



FREDERICO ANTONIO LOIOLA MENDES

**SISTEMA DE GERENCIAMENTO DE
PUBLICAÇÕES SGP DA SEXTA REGIÃO DA
POLÍCIA MILITAR DE MINAS GERAIS**

LAVRAS – MG

2014

FREDERICO ANTONIO LOIOLA MENDES

**SISTEMA DE GERENCIAMENTO DE PUBLICAÇÕES SGP DA SEXTA
REGIÃO DA POLÍCIA MILITAR DE MINAS GERAIS**

Monografia apresentada ao Colegiado do Curso de
Ciência da Computação da Universidade Federal
de Lavras como parte das exigências do curso de
Ciência da Computação para obtenção do título de
Bacharel em Ciência da Computação

Orientador

Profa. Dra. Ana Paula Piovesan Melchiori (DCC/UFLA)

Supervisor

Anderson Pereira Sales

LAVRAS – MG

2014

FREDERICO ANTONIO LOIOLA MENDES

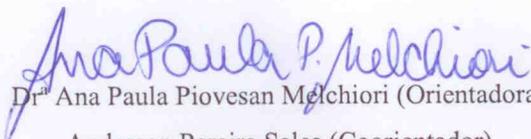
**SISTEMA DE GERENCIAMENTO DE
PUBLICAÇÕES SGP DA SEXTA REGIÃO DA
POLÍCIA MILITAR DE MINAS GERAIS.**

Trabalho de Conclusão de Curso de
Graduação apresentado ao Colegiado do
Curso de Bacharelado em Ciência da
Computação, para obtenção do título de
Bacharel.

APROVADA em 20 de novembro de 2014.

MSc. Antonio Claret dos Santos

Dr. Andre Luiz Zambalde


Dr^a Ana Paula Piovesan Melchiori (Orientadora)
Anderson Pereira Sales (Coorientador)

**LAVRAS-MG
Novembro/2014**

*A todos que acreditaram e me apoiaram para que este sonho se tornasse
realidade.*

AGRADECIMENTOS

Agradeço a Deus por sempre me orientar a seguir os melhores caminhos e estar ao meu lado sempre.

Aos meus pais Mauro e Adriani por me darem força e condições de tornar este sonho realidade e por toda dedicação.

A minha namorada Laura por toda a paciência, carinho e dedicação durante todo este tempo.

A minha família por me apoiarem e por todos os momentos juntos.

Aos professores e professoras do Curso de Ciência da Computação por todos os ensinamentos e orientações transmitidos.

A minha orientadora Profa. Ana Paula pela orientação e paciência que ajudaram na realização deste trabalho.

Ao meu amigo e supervisor Anderson por todo o tempo dedicado e ensinamentos.

A PMMG pela oportunidade de realizar este trabalho.

Se quiser ter uma boa ideia, tenha uma porção de ideias. (Thomas Edison)

RESUMO

A necessidade de novas ferramentas para auxiliar no trabalho administrativo e acompanhar a demanda por informações nas organizações é uma preocupação constante. Neste projeto foi proposto uma nova alternativa para o gerenciamento de dados referentes à ficha funcional dos integrantes da PMMG, no âmbito da 6ª RPM, capaz de facilitar a rotina dos gestores da unidade através da criação de um sistema informatizado utilizando a linguagem de marcação XML para o armazenamento de dados sem o emprego de SGBDs, devido ao grande fluxo de registros, que são gerados diariamente e também, pela dificuldade em transformá-los em informações úteis à organização. A escolha do XML para o desenvolvimento do sistema, justificou-se pela forma estruturada para a armazenagem de dados e também por sua portabilidade entre sistemas. Como resultado podê-se concluir que o sistema irá auxiliar na rotina dos usuários facilitando no armazenamento e recuperação das publicações e que o XML dependendo diretamente de sua aplicação, pode atender às necessidades de um sistema de gerenciamento de dados.

Palavras-Chave: Sistema de Gestão de Informações; Operações em arquivo de Texto; XML.

ABSTRACT

The necessity of new tools to assist in administrative work and monitor the demand for information in organizations is a constant concern. In this project was proposed a new alternative for managing data on the functional form of the members of PMMG within the 6th RPM, capable of facilitate the routine of unit managers by creating a computerized system using the XML markup language for data storage, without the use of DBMSs, given the large flow of records, which are generated daily and the difficulty to transform them into useful information for the organization. The choice of XML for the development of the system was justified by the structured approach to data storage form and also due to its portability between systems. As a result it was concluded that the system will assist in the routine of users facilitating the storage and recovery of publications and the XML depending directly from your application, can attend the needs for a data management system.

Keywords: Management Information System; Operations on text file; XML.

SUMÁRIO

1	Introdução	12
1.1	Contextualização	12
1.2	Justificativa	14
1.3	Objetivos Geral e Específicos	15
1.4	Estrutura deste Documento	16
2	Referencial Teórico	17
2.1	Gerenciamento de Arquivos	17
2.1.1	Operações em Arquivos	19
2.2	XML	19
2.3	HTML e XML	22
2.4	XSL	22
2.5	VBScript	24
2.6	Bootstrap	25
2.7	Trabalhos Relacionados	25
3	Metodologia	27
4	Desenvolvimento	29
4.1	Visão Geral do Sistema	29
4.2	Interface Sistema	32

4.3	Detalhes de Implementação	38
5	Resultados e Experimentos	41
5.1	Experimentos	41
5.2	Resultados	44
6	Conclusões	45
6.1	Trabalhos Futuros	46

LISTA DE FIGURAS

3.1	Metodologia do projeto. (Fonte: Adaptada de (JUNG; ENG, 2009)) .	27
4.1	Diagrama de casos de uso do usuário do Sistema de Gerenciamento de Publicações.	30
4.2	Diagrama de casos de uso do Sistema de Gerenciamento de Publicações.	31
4.3	Interface geral do Sistema de Gerenciamento de Publicações.	32
4.4	Interface para cadastrar do Sistema de Gerenciamento de Publicações.	33
4.5	Interface para consultar do Sistema de Gerenciamento de Publicações.	34
4.6	Resultado da consulta do Sistema de Gerenciamento de Publicações. .	34
4.7	Interface para alteração e remoção do Sistema de Gerenciamento de Publicações.	35
4.8	Resultado para alteração e remoção do Sistema de Gerenciamento de Publicações.	35
4.9	Interface para alteração do Sistema de Gerenciamento de Publicações.	36
4.10	Diagrama do Sistema de Gerenciamento de Publicações.	37
4.11	Modelo de dados para o documento XML.	37
4.12	Diagrama do fluxo de Informações do Sistema de Gerenciamento de Publicações.	38
4.13	Interface final do Sistema de Gerenciamento de Publicações.	40

LISTA DE QUADROS

5.1	Conjunto de dados utilizados nos testes.	42
5.2	Conjunto de dados utilizados nos testes.	42
5.3	Conjunto de dados utilizados nos testes.	42
5.4	Resultados dos testes.	43

1 INTRODUÇÃO

1.1 Contextualização

Dentro das organizações públicas a produção de dados tem aumentado rapidamente, tornando necessária a utilização de ferramentas informatizadas para agilizar as rotinas de trabalho. O emprego das tecnologias da informação nos mais variados setores e o desenvolvimento das ferramentas de coleta de dados, colaboram para este avanço e como consequência, evidencia-se a necessidade em acompanhar o desenvolvimento das inovações.

O uso de tecnologias que possam colaborar com os processos de tomada de decisão é um dos objetivos das organizações para atender as demandas por informações, visto que, contribui para uma visão sistêmica das atividades desempenhadas, reduz as barreiras entre os variados campos do conhecimento e proporciona maior capacidade de desenvolvimento. Neste contexto, a organização das informações na geração de relatórios podem potencializar os recursos informacionais de uma organização e estão diretamente relacionados ao sucesso e ao acompanhamento das necessidades da sociedade (DUARTE; SILVA; COSTA, 2007).

Analisando que a Constituição Federal do Brasil (artigo 37), trata sobre vários princípios da administração pública, dentre os quais o “princípio da publicidade”, que orienta sobre a importância em manter plena transparência dos seus atos e decisões e o princípio da “eficiência” do qual se deve buscar o aperfeiçoamento na prestação dos serviços públicos, desta maneira, afirma-se que a PMMG é uma autarquia estadual e segue o que a norma descrita acima prescreve, no entanto, produz uma elevada quantidade de dados dispersos em documentos impressos e digitais, inexistindo até então, qualquer ferramenta automatizada capaz de fornecer acesso de maneira rápida e precisa a estes dados.

