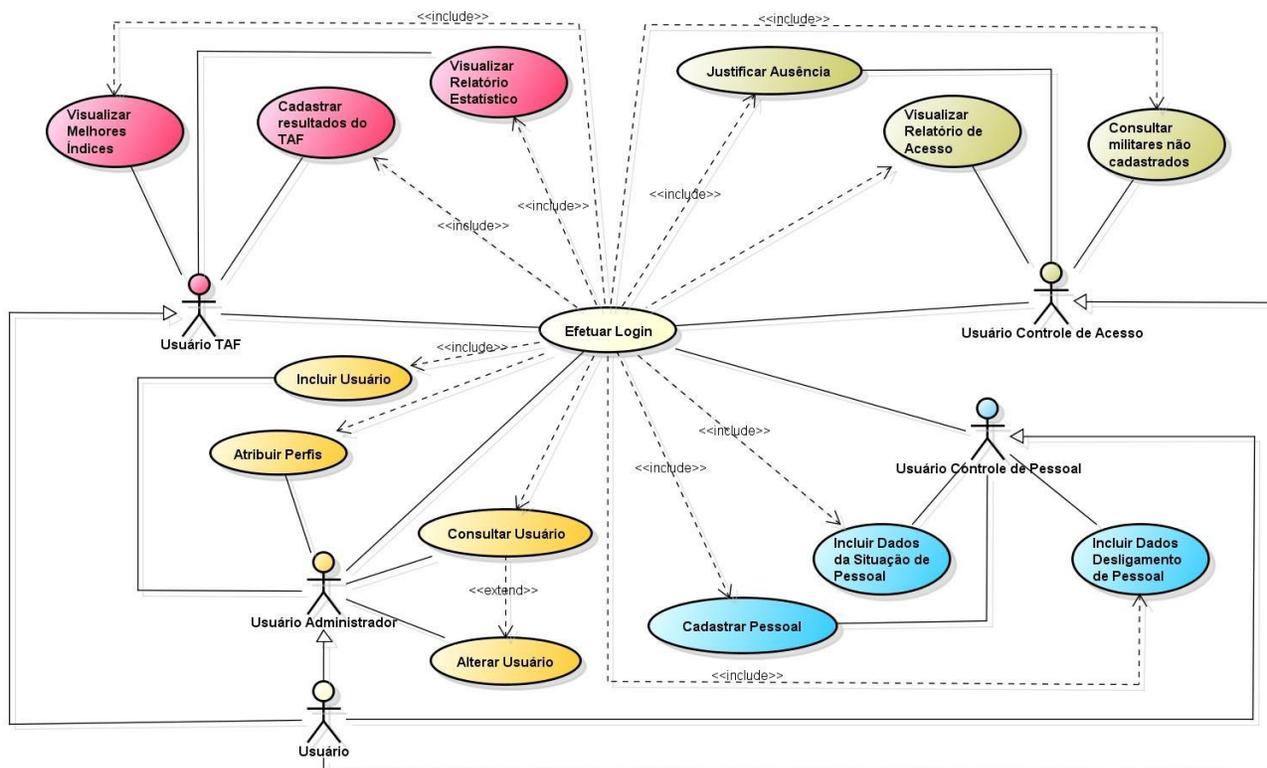


Figura 9. Diagrama de Caso de Uso.

dessa figura é possível identificar as principais funcionalidades implementadas no SiGPes. Entre os casos de uso do sistema, destacam-se: Cadastrar Pessoal, Consultar militares não cadastrados, Justificar Ausência, Cadastrar resultados do TAF, Visualizar melhores índices. Além de serem funcionalidades importantes para a gestão de pessoal, consistem no diferencial do SiGPes em relação aos sistemas legados hora existentes.

O caso de uso Cadastrar Pessoal é vital para a interoperabilidade de dados corretos do SiGPes, tendo impacto direto em outros casos de uso, abaixo descrevo alguns deles:

- a) Cadastrar resultados do TAF, só estarão disponíveis para cadastro dos índices obtidos no TAF aqueles militares que tiverem seu cadastro pessoal efetivado corretamente;
- b) Consultar militares não cadastrados, consequente e automaticamente ao sucesso no caso de uso cadastrar pessoal, o ator responsável pelo controle de acesso terá disponível a relação dos militares que precisam ter seus dados biométricos cadastrados no sistema legado de controle de acesso.

Além disso, o caso de uso Justificar Ausência é mais um exemplo da interoperabilidade entre dados e processos que o SiGPes proporciona. Outrora, era necessária a comparação manual do Relatório de Acesso, com o Cadastro de Pessoal, para se obter a presença dos militares. E, então justificar as ausências quando necessário. Com o SiGPes, essa verificação é feita pelo sistema, através da

coleta dos dados de acesso da base de dados do Sistema de Controle de Acesso e comparação com os militares cadastrados no Sistema de Controle de Pessoal. Por fim, sendo necessária apenas a justificativa das ausências.

O objetivo dos casos de uso supracitados é facilitar a interação entre os dados e processos dos sistemas que fazem parte do SiGPes, processá-los e persistir o resultado desse processamento em um SGBD (neste caso o MySQL). Enquanto que os casos de uso Visualizar Melhores Índices, Visualizar Relatório Estatístico e Efetuar *Login* são funcionalidades incorporadas aos sistemas legados, com o intuito de melhorá-los quanto aos aspectos principalmente de segurança e acesso às informações.

Especificamente, os casos de uso que dizem respeito a segurança do sistema, provendo características de autenticação e autorização ao SiGPes, são os de Incluir Usuário, Alterar Perfis, Consultar Usuário e Alterar Usuário. Igualmente, esses casos de uso proporcionam aos sistemas legados uma melhor distribuição de responsabilidades, além de contribuir para a descentralização de funções, permitindo uma maior rapidez na inserção de dados relevantes ao sistema. Como, por exemplo, a inclusão dos índices obtidos no Teste de Avaliação Física, que outrora, poderia ser feito apenas por um único usuário.

