

MÓDULO RELATOR DO SIEP GERENCIAL: UM *WEBSERVICE* PARA GERAÇÃO DE RELATÓRIOS UTILIZANDO TECNOLOGIA LIVRE

**Renildo Viana AZEVEDO(1); Marcelo Machado CHAMY (2); Viviane Gomes SILVA (3);
Diego Melo CRUZ (4); Gabriel Bezerra de Menezes ARMELIN(5)**

Centro Federal de Educação Tecnológica do Amazonas, Av.: Sete de Setembro, 1975 –Manaus – AM, CEP: 69020-120,
telefone: (92) 3621-6717

e-mail: renildo@cefetam.edu.br

(2) CEFET-AM, e-mail: chamy@cefetam.edu.br

(3) CEFET-AM, e-mail: viviane@cefetam.edu.br

(4) CEFET – AM, e-mail: diegocruz86@hotmail.com

(5) CEFET-AM, e-mail: gabrielbma@gmail.com

RESUMO

Em 2006 foi criado no âmbito da SETEC/MEC o projeto SIEP-Gerencial (Sistema de Informação da Educação Profissional e Tecnológica), proposta que consiste na elaboração de um *software* cujo objetivo é fornecer informações para suprir as necessidades de gerenciamento da Educação Tecnológica no Brasil, a partir de uma base de dados centralizada. O software foi dividido em vários módulos, e participam deste desenvolvimento vários núcleos de pesquisa, formados por professores e alunos de diversos Centros Federais de Educação Tecnológica ou CEFET. O desenvolvimento desta ferramenta pode ser entendido como distribuído, pois cada núcleo participante é localizado em uma cidade distinta do Brasil, e também colaborativo, pois além de haver a comunicação de vários *webservices*, os núcleos de pesquisa precisam manter constante contato para que haja homogeneidade nas ferramentas e padrões no desenvolvimento. Dentre os vários módulos a serem desenvolvidos, encontra-se o módulo Relator, que consiste em um *webservice*, cuja função é gerar relatórios a partir da consulta ao *webservice* Seletor do SIEP. Tais relatórios poderão ser gerados em vários formatos, como, por exemplo, no padrão PDF. Este artigo trata do trabalho de pesquisa que envolve o desenvolvimento do Módulo Relator, cuja execução do desenvolvimento envolve o uso de tecnologia livre e código aberto. A metodologia utilizada neste trabalho possui as seguintes etapas: levantamento de ferramentas livres adotadas no projeto SIEP; modelagem conceitual através da UML; caso de uso descrevendo a implementação do *webservice* do Relator em Java; teste de integração com o módulo Seletor; desenvolvimento de funcionalidades relacionadas à segurança da informação geradas pelo Relator. Ao final desta pesquisa será entregue uma ferramenta para geração de relatórios gerenciais que possa ser integrada a qualquer sistema autorizado a ter acesso às informações do SIEP-Gerencial, contribuindo, desta forma, para a melhoria da qualidade do ensino técnico e tecnológico do País.

Palavras-chave: Módulo Relator, SIEP-Gerencial, Software livre, *Webservice*, Java.

1. INTRODUÇÃO

A importância da informação para a tomada de decisões nas organizações tem impulsionado o desenvolvimento dos sistemas de processamento de informações. Tornando-se assim, mais importante existir recursos específicos para facilitar a manipulação dos dados disponíveis nos bancos de dados e o desenvolvimento de programas aplicativos em uma plataforma livre no qual tenhamos acesso ao código fonte utilizado. Neste sentido, o Núcleo do SIEP-Gerencial do Centro Federal de Educação Tecnológica do Amazonas (CEFET-AM) desenvolveu uma ferramenta capaz de recuperar as informações contidas na base de dados do SIEP utilizando um *webservice* que integra vários componentes, tornando estas informações disponíveis na forma de relatórios para as organizações autorizadas e interessadas em prover de informação suas unidades de ensino.

A criação de sistemas de informação como estratégias de expansão, modernização e melhoria contínua da Educação Profissional e Tecnológica – EPT- (BARATO, 2004) faz-se imprescindível, tanto ao nível das unidades acadêmicas que oferecem essa modalidade de educação, como ao nível das instâncias de regulação

e avaliação, como é o caso da Secretaria de Educação Profissional e Tecnológica do Ministério da Educação – SETEC/MEC.

A SETEC/MEC necessita do desenvolvimento desse tipo de sistemas para subsidiar os processos de planejamento estratégico e operacional, bem como suas rotinas administrativas, acadêmicas e de gestão, estabelecendo os indicadores necessários ao diagnóstico, monitoramento e avaliação da EPT no país (MANFREDI, 2003).