Diante deste cenário, constatou-se que manipular grandes quantidades de dados pode ser um processo complexo e implicar em consultas lentas (KROTH; NEUMANN, 2013). Assim, torna-se cada vez mais necessário recorrer à novas técnicas e ferramentas automatizadas que possam auxiliar na geração de conhecimento e extração de informações úteis aos seus gestores.

Sabemos que a evolução tecnológica afeta todas as atividades sociais, inclusive a segurança pública, principalmente na iniciativa de prestar serviços com mais eficiência. Esta tendência determina o interesse da Polícia Militar de Minas Gerais (PMMG) na exploração de novas tecnologias, pois, quanto maior a fluidez das informações relevantes, melhor se torna a capacidade de resposta da organização à comunidade local ou regional.

Encontrar novas soluções para resolver o problema de gestão de informação, foi a motivação do trabalho da equipe da 6ª Região de Polícia Militar (6ª RPM). Para isto, foi necessário o desenvolvimento de um sistema de gerenciamento de publicações, do qual os dados foram armazenados em um servidor e poderão ser facilmente recuperados quando necessário. Cabe ressaltarmos que para termos obtido o sucesso nesta proposta, alguns objetivos foram traçados e alcançados, entres eles, a sistematização das rotinas e a não sobrecarga do servidor.

A 6ª Região de Polícia Militar situada na cidade de Lavras é o comando regional responsável pelo policiamento na região centro sul do Estado de Minas Gerais. Atendem uma população superior a 860.000 habitantes, distribuídas em 44 municípios e 3 distritos. Contam com um efetivo superior a 1300 policiais, que atuam em diferentes áreas de policiamento. É referência no Estado com seus projetos tecnológicos voltados para o campo da segurança pública entre eles o Portal Corporativo Regional criado para integrar e coordenar as ações internas da administração das Unidades Operacionais da 6ª RPM.

Desta maneira o Sistema desenvolvido foi colocado a disposição dos usuários como parte do Portal Corporativo Regional que serve como ferramenta de integração entre os grupos de policiamento e atualmente engloba outras 2 regiões a 18ª RPM situada em Poços de Caldas e a 17ª RPM situada em Pouso Alegre.

Para o desenvolvimento este projeto adotou a linguagem de marcação XML para a manipulação das informações, devido às suas vantagens no tratamento de dados e principalmente a sua portabilidade (LUNKES *et al.*, 2003). O XML é comumente utilizado em sites de e-commerce devido a sua comunicação e migração de dados, é uma linguagem atual e com grande poder computacional.

Também, podemos destacar, que a manipulação de arquivos textuais contendo muitas informações podem resultar em sobrecarga no SGBD (Sistema de Gerenciamento de Banco de Dados), devido ao tempo gasto para percorrer todos os dados armazenados (ABITEBOUL *et al.*, 2001).

Sabe-se que a análise de sistemas para a manipulação de documentos em XML já fora comentada na literatura por MELLO (2003), porém, verifica-se que existem poucas discussões sobre sua aplicação e resultados obtidos para este tipo de sistema.

1.2 Justificativa

Na PMMG fatos, direitos e deveres dos policiais militares, são registrados e publicados em documentos periódicos no formato de documentos impressos e digitais. Isto, nos permitiu chegar a conclusão de que isolados os dados são incapazes de prover informações úteis aos gestores.

Antes da execução deste projeto, a Sexta Região da Polícia Militar de Minas Gerais não dispunha de uma ferramenta informatizada para facilitar no processo de consultas, desta forma o levantamento destes documentos publicados era reali-

zado manualmente, resultando em vários dias de pesquisa para recuperar todos os dados existentes, foi então, que surgiu a necessidade de criar uma ferramenta que informatizasse o processo capaz de armazenar e consultar as publicações e que suprisse alguns requisitos, como a facilidade de utilização e o tempo no retorno das informações.

As estruturas em XML, possuem potencial na organização e compartilhamento de informações segundo PINTO; SACCOL (2003) e podem ser uma boa alternativa para tratar grandes quantidades de dados devido sua portabilidade para diferentes documentos. A portabilidade, citada anteriormente, permite que o sistema funcione sob qualquer plataforma mantendo a camada de *front-end* e implementando as camadas de *back-end* em qualquer linguagem de programação. Assim, caso o sistema seja migrado para outra linguagem, poderá facilmente continuar em execução sem alterar o código fonte.

1.3 Objetivos Geral e Específicos

O principal objetivo desta pesquisa foi facilitar a consulta e armazenamento dos dados através da modelagem e desenvolvimento do Sistema de Gerenciamento de Publicações da Sexta Região da Polícia Militar de Minas Gerais.

Foram os objetivos específicos deste projeto:

1. Organizar os documentos para facilitar o acesso aos dados;
2. Implementar o Sistema Gerenciamento de Publicações;
3. Verificar as vantagens na utilização do XML;
4. Avaliar e testar o sistema.

1.4 Estrutura deste Documento

O restante deste trabalho está estruturado da seguinte forma. No capítulo 2 apresenta-se o referencial teórico sobre o conceito de Gerenciamento de Arquivos, Operações em Arquivos, o XML, HTML e XML, XSL, VBScript, Bootstrap e Trabalhos Relacionados. No capítulo 3, a metodologia do projeto. No capítulo 4, é apresentado o Desenvolvimento. No capítulo 5, os resultados. No capítulo 6, as conclusões e trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo encontra-se alguns conceitos importantes para um bom entendimento do texto, o Gerenciamento de Arquivos, Operações em Arquivos, XML, HTML e XSL, XSL, VBScript, Bootstrap e Trabalhos Relacionados.

2.1 Gerenciamento de Arquivos

Um Sistema Gerenciador de arquivos é utilizado para organizar arquivos e diretórios. A nível de sistema operacional temos o Windows Explorer, o Nautilus, entre outros.

O seu conceito é a criação, remoção e alteração de arquivos, sendo um sistema responsável por identificar cada arquivo por um nome, por exemplo, um livro em uma biblioteca, que é separado por gênero, nome, autor, ano, além de um identificador único para cada livro.

Portanto, sendo comparado a uma estante de livros, no sistema de arquivos, o nome do arquivo é armazenado no disco rígido e, é utilizado para a localização no computador.

Para o desenvolvimento de um Sistema Gerenciador de publicações, é preciso analisar alguns fatores que são oferecidos pelo Gerenciamento de Arquivos do Sistema Operacional e o Gerenciamento de Dados oferecido pelos SGBDs como: o controle de concorrência, controle de erro, controle de redundância, recuperação de falhas, segurança dos dados e controle de integridade.

Sobre o controle de concorrência, deve-se distribuir o processo entre as várias aplicações de modo que todas sejam atendidas. Em resumo, o controle de erro é a capacidade que o sistema tem de se recuperar após um erro evitando perder

informações. Já o controle de redundância, permite a integridade de um processo, ou seja, faz com que o sistema continue em perfeito funcionamento mesmo sobre falhas, ou sobrecargas. Por último, a recuperação de falhas, é a capacidade de um sistema se recuperar após um erro sem perder informações (MACHADO; MAIA, 2013).

É a segurança que protege os dados de acessos não autorizados, garantido que apenas os utilizadores tenham acesso ao sistema de acordo com os privilégios definidos. Desta maneira, a integridade dos dados torna-se a garantia de que todos os dados contidos na base de dados são válidos e não contradizem a realidade.

Para a implementação do sistema foram analisados os fatores importantes no Gerenciamento de Arquivos e levando em consideração que o XML não fornece todos esses fatores, podemos verificar que existem algumas características deste sistema que foram citadas acima e que podem ser implementadas através de ferramentas encontradas no IIS e nos objetos das linguagens utilizadas como o FSO e o DOM.

O IIS (*Internet Information System*), foi criado pela Microsoft e funciona como um servidor web que traz alguns recursos além da segurança como o isolamento do aplicativo, que concede ao processo uma identidade que configura uma área restrita e o isolamento automático que reduz o trabalho do servidor.

O objeto FSO (*File System Object*), permite que a aplicação tenha capacidade de criar, alterar, mover e excluir pastas e também obter informações sobre elas. Isto, possibilita também, o armazenamento de dados em um formato de fácil acesso e eficiente. Ao usar deste objeto, torna-se interessante armazenar dados em arquivos de texto ou em arquivos binários pois, utilizando um SGBD, não é possível acessar os dados rapidamente, o que torna o sistema muito lento (JR, 1998).