No transcorrer da fase de elaboração foi construído o diagrama de classes do SiGPes. Por opção do autor e com o intuito de proporcionar um melhor entendimento do sistema ao leitor, a Figura 10

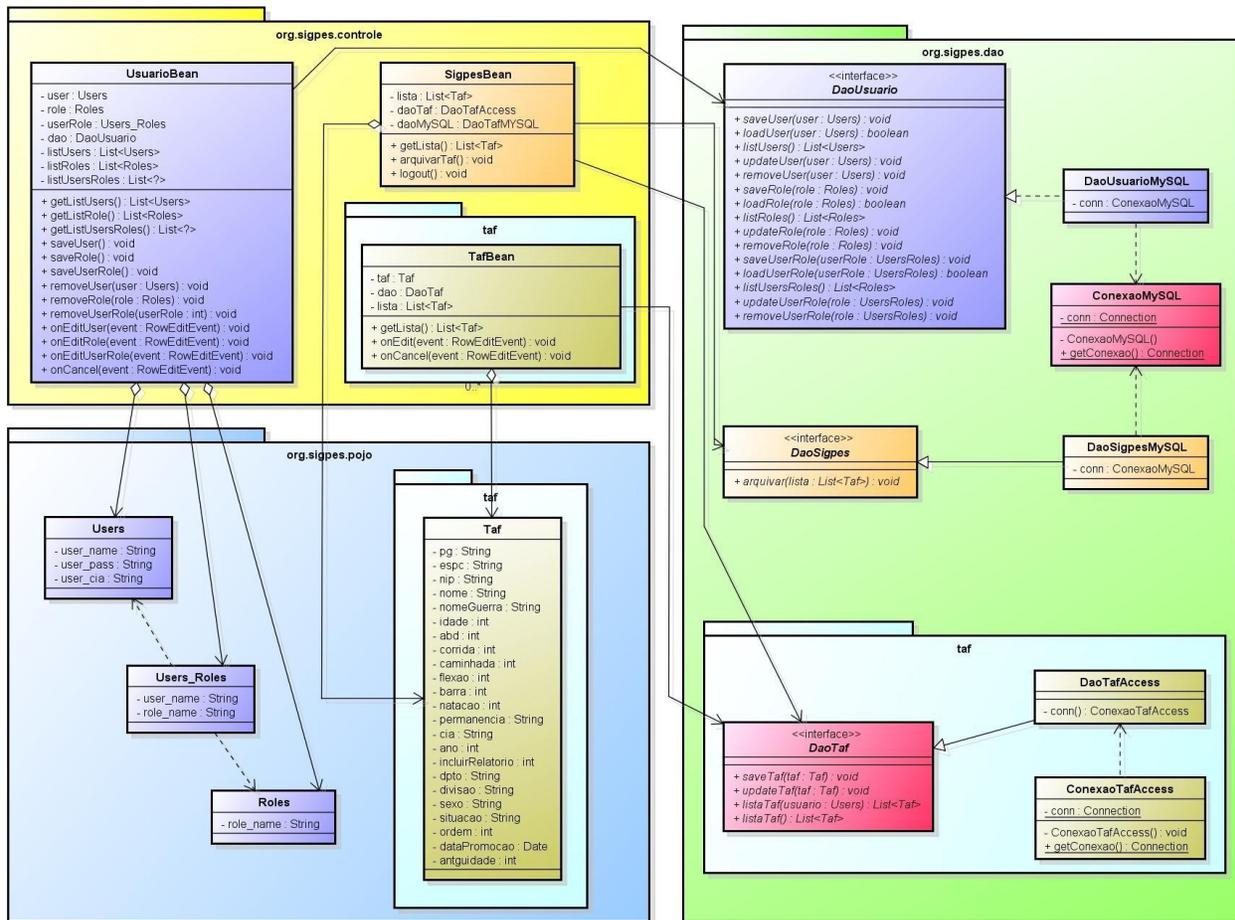


Figura 10. Diagrama de Classes.

mostra esse diagrama num contexto reduzido, levando em consideração apenas o sistema de teste de avaliação física. Entretanto, apesar de a figura não conter todas as classes do sistema, com a sua análise é possível identificar as principais classes que foram necessárias para construção do sistema, a fim de atender os seus principais objetivos de interoperabilidade de dados e processos. As classes referentes aos outros sistemas envolvidos seguem os mesmos princípios de interação entre componentes das classes do sistema de TAF.

O leitor perceberá com a análise da Figura 10 que a classe de controle *SigpesBean* possui, diferentemente das outras classes de controle, dois atributos do tipo DAO (*Data Access Object*), que terá seu conceito aprofundado na subseção seguinte. Isso ocorre pelo fato das outras classes de controle estar diretamente relacionadas com a disponibilização de funcionalidades, que muitas vezes, já são disponibilizadas pelo sistema legado. Enquanto que o *SigpesBean* tem como principal atribuição, coletar dados e prover interoperabilidade (método *arquivarTaf()*) entre os sistemas envolvidos e ele mesmo.

Em seguida, e já na fase de construção do SiGPes, foi utilizada a linguagem *Java* para a implementação do código. Essa linguagem disponibiliza uma variedade de características importantes (serão descritas na próxima subseção) que permitiram ao desenvolvedor desse trabalho, construir o projeto de maneira a atender os requisitos funcionais e não funcionais do sistema.

Por fim, foram realizados testes de homologação, simulando as atividades de controle de pessoal, controle de acesso e lançamento de índices obtidos no teste de avaliação física, com alguns militares das Seções do Grupamento de Fuzileiros Navais de Salvador responsáveis por estas atividades. Não sendo possível, no entanto, aplicar a fase de transição neste trabalho.

5.3.1. Tecnologias Utilizadas para o Desenvolvimento do SiGPes

Com o intuito de atender os requisitos não-funcionais do SiGPes, apesar de haver outras tecnologias capazes de fazê-lo, como a plataforma *Spring Framework*, foi adotada a plataforma *Java* na edição *Java EE 7 (Java Platform, Enterprise Edition 7)* para o desenvolvimento desse sistema.