O DOM (*Document Object Model*), é um modelo de objetos que representa documentos XML através de uma estrutura hierárquica em árvore, ele disponibi-

liza uma API (*Application Program Interface*) para fornecer métodos para transformar o documento em uma instância do modelo e manipular objetos.

2.1.1 Operações em Arquivos

Em um SGBD operações em arquivos podem ser agrupadas como operações de recuperação e operações de atualização, o primeiro grupo não muda qualquer dado no arquivo, ele localiza registros de modo que seus valores de campo possam ser analisados e processados. O segundo grupo muda o arquivo pela inserção ou exclusão de registros, ou pela modificação dos valores no campo (ELMASRI *et al.*, 2005).

Em um sistema operacional um arquivo é um tipo abstrato de dados e seis operações básicas são definidas sobre ele, chamadas de “sistema” são geradas para criar, escrever, ler, reposicionar, excluir e separar arquivos. O sistema também possui uma tabela, da qual controla a atividade de cada arquivo. Quando uma operação de arquivo é solicitada, o arquivo é especificado através de um índice para esta tabela. Quando o arquivo não é mais usado, ele é fechado pelo processo e o índice é removido da tabela.

2.2 XML

O XML (*Extensible Markup Language*), surgiu a partir da necessidade de uma linguagem flexível que combinasse a flexibilidade do SGML (*Standard Generalized Markup Language*) com a simplicidade do HTML e é uma recomendação da W3C (*World Wide Web Consortium*) (YERGEAU *et al.*, 2004). O principal objetivo era criar uma linguagem que pudesse ser lida por softwares e interagir com diversas linguagens (VILLELA, 2006). De acordo com BRAY *et al.* (1998) sua filosofia seria incorporada por vários princípios como:

1. Interligação de bancos de dados distintos;
2. Concentração na estrutura da informação, e não na sua aparência;
3. Separação do conteúdo da formatação;
4. Simplicidade e legibilidade, tanto para humanos quanto para computadores;
5. Possibilidade de criação de tags sem limitação;
6. Criação de arquivos para validação de estrutura (chamados DTDs).

De fato, o XML apresenta vantagens pela sua organização de dados de forma hierárquica e sua portabilidade, que não depende de plataformas de software e hardware. Ele promove um padrão para codificar o conteúdo, as semânticas e as esquematizações para uma grande variedade de aplicações desde simples até as mais complexas, (MELLO, 2003).

Também segundo ALMEIDA (2002) o XML é uma linguagem muito poderosa na internet e em sistemas Intranets, pois, possibilita a capacidade de operação conjunta dos servidores por ter um padrão flexível e aberto. Assim, as aplicações podem ser construídas e atualizadas mais rapidamente e também permitem múltiplas formas de visualização dos dados estruturados.

A linguagem XML possibilita que a partir do momento em que o dado tenha sido recebido pelo cliente, este dado possa ser manipulado, editado e visualizado sem a necessidade de acionar novamente o servidor. Isto permite aos servidores, ter menor sobrecarga, reduzindo a necessidade de computação e também reduzindo a requisição de banda passante para as comunicações entre cliente e servidor.

De acordo com VILLELA (2006), a dificuldade de se trabalhar com a integração de aplicações distintas diminuiu consideravelmente com a utilização do XML, devido ao seu padrão de substituir arquivos de texto. Por isto, os arquivos de texto

usados antigamente foram substituídos por documentos XML e também, as linguagens de programação como o C, C++, VB e Java adotaram o XML como formato universal, pois esta linguagem consegue extrair informações independentemente da plataforma ou linguagem.

Com relação aos dados, estes, são estruturados na representação de árvore. Os “nós internos” representam os elementos complexos, enquanto os “nós folhas”, representam os elementos simples. Os dados precisam seguir algumas diretrizes sintáticas. Deve haver um único elemento raiz, e cada elemento precisa incluir um par correspondente de tags de início e fim dentro das tags e de início e de fim do elemento pai (ELMASRI *et al.*, 2005).

Ressaltamos que um documento XML bem formatado precisa seguir todas as definições da W3C para o seu desenvolvimento, e um documento XML válido é o documento XML bem formatado que está vinculado a um documento DTD (*Document Type Definition*) ou a um XML Schema (HE; NAUGHTON, 2008).

Através destas colocações, afirma-se que a ideia central da validação de documentos XML é definir blocos de construção legais de um documento XML, ou seja, definir a estrutura de um documento com uma lista de elementos legais. Também cabe ressaltarmos, que o XML Schema tem o mesmo objetivo do DTD, porém, é uma nova tecnologia mais poderosa e que traz mais recursos.

É seguindo a estrutura a partir de um documento XML que torna-se possível a conversão, extração e combinação de dados de documentos XML para gerar outros documentos XML tornando muito simples a manipulação de informações em diferentes sistemas.

2.3 HTML e XML

Uma das características do XML se resume em separar a interface com o usuário dos dados estruturados. O HTML especifica como o documento deve ser apresentado na tela por um navegador. Já o XML, define o conteúdo do documento. Muitos confundem achando que o XML foi criado para substituir o HTML, isto não é verdade. Por exemplo, em HTML são utilizadas tags para definir tamanho, cor e fonte, assim como formatação de parágrafo. No XML, utiliza-se as tags para descrever os dados, como exemplo tags de assunto, título, autor, conteúdo, referências, datas, entre outros (LIE; SAARELA, 1999).

Entretanto, o HTML é mais flexível com relação as tags, permitindo que não feche algumas das tags. Neste ponto, o XML é altamente estruturado. Quando uma tag é criada ela precisa ser fechada, se não, sua estrutura não será validada e o arquivo será inválido.

2.4 XSL

O XSL (*eXtensible Stylesheet Language*) descreve folhas de estilos para o XML, ele é definido por três elementos, o XSLT, o XSL-FO e o XPath. No entanto, o termo XSL pode ter diversos significados.

O XSLT (*eXtensible Stylesheet Language for Transformation*), como o nome diz, é uma linguagem extensível para folhas de estilo de transformações ou seja, o XML usado para criar documentos XSL, é a transformação de documentos XML em outros documentos.

O XSL-FO (*XSL Formatting Objects*) é a formatação e transformação de documentos XML para um outro tipo de documento formatado, geralmente PDF. O seu uso é muito restrito.

O XPath (*XML Path Language*) é uma linguagem de consulta, usada para localizar nós no arquivo XML, ela inclui mais de cem funções prontas para o uso, e, é de extrema importância pois, sem ela não seria possível a criação de documentos XSLT.

Além do XSL, o XML utiliza recursos tais como folhas de estilo definidas como CSS (*Cascading Style Sheets*) para a apresentação de dados em um navegador. O XML separa os dados da apresentação e processo, o que permite visualizar e processar os dados como quiser, utilizando diferentes folhas de estilo e aplicações.

O documento XSL não altera o XML e não cria outro documento. O XSLT funciona como uma folha de estilo CSS, mostrando ao browser como ele deve apresentar o documento XML. O XSLT além de definir o estilo pode adicionar conteúdo a um documento XML e também esconder o conteúdo a ser apresentado no browser (CLARK *et al.*, 1999).

Assim, o XSLT transforma documentos XML com o intuito de melhorar sua portabilidade, ele corrige o problema de duas aplicações que precisavam se integrar e ocorria problemas de validação devido a problemas de estruturas em um dos documentos MØLLER; OLESEN; SCHWARTZBACH (2007). Qualquer documento XML pode gerar outro documento XML sem problemas de integração, desde que a conversão seja feita por uma folha de estilo XSLT.

Conclui-se que a combinação das linguagens citadas anteriormente, torna o XML muito eficiente, sendo capaz de realizar interações com diferentes sistemas, possibilitando a troca de informações. É importante destacar também, que no comércio eletrônico a transformação de documentos XML em outros documentos XML ocorre frequentemente e de acordo com KAY (2001) a necessidade destas transformações servem para a adequação estrutural do documento.

2.5 VBScript

O VBScript (*Microsoft Visual Basic Scripting Edition*), é uma linguagem de script, de programação leve e utiliza o COM (*Component Object Model*) para acessar os elementos do ambiente no qual ele está sendo executado para ler, atualizar e excluir arquivos (READ *et al.*, 2007).

Por padrão o VBScript é instalado em todas as versões do Windows desde o Windows 98 e é a linguagem padrão no ASP. No portal da PMMG, é utilizado o Microsoft Windows Server, portanto, o VBScript é usado como mecanismo para o ASP (*Active Server Pages*), que é uma estrutura de bibliotecas para o processamento de linguagens de script no lado do servidor para a geração de conteúdo dinâmico na Web. O ASP suporta algumas linguagens como o VBScript, JScript, Python, entre outros (MORENO, 1997).

O script é interpretado no lado do servidor e é enviado para o usuário como uma saída que é uma linguagem de marcação como HTML, XML ou XHTML.

O ASP para percorrer um arquivo, utiliza o FSO (*File System Object*), assim, é possível retornar informações sobre um arquivo especificado. O objeto é instanciado por um método que retorna as propriedades do arquivo.

O FSO possui grandes utilidades e pode poupar muito trabalho, como ler e escrever arquivos comuns além de listar, excluir e deletar (SANDVIG, 2004).

Para a manipulação de arquivos em XML, temos o XML DOM que define uma maneira padrão para acessar e manipular documentos XML. Utilizando o DOM, o programador pode criar um documento, percorrer sua estrutura, adicionar, deletar e modificá-lo.

2.6 Bootstrap

O Bootstrap é um framework desenvolvido por Mark Otto e Jacob Thornton no Twitter OTTO; THORNTON (2010), cujo o objetivo é facilitar na criação de um website. Antes do Bootstrap, diversas bibliotecas foram utilizadas para o desenvolvimento da interface, o que pode levar a inconsistência dos dados e um custo muito elevado para a manutenção.

Ao construir um novo projeto não é necessário começar do zero, o framework dá suporte ao HTML/CSS e traz toda uma base construída de onde a aplicação pode ser iniciada. Alguns dos recursos oferecidos por esta ferramenta são: Reset CSS, Ícones, Plugins JavaScript, componentes CSS e estilo visual para as tags.

O objetivo principal do Bootstrap é reduzir o tempo gasto no desenvolvimento de um website, seja ele uma página simples estática ou um grande portal dinâmico.

2.7 Trabalhos Relacionados

O uso do XML é cada vez mais intensivo e tem despertado o interesse pelo seu estudo em diversas áreas da ciência da computação. Existem várias utilizações para o XML, sobretudo para representar a troca de dados.

Observa-se que o uso desta linguagem é mais abrangente em aplicações que utilizam a Web, pois, é considerado um padrão para a transferência de dados entre diferentes domínios na Web, além de possuir um formato irregular que combina texto com informações estruturadas.

Nos artigos PINTO; SACCOL (2003); SALGADO; LÓSCIO (2001) o XML é estudado e discutido sobre suas vantagens comparado a SGBDs e BDs (Banco de Dados) em XML nativo em relação ao armazenamento e acesso aos dados.

Em MELLO (2003) é discutido o porquê investir em BDs com XML nativo se já existem tecnologias para SGBDs que oferecem suporte ao XML. Alguns argumentos foram apresentados e justificam essa opção, como: dados XML apresentam estruturas muito irregulares, mesmo com um BD relacional com suporte a XML, é necessário utilizar linguagens de manipulação que não são adequadas ao XML, a flexibilidade do XML que suporta mudanças no projeto e a necessidade de processamento em tempo real. Outro fator importante, é o alto investimento com hardware, software e treinamento para investir em um SGBD.

Os artigos estudados foram de extrema relevância para apresentarmos uma solução e implementarmos um sistema de gerenciamento de arquivos com base em XML.

3 METODOLOGIA

A metodologia de pesquisa é um conjunto de métodos, técnicas e procedimentos cuja finalidade é viabilizar a execução da pesquisa que tem como resultado um novo produto, processo ou conhecimento (JUNG; ENG, 2009).

Pode-se classificar a pesquisa, quanto ao tipo, de acordo com a Figura 3.1 abaixo.

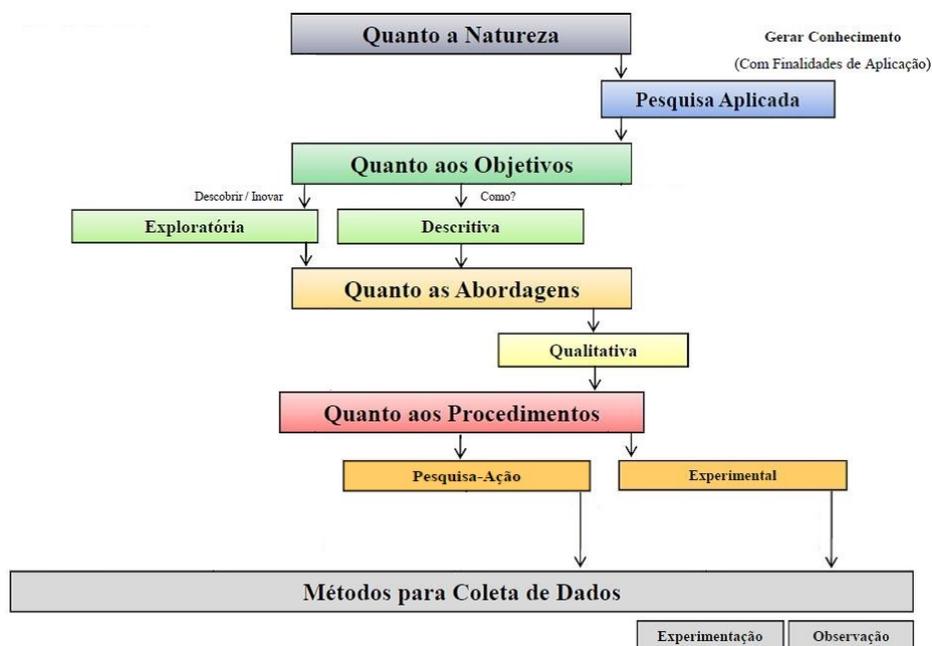


Figura 3.1: Metodologia do projeto. (Fonte: Adaptada de (JUNG; ENG, 2009))

Quanto à natureza, este trabalho pode ser classificado como uma **pesquisa aplicada**, pois, o problema tratado possui finalidade de aplicação. Quanto aos objetivos, pode ser caracterizado como uma **pesquisa exploratória**, pois testa a implementação de um sistema de gerenciamento de arquivos em XML. Além disso, pode ser caracterizado como um trabalho com objetivos de **pesquisa descritiva**, visto que a estratégia utilizada em XML é explicada. Quanto à sua abordagem,

é **qualitativa**, pois visa buscar novas alternativas em relação ao uso de SGBDs, levando em consideração os seguintes fatores: controle de concorrência, integridade e redundância dos dados, além de controle de erros e segurança da informação. Quanto aos procedimentos, este trabalho pode ser caracterizado como uma **pesquisa-ação**, pois é apresentado um estudo de um novo sistema, que propõe a resolução de um problema e também **experimental** pois é proposto uma nova alternativa sem utilizar um SGBD. A coleta de dados é por meio da **observação** do código fonte, visto que, um domínio foi modelado com a observação de seus requisitos necessários em diferentes tecnologias e também por meio de **experimentação** pois será realizado diversos testes para observar o comportamento do sistema.

Inicialmente, durante a definição do trabalho, verificou-se que na 6ª RPM eram gerados muitos documentos diariamente, e não existia uma ferramenta informatizada para auxiliar no cadastramento e consulta destes dados.

Foi feito um levantamento de requisitos das necessidades dos usuários dentro do sistema e das necessidades que o sistema deveria suprir, com essas informações foi possível traçar os dados relevantes para o armazenamento e iniciar a construção dos formulários em XML.

Os dados armazenados possuem muitas informações e a necessidade pela padronização dos dados, foram os motivos da opção na utilização do XML. O sistema foi desenvolvido tratando os problemas de controle de acesso, segurança, controle de concorrência, controle de erros, entre outros.

Desta forma foi possível a construção do sistema e uma interface para testes utilizando páginas HTML com o framework Bootstrap, o sistema foi testado com varias entradas, incluído base de dados vazias, busca por dados inexistentes, grandes quantidades de dados e foi observado o seu comportamento.

4 DESENVOLVIMENTO

Neste capítulo, é descrito uma visão geral do que foi feito. Posteriormente, é detalhado sobre a construção, as funções e por ultimo, sobre a implementação do sistema em questão.

4.1 Visão Geral do Sistema

Implementou-se um protótipo de um Sistema Gerenciador de Publicações para adicionar, consultar, alterar e remover as publicações, de forma simples e organizada.

Na Figura 4.1 é apresentado o diagrama de casos de uso, que é um mapeamento superficial do sistema que mostra de modo geral, as necessidades do usuário dentro do sistema.