O *Java EE* é uma plataforma amplamente usada que contém um conjunto de tecnologias coordenadas que reduz significativamente o custo e a complexidade do desenvolvimento, implantação e gerenciamento de aplicações de várias camadas centradas no servidor. O *Java EE* é construído sobre a plataforma *Java SE (Standard Edition)* e oferece um conjunto de APIs para desenvolvimento e execução de aplicações portáteis, robustas, escaláveis, confiáveis e seguras no lado do servidor [16]. A linguagem de programação padrão do *Java EE* é a *Java*. De acordo com [7], possui características importantes, que foram fundamentais no desenvolvimento do SiGPes:

- (a) É uma linguagem orientada a objetos, o que facilita a transformação das necessidades reais do usuário em funcionalidades do sistema;
- (b) Permite o desenvolvimento de uma aplicação robusta, simples e dinâmica, aumentando a produtividade no desenvolvimento;
- (c) Fornece maior segurança pelo fato de existirem diversos mecanismos para controle de segurança;

- (d) Possui um conjunto bastante completo de interfaces de programação, com ênfase no suporte a comunicação em rede e distribuição.

O *Java EE* a cada lançamento integra novas funcionalidades que se alinham com as necessidades da indústria, melhora a portabilidade de aplicações e aumenta a produtividade do desenvolvedor [18]. Ademais, o modelo de programação *Java EE* incentiva o uso de padrões para criação das aplicações do sistema. No caso específico do SiGPes foram utilizados os seguintes padrões: *Plain Old Java Object (POJO)*, *Data Access Object (DAO)* e *Model-View-Controller (MVC)*.

O padrão de *Persistent Object (PO) POJO*, foi um termo desenvolvido por Martin Fowler, Rebecca Parsons e Josh MacKenzie, em setembro de 2000, com o intuito de designar um objeto *Java* simples dentro do modelo de classes de um projeto. O *POJO* segue um desenho simplificado em contraposição aos *EJBs (Enterprise Java Beans)*. Esse padrão se baseia na ideia de que o projeto é melhor quanto mais simples ele for. No desenvolvimento do SiGPes foi seguida as definições rígidas de estrutura desse padrão: construtor *default* sem argumentos e métodos que seguem o padrão de *getters* e *setters* para seus atributos das classes.

O padrão de Projeto *DAO* tem a finalidade de abstrair o mecanismo de persistência utilizado numa aplicação, encapsulando o acesso aos dados persistidos e sua localização, seja ela num *SGBD* ou arquivo *XML*. Segundo [7], as vantagens de se utilizar o padrão *DAO* é que ele permite organizar a lógica de acesso a dados e encapsular características específicas para facilitar a portabilidade e manutenção, e permite desenvolver e testar toda a camada responsável pela persistência dos dados separadamente do resto da aplicação.

O padrão arquitetural *MVC* é um dos mais usados pelos desenvolvedores, tendo por objetivo promover a separação das camadas de uma aplicação em diferentes níveis, a fim de tornar o *software* mais compreensível e de fácil manutenção. Para tanto, as aplicações baseadas nessa arquitetura, são divididas em três componentes (*Model*, *View* e *Controller*) com funções específicas e interligados (componentes) entre si. O *View* diz respeito a interface gráfica da aplicação exibida para o usuário. O *Model* está relacionado com a acessibilidade e interação dos dados pelo usuário. E por fim, o *Controller* que interpreta as ações do usuário no *View* e as mapeia para chamados do *Model*.

Conforme foi mencionado anteriormente, a plataforma *Java EE* oferece um conjunto robusto de serviços que facilita o desenvolvimento de aplicações e facilitam sua portabilidade e manutenção. Na implementação do SiGPes foram utilizados as *Java EE* tecnologias *JDBC (Java Database Connectivity)*, *JavaMail* e *JSF (Java Server Faces 2.2)*, com componente *PrimeFaces*.

JDBC é uma API *Java* que pode acessar qualquer tipo de dados tabulares, especialmente dados armazenados em um *SGBD*. Essa API ajuda a escrita de aplicações *Java* que gerenciam três atividades de programação: 1) se conectar a uma fonte de dados, como um banco de dados; 2) enviar consultas e atualizações para o banco de dados; e 3) recuperar e processar os resultados recebidos do banco de dados através de consultas [18].

A API *JavaMail* fornece uma estrutura independente de plataforma e de protocolo para construir aplicativos de email e mensagens [18]. Portanto, permite o envio de mensagens de correio eletrônico a partir do SiGPes, que foi desenvolvido na linguagem *Java*. No

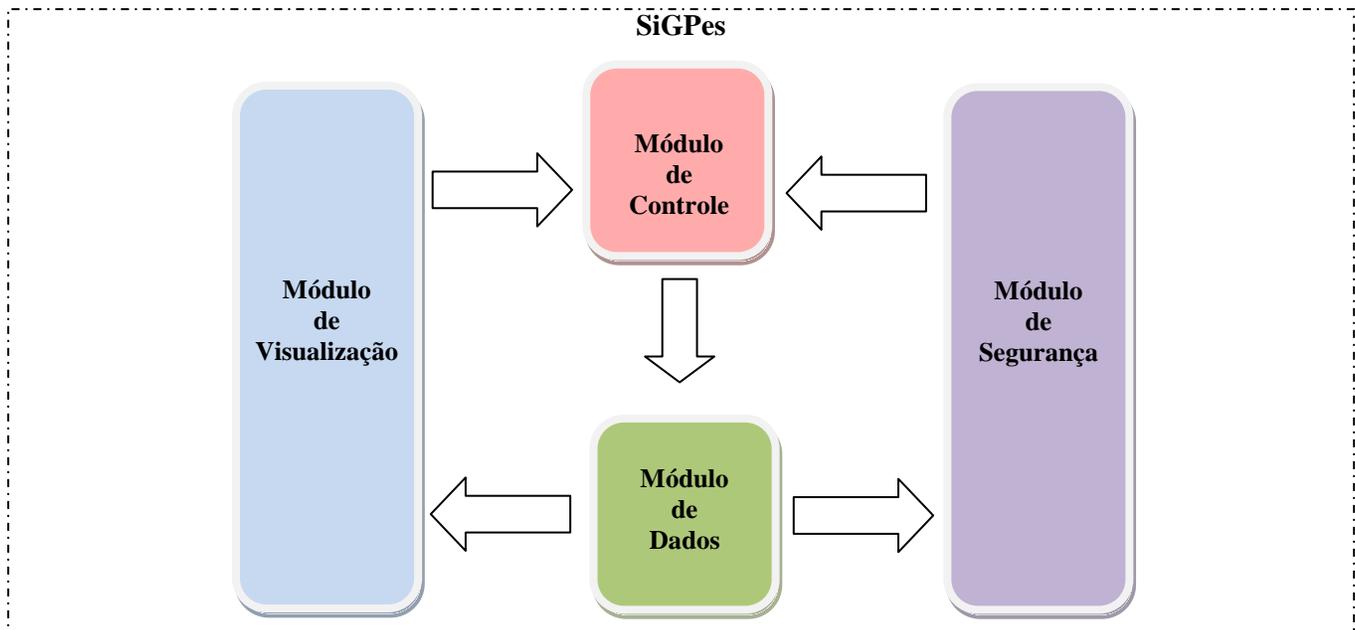


Figura 11. Arquitetura do SiGPes.

SiGPes essa tecnologia foi utilizada para notificar os usuários sobre a inclusão de processos, como o início do período de lançamento dos resultados do TAF, ou confirmar o cadastro dos mesmos no sistema.

A tecnologia JSF 2.2 (*JavaServer Faces*) foi utilizada no componente *View* do SiGPes. O JSF simplifica a construção de *interfaces* de usuário para aplicações *JavaServer*. Consequentemente, facilita e torna rápido o processo de construção de aplicações *web*, além de montagem de componentes de *interface* do usuário reutilizáveis em uma página, que liga esses componentes a uma fonte de dados de aplicativos. O código de *interface* do usuário é executado no servidor, em resposta a eventos gerados pelo cliente, permitindo ao desenvolvedor do aplicativo concentra-se apenas no código-fonte do aplicativo [18].

No SiGPes, a fim de separar claramente a lógica de negócio da camada de apresentação, aumentando a produtividade de desenvolvimento *web*, foram utilizadas: 1) páginas no formato XHTML (*eXtensible Hypertext Markup Language*), que combina as *tags* de marcação HTML com regras da XML; e 2) o componente *PrimeFaces* 5.0, que é uma biblioteca leve, composta apenas de um único JAR sem dependência e configurações necessárias, para aplicações baseadas no framework de *interface* de usuário (IU) para aplicações *Java Web JSF*.

Além dessas tecnologias disponíveis no Java EE 7, também foi utilizado um *Realm* que é uma “*database*” de nomes de usuários e senhas que identificam usuários válidos de uma aplicação *web* (ou conjunto de aplicações *web*), além de uma enumeração da lista de *roles* associados a cada usuário válido. Para tanto, o Tomcat define uma *interface Java* (*org.apache.catalina.Realm*) a ser implementado por *plug-in* para estabelecer esta ligação. Seis *plug-ins* padrões são fornecidos, suportando ligações a várias fontes de informação de autenticação: *JDBCRealm*, *DataSourceRealm*, *JNDIRealm*, *UserDatabaseRealm*, *MemoryRealm* e *JAASRealm* [26]. Esse autor optou por utilizar o *JDBCRealm*, que acessa informações de autenticação armazenadas em um banco de dados relacional, através de um *driver JDBC*.

5.4. Arquitetura do SiGPes

O SiGPes está dividido em quatro módulos: módulo de segurança, módulo de dados, módulo de visualização e módulo de controle. Esses módulos são responsáveis pela autorização e autenticação de usuários, obtenção e inserção de informações em banco de dados, pela criação de *interface* que possibilite a realização das atividades dos usuários e controle e interação entre os sistemas que são interoperáveis pelo SiGPes. A Figura 11 ilustra a arquitetura do SiGPes.

5.4.1. Módulo de Segurança

O objetivo deste módulo é garantir que o SiGPes seja acessado apenas por usuários que, de fato, necessitem realizar alguma atividade nele, já que o *link* de acesso ao SiGPes está disponibilizado no servidor *web* do GptFNSa, sendo disponibilizado para todas os militares que tem acesso ao mesmo. Para tanto existem diferentes tipos de usuários com permissões específicas.

O módulo permite a associação de perfis de utilização para os usuários do SiGPes. O perfil de usuário é definido de acordo com critérios funcionais do usuário. Os perfis disponíveis na ferramenta são: usuário do TAF, usuário do controle de pessoal, usuário do controle de acesso e usuário administrador.

Os perfis de usuários têm relação direta com as funcionalidades do sistema. O perfil de administrador não possui limitações de acesso. Entretanto, os outros perfis possuem acesso limitado aos recursos que estão ligados aos suas próprias definições. Sendo assim, o usuário do TAF terá acesso apenas às funcionalidades relacionadas ao TAF, como ocorrerá com os outros usuários e suas respectivas funções. Um usuário que não seja o administrador, por exemplo, não poderá visualizar e nem alterar dados dos usuários, que não seja ele mesmo.

A fim de implementar a lógica de segurança, o SiGPes utilizou o *plug-in JDBCRealm*, *interface Java* definida pelo Tomcat, que tem o propósito, como foi visto anteriormente, de disponibilizar acesso a informações de autenticação armazenadas em um banco de dados relacional, através de um *driver JDBC*.

Para concretizar as configurações do *JDBCRealm*, foram necessárias efetuar alterações no servidor Tomcat, a criação de um formulário de segurança, através do qual o usuário efetua o *login* no sistema, conforme Figura 12. E, também, a definição de perfis de usuário e restrição de acesso dos recursos disponíveis no SiGPes, no arquivo *web.xml* da aplicação. Os recursos do SiGPes estão separados em diferentes pastas ou diretórios da aplicação, de acordo com cada sistema legado envolvido.