O diagrama de casos de uso da Figura 4.2, foi elaborado para ressaltar as necessidades que o sistema deve suprir para atender as demandas e alcançar os objetivos propostos.

A partir do diagrama das necessidades do sistema, foi feito um levantamento do que seria relevante nos documentos para o armazenamento dos dados, entre eles: o nome do Policial, o número do Policial, o número da Publicação, a data da Publicação, o tipo de Publicação, a Unidade, a data em que a Publicação se tornou pública e o Assunto. Desta forma, com base nestas informações o formulário de cadastramento foi desenvolvido.

O campo Unidade está dividido em 4 opções, são eles: 6ª Região de Polícia Militar (6ª RPM), 8ª Batalhão (8ª BPM), 6ª Companhia Independente do Meio Ambiente (6ª Cia Ind Mat) e 6ª Centro de Apoio Administrativo (CAA/6). E

o campo tipo de Publicação está dividido em 6 opções de Boletins Informativos como: BI, BIR, BIE, BGPM, BGPMR, BEPM. Os outros dados devem ser preenchidos de acordo com os campos solicitados.

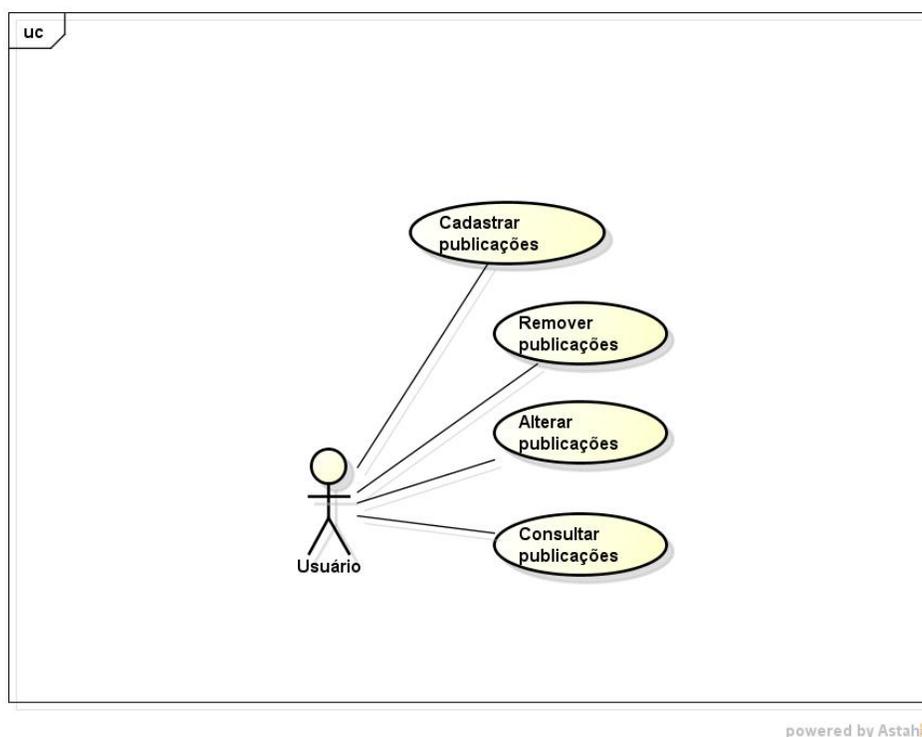


Figura 4.1: Diagrama de casos de uso do usuário do Sistema de Gerenciamento de Publicações.

O objetivo do sistema é o gerenciamento de publicações dos usuários, acerca dos atos públicos gerados. Entretanto, por se tratar de uma autarquia estadual, há a necessidade de controlar o acesso às informações. Com base nessa restrição, o controle de permissões foi implementado.

A ideia principal desse controle é associar uma permissão para acesso ao sistema através de informações do banco de dados. Como este não foi o foco do trabalho, o controle foi implementado utilizando uma estrutura de SGBD, já em funcionamento no portal corporativo.

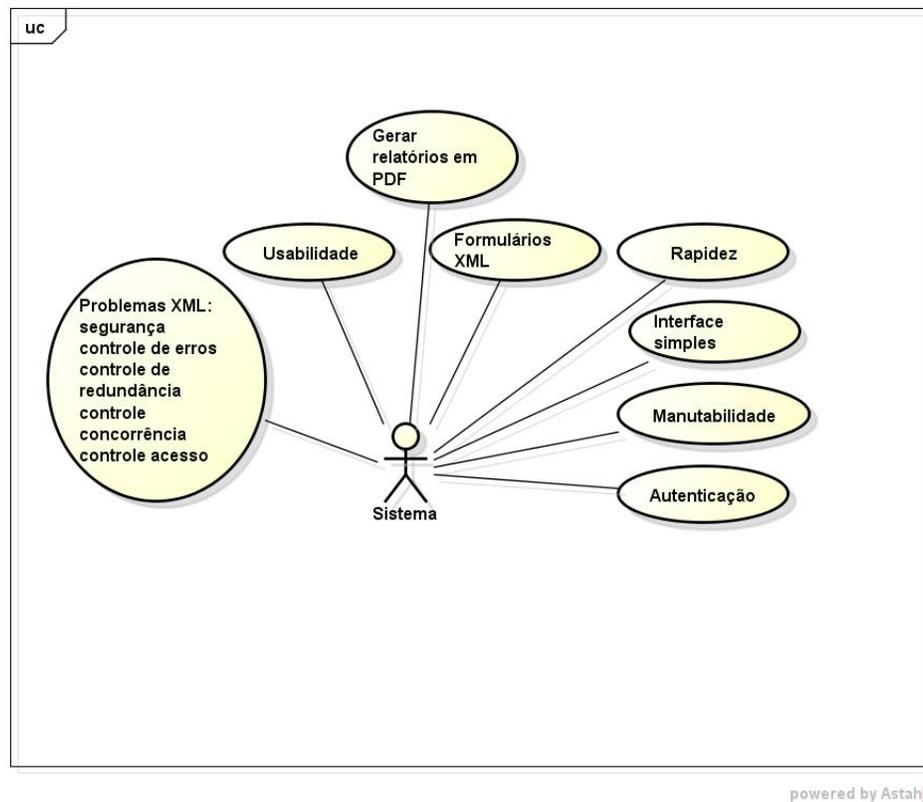


Figura 4.2: Diagrama de casos de uso do Sistema de Gerenciamento de Publicações.

A usabilidade do sistema é muito simples, na interface principal existem apenas três opções listadas em uma coluna à esquerda, que permitem ao usuário cadastrar, consultar, alterar e remover publicações.

O sistema exige que o usuário preencha corretamente os campos e se atente a erros antes de salvar definitivamente os dados. Uma mensagem confirma se os dados foram armazenados com sucesso, caso ocorra algum erro, o usuário poderá consultá-los na guia “consultar” ou, corrigi-los na guia “alterar”.

Com o propósito de melhorar o sistema, mensagens de erros e tratamentos de exceções foram inseridos. A segurança é feita pelo Sistema Operacional, que

gerencia as pastas e arquivos onde é definido os níveis de permissões, e através do browser.

Quando o usuário acessa o portal, toda a comunicação de entrada e saída dos dados passa pelo servidor web IIS, que gerencia os dados. Desta forma, é feito o isolamento da aplicação, ou seja, os processos possuem uma identidade exclusiva e uma configuração de área restrita. O usuário externo não tem acesso aos dados, pois, estão localizados fora da área de aplicação.

O controle de concorrência é o problema que ocorre quando vários usuários utilizam o sistema ao mesmo tempo, podendo ocorrer conflito entre operações simultâneas. Este problema foi verificado que não ocorre pois a API DOM possui protocolos para tratar a concorrência, um deles é o *Lock and Unlock*, o qual previne a modificação em um arquivo enquanto ele estiver sendo usado por algum usuário.

4.2 Interface Sistema



Figura 4.3: Interface geral do Sistema de Gerenciamento de Publicações.

Na Figura 4.3 é apresentado a tela inicial do sistema. Essa interface foi elaborada para a realização dos testes e contém um menu que permite o acesso as

funções do sistema. No primeiro campo, o usuário tem a opção de Cadastrar Publicação, no segundo campo, tem a opção de Consultar Publicação, e no terceiro campo, tem a opção de Alterar ou Remover Publicação.