Figura 12. Arquitetura do SiGPes.

Portanto, todas as vezes que um usuário solicita um serviço do SiGPes que possui restrição, o formulário de segurança é carregado para que sejam colocadas as credenciais de um usuário. É claro no caso de já não existir nenhum usuário autenticado fazendo uso do sistema, ou no caso de um usuário autenticado solicitar um recurso que não está autorizado a dispor. Após o usuário apertar o botão Entrar, o SiGPes verifica no banco de dados as informações e poderá ocorrer três situações:

- (a) No caso de inconsistência das informações enviadas, *login* ou senha inexistentes, o usuário receberá a informação da incoerência e a possibilidade tentar novamente efetuar o *login*;
- (b) No caso de autenticação do usuário, ou seja, os dados de *login* e senha estiverem corretos, será verificada a situação de autorização para o recurso solicitado;
- (c) Se autorizado o usuário visualizará as informações do recurso, caso contrário será informado que não possui autorização para tal.

5.4.2. Módulo de Dados

Nesse módulo é realizado busca, inserção, alteração e exclusão de dados no SiGPes. Dados referentes aos militares que trabalham na OM, usuários, situação do militar, situação de desligamento de militar, justificativa de ausência, departamentos, índices obtidos no TAF são inseridos, consultados e armazenados de acordo com esse módulo.

Para a implementação da lógica do módulo de Dados, foi utilizada a API JDBC pelas classes da camada de persistência, que por sua vez utilizam os padrões de persistência DAO e POJO. JDBC disponibiliza mecanismos de acesso aos dados, sejam eles para consulta ou atualizações, armazenados em banco de dados relacional, especialmente quando esse armazenamento se dá num SGBD. Já os padrões de persistência: DAO encapsula o acesso aos dados persistentes e sua localização; e o POJO propicia objetos *Java* simples que serão manipulados tanto para persistência quanto para visualização de informações na *interface* do usuário.

JDBC é uma API *Java* que pode acessar qualquer tipo de dados tabulares, especialmente dados armazenados em um SGBD. Essa

API ajuda a escrita de aplicações *Java* que gerenciam três atividades de programação: 1) se conectar a uma fonte de dados, como um banco de dados; 2) enviar consultas e atualizações para o banco de dados; e 3) recuperar e processar os resultados recebidos do banco de dados através de consultas.

O módulo de Dados sofre influência direta do módulo de segurança, já que para ter acesso a qualquer informação, o usuário necessitar ter as devidas permissões. Então, após o usuário ser autenticado e autorizado, o módulo de Dados poderá interagir com o módulo de Visualização (solicitações do usuário), através do módulo de Controle, responsável por alterar o estado do módulo de Dados. Por fim, o usuário receberá as informações atualizadas do módulo de Dados, através do módulo de Visualização.

5.4.3. Módulo de Visualização

Constitui o único meio de acesso do usuário ao SiGPes, através desse módulo acontece todas as interações do usuário com o sistema. Como visto na subseção anterior, o módulo de visualização externaliza o estado do módulo de Dados, por intermédio do módulo de Controle.

Nesse módulo foi utilizada a tecnologia JSF 2.2, com o intuito de simplificar a construção de *interface* do usuário, disponibilizadas através de páginas XHTML, e com a utilização do componente *PrimeFaces* 5.0, que auxilia e tornam fáceis a construções de aplicações baseadas nessa tecnologia.

5.4.4. Módulo de Controle

É o módulo responsável por gerenciar e interligar todos os outros módulos, como ilustrado na Figura 11. Desde solicitar as alterações no módulo de Dados para que possam ser disponibilizadas para os usuários, através do módulo de Visualização, até permitir a interação dos módulos de Dados e Segurança, a fim de proporcionar a proteção das informações constantes do SiGPes. Esse módulo segue as características do componente *Controller* do padrão arquitetural MVC, interpretando as ações do usuário no componente *View* e as mapeando para chamadas do componente *Model*. Portanto, é nesse módulo que está o centro de interoperabilidade dos sistemas legados citados neste trabalho.

Para cumprir as responsabilidades do módulo de Controle, no SiGPes, foram utilizadas classes de modelo que interagem com os componentes JSF disponibilizados no módulo de Visualização, as chamadas *Managed Beans* ou *Bean* Gerenciado JSF. Essas classes servem para manipular os dados solicitados pelos usuários através do módulo de Visualização.

5.5. Avaliação do SiGPes

O SiGPes foi apresentado a um grupo de seis militares do Grupamento de Fuzileiros Navais de Salvador, tendo como objetivo verificar se a ferramenta apresentada neste trabalho atende as necessidades deles.

Em seguida a apresentação, as funcionalidades do SiGPes foram avaliadas levando em consideração o resultado da pesquisa de satisfação, Figura 13, respondida pelos militares da Seção de Pessoal, de Operações e de Inteligência, responsáveis pela utilização dos sistemas de Controle de Pessoal, Teste de Avaliação Física e Controle de Acesso, respectivamente.

O resultado obtido do com a pesquisa de satisfação foi relevante. Tendo o SiGPes, sido aprovado por todos os entrevistados, especialmente aos militares responsáveis pelo Controle de Acesso, pois a

Controle de Pessoal
Controle de Acesso
TAF

Pesquisa de Satisfação sobre o Sistema de Gestão de Pessoal

Entrevistado

Nome: Setor: Função:

Questionário

1) O SiGPes pode ser facilmente compreendido e utilizado?	<input type="button" value="Sim"/>	2) O SiGPes atende as suas necessidades?	<input type="button" value="Sim"/>
3) Em sua opinião, o SiGPes torna mais dinâmico as suas atividades?	<input type="button" value="Sim"/>	4) Em sua opinião, o SiGPes torna suas atividades mais seguras?	<input type="button" value="Sim"/>
5) O SiGPes, pode trazer benefícios a gestão de pessoal do GptFNSa? Explique.	<input type="text"/>		
6) Qual a funcionalidade do SiGPes você mais gostou? Por quê?	<input type="text"/>		
7) Qual a funcionalidade do SiGPes você não gostou? Por quê?	<input type="text"/>		
8) Qual melhoria você sugere ao SiGPes?	<input type="text"/>		

Figura 13. Pesquisa de Satisfação

verificação de presença é praticamente inviável com o sistema atual, já que o processo é quase totalmente feito manualmente. Com o SiGPes essa verificação é automática, sendo disponibilizado para justificativa de ausência apenas aqueles militares que de fato não deram entrada no quartel.

Outra melhoria importante destacada no resultado da pesquisa, foi a rapidez de lançamento dos índices obtidos no Teste de Avaliação Física, com a utilização do SiGPes. Antes esses lançamentos eram centralizados na Seção de Operações, sendo permitida a utilização do sistema apenas por um militar. Com o SiGPes, a responsabilidade de lançamento é dada a um ou mais militares por Companhia (setor), o que torna o processo mais rápido. Entretanto, para fiscalizar esses lançamentos a ferramenta ainda dispõe de mecanismos de controle, dados ao Supervisor do TAF para fiscalizar o correto lançamento dos índices obtidos e eventuais discrepâncias na apuração do resultado final do TAF.

Em relação ao Controle de Pessoal, apesar de atender as necessidades do setor responsável por esse controle, com por exemplo: disponibilização de meios de controlar o processo de desligamento de pessoal do GptFNSa; ter uma melhor compreensão da situação funcional de cada militar; e o compartilhamento das atualizações dos dados dos militares que servem neste Grupamento com as outras seções que fazem uso delas. Os militares da Seção de Pessoal levantaram algumas melhorias que podem ser feitas no sistema, tais como: inclusão de controle de inspeções de saúde e plano de chamada, bem como a inclusão de outros meios de busca de militares, além do Nip (matricula financeira de cada militar), como por nome, posto ou graduação, especialidade, data de promoção.

6. CONCLUSÕES

O GptFNSa possui uma estrutura administrativa em que delega a Seção de Pessoal, Seção de Inteligência e Seção de Operações as funções, respectivamente, de controle de pessoal, controle de acesso e lançamento dos índices obtidos no teste de avaliação física. No entanto é recorrente a incidência de problemas na execução das três atividades supracitadas, sendo que muitos desses problemas poderiam ser evitados com a existência de um sistema capaz de fazer interoperar essas atividades.

O principal objetivo deste trabalho, de criar um sistema capaz de prover interoperabilidades entre os sistemas legados, foi cumprido. Desenvolveu-se o SiGPes, um sistema computacional que fornece funções básicas de gestão de pessoal, visando fornecer apoio nas tarefas de responsabilidade do GptFNSa frente ao Administração Naval, na figura do Comando do Pessoal de Fuzileiros Navais, situado na cidade do Rio de Janeiro.

O SiGPes é implementado em *Java* e fornece serviços através da plataforma *web*, mais comumente utilizada para disponibilização de aplicações no âmbito da Marinha do Brasil, que possibilitam a interoperabilidade entre sistemas envolvidos nesse trabalho, com a utilização de *web services* e da linguagem XHTML baseada nas regras do XML.

Esse sistema permite o cadastro, alteração, exclusão, consulta e visualização gráfica de informações contidas nos sistemas envolvidos, além do cadastro administrativo de departamentos, posto e graduação, situações em que o militar pode se enquadrar e, também, usuários.

O diferencial oferecido pelo SiGPes consiste na utilização dos mecanismos de interoperabilidade entre sistemas para coleta de dados nas diversas bases de dados (*Microsoft Office Access*) dos sistemas legados, que não propiciam muitas funcionalidades, e a geração de informação consistente persistidas num SGBD. Além, de propiciar uma reestruturação de todo o sistema hoje utilizado, sem a necessidade de uma mudança drástica, já que os sistemas legados podem continuar sendo utilizados, possibilitando adequar os gastos com o orçamento disponível para tal.

É possível concluir que este trabalho é de relevante importância para o GptFNSa, levando em consideração que o mesmo tem a pretensão de garantir a qualidade dos serviços prestados pelas Seções acima mencionadas, cujas atividades impactam na gestão do pessoal militar dessa Instituição. Igualmente, é de se considerar a possibilidade que o SiGPes disponibiliza para utilização de técnicas como BI (*Business Intelligence*)¹⁴ para torná-la uma ferramenta de apoio a decisão, já que agora os dados mais importantes das atividades acima, são processados e persistidos num SGBD.

7. TRABALHOS FUTUROS

Embora o principal objetivo desse trabalho tenha sido alcançado, a análise dos resultados do estudo de caso SiGPes revela que esse sistema possibilita o desenvolvimento de trabalhos futuros. Existem funcionalidades que podem ser implementadas com o intuito de atender uma maior demanda de sistemas que podem interoperar e novos requisitos funcionais.

Como sugestão para trabalhos futuros, pode-se citar:

- I) Estudo mais aprofundado dos recursos oferecidos pela linguagem XML e *Web Services* para interoperabilidade entre os sistemas por meio do SiGPes.
- II) Inclusão de outros sistemas ao conjunto de sistemas interoperáveis através do SiGPes, como o controle de inspeções de saúde e plano de chamada dos militares do GptFNSa.
- III) Utilização das técnicas de BI para tornar o SiGPes efetivamente uma ferramenta de apoio a decisão.
- IV) Aprofundar os estudos feitos neste trabalho no sentido de validar os resultados obtidos.