Na Figura 4.4 podemos ver a tela para “Cadastrar Publicações”. Nela o usuário poderá cadastrar uma nova publicação, para isso, todos os campos devem ser obrigatoriamente preenchidos, ao contrário, não é possível cadastrar. Após o cadastro, uma mensagem confirma se os dados foram corretamente armazenados.

The screenshot shows a web interface for 'Ficha de Cadastro de Requisição de Serviços'. On the left, there is a 'Menu de Publicação' with options: 'Cadastrar', 'Consultar', and 'Alterar/Remover'. The main form area contains the following fields and instructions:

- Header: 'Ficha de Cadastro de Requisição de Serviços' and 'Contém informações e dados para inclusão de requisição de serviço no sistema.'
- Instruction: 'Preencha os dados e clique em salvar.'
- Fields: 'Unidade:' (dropdown), 'Dia:' (input), 'Mes:' (dropdown), 'Ano BI:' (input), 'Tipo de Publicação:' (dropdown), 'Data Publicação:' (input), 'Nr. BI:' (input), 'Nr. PM:' (input, highlighted with a red border), and 'Posto/Grad./Nome do Militar:' (input with example 'Ex.: Sd PM João da Silva').
- Field: 'Assunto:' (text area).
- Button: 'Salvar' (blue button at the bottom right).

Figura 4.4: Interface para cadastrar do Sistema de Gerenciamento de Publicações.

Na Figura 4.5, temos a tela de “Consultar Publicação”. Nesta tela, o usuário poderá pesquisar por publicações que foram armazenadas no sistema, para isto existem três formas de consultas, que podem ser o número do PM¹ ou, o número do BI² ou mesmos os dois campos juntos devem ser informados, e a pesquisa retorna todas as publicações referentes aos dados informados. Nesta opção serão listadas todas as entradas referentes aos dados e o usuário poderá gerar um arquivo em PDF com todos os dados referentes à pesquisa de acordo com a Figura 4.6.

¹Número referente a matrícula do Policial Militar.

²Número referente a ficha funcional do Policial Militar.

Figura 4.5: Interface para consultar do Sistema de Gerenciamento de Publicações.

Figura 4.6: Resultado da consulta do Sistema de Gerenciamento de Publicações.

Na Figura 4.7 é apresentado a tela de “Alterar e Remover Publicação”. Nela o usuário poderá pesquisar por publicações que foram armazenadas no sistema da mesma forma que no menu “Consultar”, porém, nesta tela devido ao grande

número de informações vinculadas a um policial militar ou a um BI, devem ser informados o número do PM e o número do BI.

Figura 4.7: Interface para alteração e remoção do Sistema de Gerenciamento de Publicações.

Após entrar com os dados, a tela da Figura 4.8 será gerada permitindo ao usuário alterar ou remover.

Figura 4.8: Resultado para alteração e remoção do Sistema de Gerenciamento de Publicações.

Para alterar os dados será gerada a seguinte tela da Figura 4.9, da qual é permitida a alteração e após alterar é necessário salvar, com isso, uma nova mensagem

informa ao usuário que os dados foram alterados com sucesso. A ação remover apenas informa que os dados foram removidos com sucesso.

Figura 4.9: Interface para alteração do Sistema de Gerenciamento de Publicações.

Para todas as ações no sistema uma mensagem confirma se foi bem sucedida essas ações e o usuário poderá sempre consultar se os dados foram inseridos com sucesso.

O funcionamento de todo o sistema descrito nesta seção pode ser facilmente visualizado na Figura 4.10, em que é apresentado o Diagrama do Sistema, ou seja, um resumo simples do sistema. As publicações são cadastradas no sistema pelos usuários e podem ser alteradas, removidas e consultadas.

O arquivo XML gerado utiliza a sintaxe da Figura 4.11, os dados são armazenados seguindo a estrutura de uma árvore estruturada com a respectiva tag introdutória. O formato do arquivo é simples, porém, ele é auto documentado, o próprio formato descreve a sua estrutura e nomes de campos, assim como valores válidos (BOSAK; BRAY *et al.*, 1999).

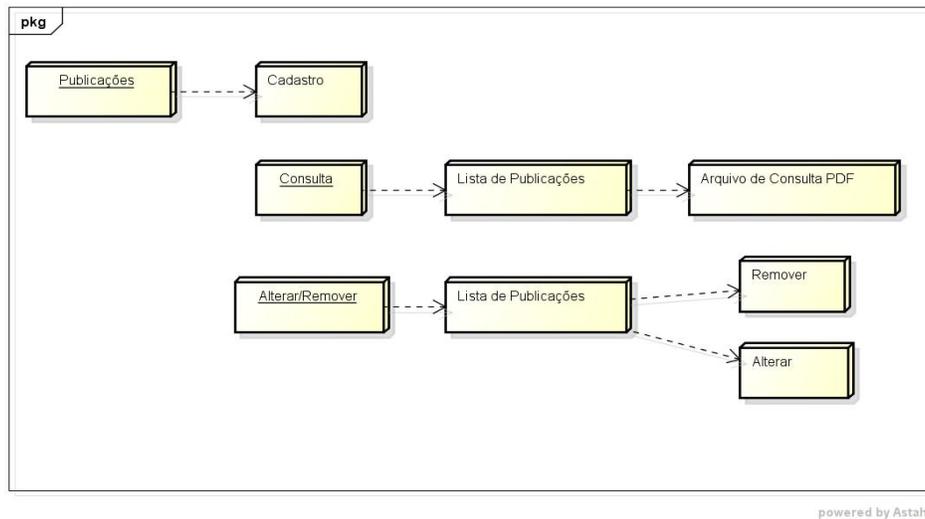


Figura 4.10: Diagrama do Sistema de Gerenciamento de Publicações.