8. REFERÊNCIAS

- [1] Albinader Neto, J. A.; LINS, R. D. *Web Services em Java*. – Rio de Janeiro: Brasport, 2006.
- [2] Biometrius. BioConfigurador Acesso - Configurador para Equipamentos de Acesso; Disponível em http://www.biometrus.com.br/biowiki/doku.php?id=publico:configurador:configurador_main. Acessado em 18-06-2014.
- [3] Coyle, F. P. *XML, web services and data revolution*. Wesley Information Technology Series, 2002.
- [4] Cubera, F.; Nagy, W. A.; Weerawarana, S.; “Web Services: Why and How”; IBM T. J. Watson Research Center; 09-08-2001; Disponível em http://www.researchgate.net/publication/2406580_Web_Services_Why_and_How/file/3deec5239ac652278e.pdf. Acessado em 21-06-2014.
- [5] Edwald, Tim; “Understanding XML Web Services The Web Services Idea”; Microsoft Corporation, 27-09-2002; Disponível em http://www.projectlounge.com/QuickPlace/buildingvision/Main.nsf/h_523105049F00C9D885256E0F0041E06D/DA34053151CCC24D85256CB1000F9D8B/?OpenDocument. Acessado em 28-06-2014.
- [6] Facunte, E. *Delphi 7: Internet e Banco de Dados*. Rio de Janeiro: Brasport, 2003.
- [7] Gonçalves, L. A. *SARA-ES: Sistema para Acompanhamento de Registros Acadêmicos do Ensino Superior*.
- [8] GptFNSa. Missão; Disponível em <http://www.mar.mil.br/gfnsalv>. Início>Missão; Acessado em 14-05-2014.
- [9] Kratz, R. A. *Fábrica de adequação de conteúdo de ensino para objetos de aprendizagem reutilizáveis (RLOs) respeitando a norma SCORM*. In: Universidade do Vale do Rio dos Sinos, 2005. Disponível em <http://www.professores.uff.br/screspo/kratz.pdf>. Acessado em 30-06-2014.
- [10] Larman, C. *Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*. 3ª edição. Porto Alegre: Bookman, 2005.
- [11] Leandro, R. C. *Introdução aos Web Services*. Disponível em <http://www.comp.ita.br/~gian/tes04/trabalhos/web-services-all.signed.pdf>. Acessado em 04-07-2014.
- [12] Manes, A. T.; “Web Services Basics”; Disponível em <http://www.informit.com/articles/article.aspx?p=99979>. Acessado em 26-06-2014.
- [13] Martins, R. S. M. *Composição Dinâmica de Web Services*. In: Universidade do Vale do Rio dos Sinos, 2007. Disponível em <http://www.professores.uff.br/screspo/rogerio.pdf>. Acessado em 29-06-2014.
- [14] Mcgovern, J.; Tyagi, S.; Stevens, M.; Mathew, S.; “Java Web Services Architecture”. Morgan Kaufmann. San Francisco – CA – USA, 2003.
- [15] Nagappan, R, Skoczylas, R, Sriganesh, R. P.; “Developing Java Web Services: Architecting and Developing Secure Web Services Using Java”; John Wiley & Sons; Indianapolis – Indiana – USA; Fevereiro/2003.
- [16] Netbeans. www.netbeans.org.
- [17] Oellermann, Willian L. *Architecting Web Services*. Apress, 2001.
- [18] Oracle. www.oracle.com.
- [19] Oya, J. K.; Silva, D. T. *Interoperabilidade entre aplicações utilizando Web Services e SOA*. In: Universidade Federal do Pará, 2006. Disponível em <http://www.portaltcc.icen.ufpa.br/portaltcc/principal/Tcc/action.do?act=download&id=51>. Acessado em 28-06-2014.
- [20] Portal de Governo Eletrônico. *O que é interoperabilidade*. <http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-de-interoperabilidade/o-que-e-interoperabilidade>. Acessado 16-07-2014.

¹⁴ Basicamente, refere-se ao processo de coleta, organização, análise, compartilhamento e monitoramento de informações que oferecem suporte a gestão de negócios.

- [21] Potts, S.; Kopack, M. Aprenda em 24 horas Web Services. Rio de Janeiro: Elsevier, 2003.
- [22] PUC-RIO – Certificação Digital Nº 0220942/CA. Capítulo 3. Serviços na Web (Web services). Disponível em http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0220942_04_cap_03.pdf. Acessado em 30-06-2014.
- [23] Santanchè, A.; Teixeira, C. A. C. Mais pontes e menos ilhas estratégia para integração de software educacional. In: Simpósio Brasileiro de Informática na Educação, 2003. Disponível em <http://www.lis.ic.unicamp.br/~santanch/publications/SBIE2002-Pontes.pdf>. Acessado em 01-07-2014.
- [24] Scopel, M. WSMELE: uma arquitetura para integração de serviços educacionais usando dispositivos móveis na formação de comunidades virtuais espontâneas. In: Universidade do Vale do Rio dos Sinos, 2005. Disponível em <http://www.professores.uff.br/screspo/scopel.pdf>. Acessado em 01-07-2014.
- [25] Software AG. Web Services. Making the most of your business assets. Software AG, Março/2002.
- [26] Tomcat. Realm how to. Disponível em <http://tomcat.apache.org/tomcat-7.0-doc/realms-howto.html>. Acessado 02-08-14.
- [27] W3C. Extensible Markup Language (XML). Disponível em <http://www.w3.org/XML>. Acessado em 21-06-2014.

9. APÊNDICE

9.1. Outras Telas do SiGPes

1. A Figura 14 apresenta a pontuação obtida pelo militar do GptFNSa em cada modalidade do Teste de Avaliação Física.

Modalidade	Pontuação
Nip	77777777
Corrida	70
Caminhada	0
Abdominal	80
Flexão na Barra	90
Flexão no Solo	0
Natação	100
Permanência	50
TOTAL	390

Figura 14. Pontuação no TAF.

2. A Figura 15 ilustra a tela de cadastro de militar. Para tanto, foi utilizado o componente *dialog* da *PrimeFaces*, um *panel widget* de sobreposição de elementos. É importante salientar que quando o usuário da Seção de Pessoal que é responsável pelo cadastro de militares clica no botão Salvar, o SiGPes por sua vez se encarrega de persistir os dados na base de dados do Controle de Pessoal e replicar essa persistência nas bases de dados do Controle de Acesso e do Sistema de

Cadastro de Teste de Avaliação Física, garantindo assim a integridade entre as bases de dados.