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <controlePub>
3   <numPM id="4444444">
4     <ano>2010</ano>
5     <mes>Maio</mes>
6     <nomePM>Sd Joao Silva</nomePM>
7     <unidPub>6 Cia Ind MAT</unidPub>
8     <tipoPub>BGPM</tipoPub>
9     <dataPub>21/10/2014</dataPub>
10    <diaPub>12</diaPub>
11    <numPub>212122</numPub>
12    <assuntoPub>Lorem ipsum dolor sit amet
13      consectetuer adipiscing elit. Etiam eget
14      ligula eu lectus lobortis condimentum</assuntoPub>
15  </numPM>
16 </controlePub>
  
```

Figura 4.11: Modelo de dados para o documento XML.

4.3 Detalhes de Implementação

Durante a implementação foi construído um diagrama, apresentado na Figura 4.12, com todas as ferramentas que foram utilizadas para se entender melhor o funcionamento do sistema e entender todo o fluxo de informações dentro do sistema.

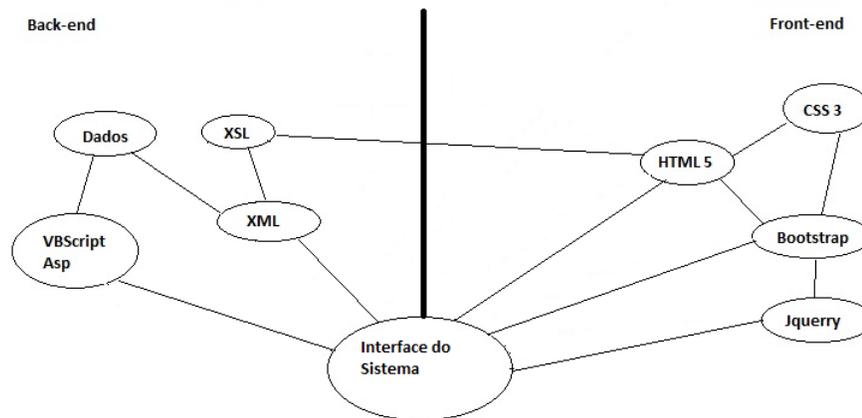


Figura 4.12: Diagrama do fluxo de Informações do Sistema de Gerenciamento de Publicações.

Neste diagrama, o usuário se comunica com o sistema através do seu layout que responde através do HTML e utiliza as estruturas como Bootstrap, jQuery e CSS, os formulários são preenchidos em linguagem corrente, por intermédio do HTML com XML e ASP/VScript que a partir do XSL atualizam os dados e retornam no HTML, e assim a cada solicitação dos usuários.

O sistema foi construído separando os códigos de *back-end* e *front-end*, que são camadas lógicas divididas em três, são elas: apresentação, negócio e acesso a dados.

A camada de apresentação é o *front-end* onde é exibido e coletado as informações dos usuários. As camadas de negócio e acesso a dados são de *back-end* em que é definido a lógica da aplicação e a comunicação entre as transações, respectivamente.

A programação em camadas traz algumas vantagens como a flexibilidade e organização do código e como citado anteriormente o XML dá suporte a várias linguagens, facilitando assim a manutenção, migração e integração do sistema com o portal.

Este sistema não utiliza um BD tradicional para armazenar os dados, então alguns cuidados foram necessários durante a implementação, em relação ao controle de concorrência, redundância, segurança e integridade dos dados. Estes são alguns fatores que podem comprometer toda a viabilidade do sistema, e cuidados foram tomados durante a implementação para produzir um bom sistema que funcionasse de maneira correta.

O XSL é uma linguagem poderosa que traz vários recursos. Durante a implementação, várias técnicas foram pesquisadas para obter um bom resultado. O XML foi responsável pelo armazenamento dos dados de forma correta, as funções na linguagem ASP foram responsáveis por processar os dados para os arquivos XSL para posteriormente salva-los no arquivo XML.

A linguagem ASP é uma linguagem de geração de páginas dinâmicas na web e foi a primeira linguagem utilizada pela Microsoft para processar dados ao lado do servidor e foi mantida por alguns motivos: primeiro atende a todos os requisitos do sistema e o IIS e FSO dão suporte a ela, o servidor utiliza o Windows Server e a experiência com a linguagem.

Para garantir a integridade dos dados um script de backup foi implementado e periodicamente faz o backup dos dados, através do agendador de tarefas do Windows, e salva este arquivo na pasta especificada.

Após testar todas as funções, foi introduzido o layout definitivo da página com o *framework* Bootstrap e alguns detalhes finais de implementação para garantir a integração do sistema com a estrutura do portal, portanto na Figura 4.13 é apresentado a versão final do sistema.

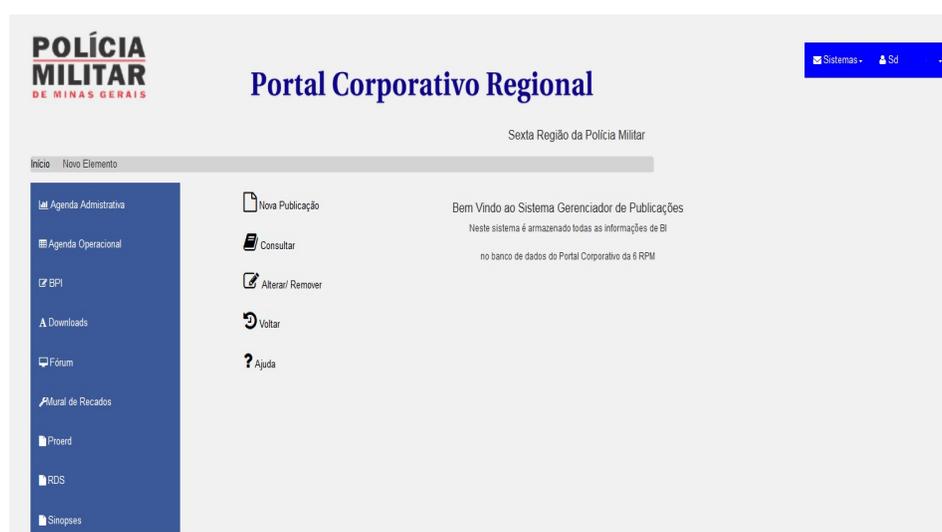


Figura 4.13: Interface final do Sistema de Gerenciamento de Publicações.

5 RESULTADOS E EXPERIMENTOS

Neste capítulo, foram desenvolvidas várias sequências de testes para observar o comportamento do sistema sobre diferentes situações e avaliá-lo.

5.1 Experimentos

Durante a fase de testes foi utilizado o teste de caixa-preta para fornecer informações sobre a qualidade do sistema e força-lo a encontrar possíveis erros. O teste de caixa-preta é um teste funcional que avalia se o sistema está em conformidade com os objetivos propostos e avaliar o funcionamento em relação a entrada e saída de dados (PRESSMAN, 2014).

Cada tipo de teste possui diversas técnicas, que visam assegurar o funcionamento de alguns aspectos do sistema. Este tipo de teste tem o objetivo de verificar o sistema do ponto de vista do usuário, que está interessado em utilizar o sistema sem se preocupar com os detalhes da sua construção.

Os testes foram elaborados com base em informações duplicadas, grandes quantidades de dados, geração de relatórios, base de dados vazias e busca por registros inexistentes.

No conjunto de testes inicial, 20 entradas foram criadas, essas entradas possuem dados que tentam provocar erros no sistema, como entradas duplicadas e números iguais. Estes testes foram executados em uma sequência, como podemos observar a seguir.

Primeiramente, 4 números de PM foram criados, com números de BIs idênticos de acordo com o Quadro 5.1, esses dados foram cadastrados, consultados, alterados e removidos com sucesso.

Número BI	Número PM	Nome PM
21	1111111	Sd Joao
21	2222222	Sd Maria
21	3333333	Sd José
21	4444444	Sd Silva

Quadro 5.1: Conjunto de dados utilizados nos testes.

Continuamos os testes introduzindo as entradas do Quadro 5.2 com números de BIs diferentes para os mesmos usuários e novamente as ações do sistema foram executadas com sucesso.

Número BI	Número PM	Nome PM
103	1111111	Sd Joao
104	2222222	Sd Maria
105	3333333	Sd José
106	4444444	Sd Silva

Quadro 5.2: Conjunto de dados utilizados nos testes.

Novos números de PM representados no Quadro 5.3, foram criados e testados com sucesso, para finalizar os primeiros testes entradas repetidas com mesmo número de BI e mesmo número de PM também foram criadas e testadas, e o sistema se comportou corretamente, sem nenhum erro ou falha que pudesse comprometer os dados.

Número BI	Número PM	Nome PM
32	2222211	Sd Joao
40	4444422	Sd Silva

Quadro 5.3: Conjunto de dados utilizados nos testes.

Os 20 dados criados possuíam apenas 6 usuários ou seja 6 números de PM diferentes, destes usuários foram criados vários registros iguais com o mesmo número de publicação e outras entradas com números de publicações diferentes.

Após os testes iniciais uma estrutura de inserção de dados foi construída, a amostra utilizada continha 500, 1000, 2000, 4000 e 8000 dados e um contador foi implementado para calcular o tempo de execução para inserção, consulta e remoção.

Os resultados observados estão disponíveis no Quadro 5.4, os tempos foram medidos utilizando a unidade segundos, e a medida que o arquivo aumenta o número de registros o tempo tende a aumentar, pois é necessário abrir o arquivo e percorrer toda a sua estrutura. Os intervalos de tempo para remoção e consulta são bem pequenos, pois, uma vez que o arquivo é carregado ele é colocado na memória, a inserção necessita recarregar o arquivo a cada item inserido.

Número de Publicações	500	1000	2000	4000	8000
Cadastro	0,03s	0,07s	0,15s	0,21s	0,30s
Consulta	0,02s	0,06s	0,13s	0,18s	0,29s
Remoção	0,02s	0,03s	0,05s	0,11s	0,15s
Tamanho do Arquivo	286kb	571kb	1132kb	2292kb	4573kb

Quadro 5.4: Resultados dos testes.

Estes dados demonstram a simplicidade e rapidez do sistema para cada operação. Os arquivos gerados são muito pequenos, comparados a SGBDs pois não possuem toda a carga de dados, os dados são armazenados em texto puro.

Uma segunda estrutura foi construída para simular a inserção de dados por dois usuários simultaneamente e os dados foram inseridos com sucesso. O processo gerado para inserção começou inserindo o pedido do primeiro usuário. Assim que o segundo usuário fizer uma nova solicitação, será gerado um novo pro-

cesso, porém, este ficará aguardando o término do primeiro, e somente quando o primeiro for concluído, que o segundo será iniciado. Se o tempo de resposta configurado no servidor for ultrapassado, uma mensagem é enviada informando que o processo está sendo usada por outro usuário.