P_G *	<input type="text"/>
Espc	<input type="text"/>
Nip *	<input type="text"/>
Nome *	<input type="text"/>
NomeGuerra	<input type="text"/>
CorpoQuadro	<input type="text"/>
DataNascimento	<input type="text"/>
DataIncorporação	<input type="text"/>
DataPromoção *	<input type="text"/>
Antiguidade *	<input type="text" value="0"/>
Setor	Setor
Função	<input type="text"/>

Salvar

Figura 15. Cadastro de Militar.

3. A Figura 16 mostra a tela de filtro ou busca de militar por Nip (matricula financeira de cada militar).

P_G	Espc	Nip	Nome	Nome de Guerra
CMG	(FN)	88888888	MARCIO DA SILVA CORDEIRO	CORDEIRO

Digite o Nip:

Figura 16. Busca de Militar.

4. As Figuras 17, 18 e 19 dizem respeito ao Controle de Pessoal do Grupamento de Fuzileiros Navais de Salvador. O militar com o perfil para ter acesso a esse recurso do SiGPes, poderá efetuar o cadastro, alteração e exclusão dos dados de todos os militares que trabalham ou já trabalharam no GptFNSa, bem como fazer o controle de desligamento do serviço ativo da Marinha e situação funcional de cada militar.

SISTEMA DE GESTÃO DE PESSOAL

Home Controle de Pessoal

Militares

Controle DSAM
Relação Geral
Sair

P_G	Espec	Nome	Nome de Guerra	C_Q	DataNasc	DataInc	DataProm	Ant	Setor	Função	DSAM	SIT
CMG	(FN)	77777777	JULIO MASCARENHAS SALGADO	SALGADO	CFN FN	03/03/1966	03/03/1984	13/12/2000	1	01	Comandante	Yes
CMG	(FN)	88888888	MARCIO DA SILVA CORDEIRO	CORDEIRO	CFN FN	05/05/1970	12/05/1990	13/12/2012	2	11	Oficial de Pessoal	No
CMG	(FN)	66666666	PAULO CRUZ ROBERTO	CRUZ	CFN FN	12/11/1975	14/05/1995	11/06/2013	3	30	Oficial de Operações	No
CMG	(FN)	99999999	JULIO NETO MASCARENHAS	MASCARENHAS	CFN FN	10/12/1978	11/02/1996	13/12/2013	4	10	Comandante de Companhia	No
CMG	(FN)	44444444	ROBERTO DA COSTA PEREIRA	PEREIRA	CFN FN	10/12/1979	01/03/1998	13/12/2013	5	12	ASSESSOR JURÍDICO	No

Filter: Digite o Nip: 88888888 [Filtrar] [Mostrar Relação Geral]

[+ Novo]

Figura 17. Relação Geral de Militares do GptFNSa.

SISTEMA DE GESTÃO DE PESSOAL

Voltar Controle de Pessoal

Dados da Situação do SALGADO atualizados!

ATUALIZAÇÃO DE SITUAÇÃO DO MILITAR

P_G	Espec	NomeGuerra	SituaçãoOM	Sit_Funcional	Inicio	Término	InfoCompl	DocAPS	OM_Origem	DataAPS	DocDLG	OMDestino	DataDLG
CMG	(FN)	SALGADO	Servir Normalmente	Servindo				Circ 2/2013	CIASC	10/02/2014			

Figura 18. Controle da Situação dos Militares.

SISTEMA DE GESTÃO DE PESSOAL

Home Controle de Pessoal

Dados do Desligamento do SALGADO atualizados!

Controle de Desligamento do Serviço Ativo da Marinha

P_G	Espec	NomeGuerra	QuitaçãoFN	Portaria_Agregacao	Portaria_DSAM	OS_DSAM	MSG_DSAM	PCTS	DCIN	DSAM
CMG	(FN)	SALGADO	Ofício 234/2014	256/2014 CPesFN	257/2014 CPesFN	354/2014	R-171700Z	Of. 504/2014 CPesFN	Ofício 356/2014	Yes

Figura 19. Controle de Desligamento do Serviço Ativo da Marinha.

9.2. Listas de Abreviaturas e Siglas

API	<i>Application Program Interface</i>
BI	<i>Business Intelligence</i>
CPD	Centro de Processamento de Dados
CORBA	<i>Common Object Request Broker Architecture</i>
DAO	<i>Data Access Object</i>
DCOM	<i>Distributed Component Object Model</i>
DNS	<i>Domain Naming System</i>
EJB	<i>Enterprise Java Bean</i>
GptFNSa	Grupamento de Fuzileiros Navais de Salvador
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
Java EE	<i>Java Platform, Enterprise Edition</i>
JDBC	<i>Java Database Connectivity</i>
JSF	<i>JavaServer Faces</i>
MB	Marinha do Brasil
MIME	<i>Multipurpose Internet Mail Extensions</i>
MVC	<i>Model-View-Controller</i>
OM	Organização Militar
POJO	<i>Plain Old Java Object</i>
PO	<i>Persistent Object</i>
PU	Processo Unificado
RPC	<i>Remote Procedure Call</i>
S1	Seção de Pessoal
S2	Seção de Inteligência
S3	Seção de Operações
SiGPes	Sistema de Gestão de Pessoal
SGBD	Sistema Gerenciador de Banco de Dados
SGML	<i>Standar Generalized Markup Language</i>
SOAP	<i>Simple Object Access Protocol</i>
TAF	Teste de Avaliação Física
TCP/IP	<i>Transmission Control Protocol/ Internet Protocol</i>
UDDI	<i>Universal Description, Discovery and Integration</i>
URL	<i>Universal Resource Loader</i>
W3C	<i>Worldwide Web Consortium</i>
WSDL	<i>Web Service Description Language</i>
XHTML	<i>eXtensible Hypertext Markup Language</i>
XML	<i>eXtensible Markup Language</i>
XP	<i>eXtreme Programming</i>