5.2 Resultados

O teste de caixa-preta verificou a consistência dos dados para todas as entradas de dados geradas e foi observado o comportamento do sistema.

Para todos os testes realizados o sistema se comportou com sucesso, sem nenhum erro ou entrada duplicada que poderiam retornar resultados errados e comprometer a integridade dos dados.

Em todos os testes feitos, os dados foram consultados, alterados e removidos com sucesso, o que demonstrou um bom comportamento para o sistema.

Portanto, o sistema alcançou o resultado esperado para o problema do gerenciamento de publicações, que buscou facilitar no processo de busca e armazenamento de dados, empregando técnicas de implementação na linguagem XML.

6 CONCLUSÕES

O sistema implementado funcionou como o esperado, o que demonstra o grande potencial do XML para tratar este tipo de problema, e, como a linguagem XSL é completa. É importante ressaltar, que arquivos de texto não são uma boa opção para este tipo de sistema, pois, não são estruturados e não possuem ferramentas para auxiliar na manipulação dos dados.

O arquivo final onde os registros do sistema são armazenados gerou um arquivo muito leve comparado a SGBDs, que pode ser importado para qualquer sistema e não existe toda a carga de dados que os SGBDs geram, por exemplo, um arquivo com 500 registros como o que foi usado nos testes possuem aproximadamente 286kbytes.

Portanto, o XML pode ser usado como uma variável para SGBDs, sua implementação tem que ser bem elaborada, pois os SGBDs oferecem uma grande quantidade de vantagens e não existem soluções para todos os aspectos do gerenciamento de dados, pois, são sistemas projetados e pensados para tratar problemas de redundância de dados, segurança, tolerância a falhas, controle de erros, controle de integridade, controle de concorrência, gerenciamento de transações, indexação, entre outros.

Entretanto, dependendo da aplicação, o XML pode apresentar melhores resultados devido a sua simplicidade, que foi o caso demonstrado neste trabalho, do qual arquivos de texto muito grandes poderiam acarretar sobrecarga para o SGBD comprometendo o sistema. De acordo com MLYNKOVA; TOMAN; POKORNÝ (2006) em aplicações que exigem dados em tempo real, os SGBDs podem gerar grandes sobrecargas.

Há algumas aplicações que não possuem regras de negócios bem definidas, neste caso, é interessante criar uma estrutura de armazenamento personalizado

utilizando documentos XML com formatações mais flexíveis. Os dados XML possuem estruturas muito irregulares o que torna o armazenamento complexo em um BD relacional, que pode não conseguir recuperar toda a estrutura física do documento, o que acaba impossibilitando a realização de transações sobre ele.

De acordo com ELMASRI (2008) o SGBD necessita ser bem projetado e administrado, pois, se nenhum dos dois ocorrer, tanto a segurança quanto a integridade do sistema podem ser comprometidos.

Cada aplicação precisa ser avaliada para buscar a melhor ferramenta para atender suas necessidades, ambos possuem suas vantagens. Neste estudo, foi comprovado que para aplicações simples sem um grande investimento em um projeto de BD, o XML provou ser uma boa opção.

Desta maneira, os objetivos do trabalho foram alcançados e as informações agora podem ser rapidamente armazenadas, facilitando a consulta e a geração de relatórios, tornando o trabalho dos usuários mais simples, e auxiliando no controle de informações.

6.1 Trabalhos Futuros

Trabalhos futuros podem ser realizados na importação automática de publicações que estão em arquivos PDF para o sistema criado, de forma que todos os dados em PDF possam ser rapidamente inseridos e armazenados no sistema.

Outro trabalho futuro pode ser feito na elaboração de uma estrutura para replicação dos dados, sincronizando-os em outro arquivo garantindo a integridade.

REFERÊNCIAS BIBLIOGRÁFICAS

ABITEBOUL, S.; AGUILERA, V.; AILLERET, S.; AMANN, B.; ARAMBARRI, F.; CLUET, S.; COBENA, G.; CORONA, G.; FERRAN, G.; GALLAND, A. *et al.* Xyleme, a dynamic warehouse for xml data of the web. p. 3–8, 2001.

ALMEIDA, M. B. Uma introdução ao xml, sua utilização na internet e alguns conceitos complementares. *Ciência da Informação, Brasília, SciELO Brasil*, v. 31, n. 2, p. 5–13, 2002.

BOSAK, J.; BRAY, T. *et al.* Xml and the second-generation web. *Scientific American*, v. 280, n. 5, p. 89–93, 1999.

BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E.; YERGEAU, F. Extensible markup language (xml). *World Wide Web Consortium Recommendation REC-xml-19980210*. <http://www.w3.org/TR/1998/REC-xml-19980210>, 1998.

CLARK, J. *et al.* Xsl transformations (xslt). *World Wide Web Consortium (W3C)*. URL <http://www.w3.org/TR/xslt>, 1999.

DUARTE, E. N.; SILVA, A. K. A. da; COSTA, S. Q. da. Gestão da informação e do conhecimento: práticas de empresa “excelente em gestão empresarial” extensivas à unidades de informação. *Informação & Sociedade: Estudos*, v. 17, n. 1, 2007.

ELMASRI, R. *Fundamentals of database systems*. 6. ed. Boston, Ma: Addison-Wesley, 2008. 35–110 p. (DBMS, v. 1).

ELMASRI, R.; NAVATHE, S. B.; PINHEIRO, M. G.; CANHETTE, C. C.; MELO, G. C. V.; AMADEU, C. V.; MORAIS, R. de O. Sistemas de banco de dados. *Pearson Addison Wesley*, 2005.

HE, J. S. K. T. G.; NAUGHTON, C. Z. D. D. J. Relational databases for querying xml documents: Limitations and opportunities. 25th International Conference on Very Large Data Bases, p. 302–314, 07-10 2008.

JR, R. C. G. File system and method for file system object customization which automatically invokes procedures in response to accessing an inode. Google Patents, junho, 23 1998. US Patent 5,771,379.

JUNG, C. F.; ENG, M. Metodologia aplicada a projetos de pesquisa: Sistemas de informação e ciência da computação. *Ênfase em pesquisa tecnológica*, v. 3, 2009.

KAY, M. (Ed.). *XSLT programmer's reference*. 2. ed. USA: Peer Information, 2001. (Programmer to Programmer, v. 3).

KROTH, E.; NEUMANN, L. R. Benchmark entre técnicas de aumento de desempenho de tabelas com grande volume de dados em bancos de dados relacionais. *Anais do Salão de Ensino e de Extensão*, p. 365, 2013.

LIE, H. W.; SAARELA, J. Multipurpose web publishing using html, xml, and css. *Communications of the ACM*, ACM, v. 42, n. 10, p. 95–101, 1999.

LUNKES, C. M. *et al.* Xml para criação de relatórios dinâmicos em bancos de dados relacionais. Florianópolis, SC, 2003.

MACHADO, F. B.; MAIA, L. P. *Arquitetura de sistemas operacionais*. Rio De Janeiro: LTC, 2013. 35–92 p.

MELLO, R. dos S. Gerenciamento de dados xml. *EIN-Escola de Informtica Norte da SBC*, p. 15–34, 2003.

MLYNKOVA, I.; TOMAN, K.; POKORNÝ, J. Statistical analysis of real xml data collections. v. 6, p. 20–31, 2006.

MØLLER, A.; OLESEN, M. Ø.; SCHWARTZBACH, M. I. Static validation of xsl transformations. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, ACM, v. 29, n. 4, p. 21, 2007.

MORENO, Ó. G. *VBScript y programación ASP*. Madrid: Anaya Multimedia-Anaya Interactiva, 1997. 45–63 p.

OTTO, M.; THORNTON, J. Bootstrap. *Twitter Bootstrap*, 2010.

PINTO, M. B.; SACCOL, D. de B. Um estudo sobre esquemas para documentos xml. *V Encontro de Estudantes de Informática do Tocantins, ANAIS ENCOINFO/EIN*, p. 211–220, 2003.

PRESSMAN, R. S. *Engenharia de software*. 8. ed. New York NY: McGraw Hill Brasil, 2014. 139–231 p. (Enginerring, v. 3).

READ, D.; KINGSLEY-HUGHES, A.; WILTON, P.; UPDEGRAVE, S.; NELSON, E.; MAHARRY, D. *VBScript programmer's reference*. 3. ed. Newark: Wrox Press Ltd., 2007. 95–120 p.

SALGADO, A. C.; LÓSCIO, B. F. Integração de dados na web. *ESCOLA REGIONAL DE INFORMÁTICA DA SBC–REGIONAL DE SÃO PAULO*, v. 6, p. 157–174, 2001.

SANDVIG, J. C. Active server pages. *The Internet Encyclopedia*, Wiley Online Library, v. 2, 2004.

VILLELA, R. T. N. B. *XSLT: Manipulação de documentos XML*. Belo Horizonte, MG: Fumec, junho 2006.

YERGEAU, F.; BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E. Extensible markup language (xml) 1.0. *W3C Recommendation*, v. 4, 2004.