O SIEP - Gerencial tem como objetivos desenvolver, implantar, dar suporte e manutenção ao Módulo Gerencial do Sistema de Informações da Educação Profissional - SIEP Gerencial com código aberto, utilizando tecnologia livre, para municiar a SETEC/MEC de instrumentos e ferramentas que possibilitem o exercício de sua função definidora de políticas e supervisão, estimulando um processo contínuo de avaliação, monitoramento, modernização e transparência da oferta e da expansão da Educação Profissional e Tecnológica no Brasil.

Neste artigo, os trabalhos desenvolvidos no Módulo Relator do SIEP Gerencial serão apresentados da seguinte forma: descrição das tecnologias livre utilizadas, o modelo conceitual e estudo de caso mostrando a implementação do *webservice* Relator em Java e sua integração com o módulo Seletor.

2. FUNDAMENTAÇÃO TEÓRICA

Para integrar o Módulo Relator ao SIEP Gerencial foi necessária ampla pesquisa a respeito de soluções utilizando tecnologia livre que tornasse possível a integração com os outros módulos construídos em paralelo dentro do projeto SIEP Gerencial, e que tivesse portabilidade com a plataforma Java Enterprise Edition – JEE.

Java é uma linguagem madura, idealizada pela Sun Microsystems. As suas principais características são: ser orientada a objetos, portabilidade, robustez, segurança e alta performance (DEITEL, 2007). A orientação a objetos é um paradigma de programação que traz diversas vantagens de projeto e implementação de *software* modularizados, facilitando significativamente quase todas as etapas do ciclo de desenvolvimento. A portabilidade é fundamental devido ao fato de o projeto ter que rodar em uma grande quantidade de ambientes diferentes, envolvendo, inclusive, diferença de sistemas operacionais. No entanto, o Java executa sobre uma “máquina virtual”, a Java Virtual Machine (JVM). Isto torna o Java independente do sistema operacional, precisando apenas de uma máquina virtual apropriada ao sistema operacional em uso.

Existem verificações de código em tempo de compilação e em tempo de execução, um sistema adequado de tratamento de exceções, um modelo de gerenciamento automático de memória, entre outros fatores (LEWIS, 2007). A segurança é implementada no núcleo da plataforma, de maneira a evitar ataques de códigos maliciosos em ambientes isolados ou distribuídos. A alta performance provém de diversos fatores, que vão desde os algoritmos eficientes de execução do interpretador e da liberação automática de memória não utilizada, até a possibilidade de que determinados pontos críticos do código sejam compilados em linguagem nativa (SAVITCH, 2007).

Java é a linguagem de programação mais recente e mais poderosa entre as linguagens populares, e não usufruir os seus recursos é perder a oportunidade de escrever aplicativos melhores, em menos tempo e com custo menor. Felizmente muitos grupos importantes de desenvolvedores de *software* livre perceberam o potencial da linguagem e cuidaram para que o mundo livre não ficasse sem boas ferramentas para esta linguagem. Concluindo, Java oferece excelentes soluções para diversos tipos de aplicações (comerciais, científicas, móveis, industriais, entre outras). Além de ser livre e de programas escritos em Java sejam portáteis, Java foi desenvolvida sob o paradigma de orientação a objetos, o que oferece facilidade para organizar o código para uma futura reutilização dos mesmos em outras aplicações, sem a necessidade de reescrevê-los.

2.1. *Webservice*

Um *webservice* é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Os *webservices* são componentes que permitem às aplicações enviar e receber dados em formato XML, por exemplo.

A tecnologia de *webservice* está baseada na arquitetura orientada a serviços. Os objetos que são manipulados são o serviço e a descrição do serviço. As operações que são executadas pelos atores que fazem uso desses objetos são: publicar, encontrar e conectar [Gottschalk et al. 2002].

O *webservice* é um componente baseado em padrões, bem definido e autônomo, que pode ser acessado via protocolos baseados na Web [BEA et al. 2002]. A composição do *webservice* é um paradigma emergente que possibilita a integração tanto de aplicações internas, como de aplicações que transpõem as fronteiras organizacionais [Wohed et al. 2002]. Desse modo, linguagens e técnicas para composição de *webservice* tem surgido e estão sendo continuamente aprimoradas com novas propostas de diferentes empresas e coalizões.

Conforme apresentado na Figura 1, o *webservice* pode se registrar em um repositório central, onde softwares clientes que necessitam dos serviços oferecidos pelo *webservice* poderão pesquisá-lo. Uma vez localizado, o software cliente receberá uma referência para o serviço encontrado, podendo usufruir dos serviços apenas executando os vários métodos implementados com a ajuda das interfaces públicas. Assim, cada serviço precisa publicar sua interface para que os softwares clientes a utilizem (AYALA et al., 2002).

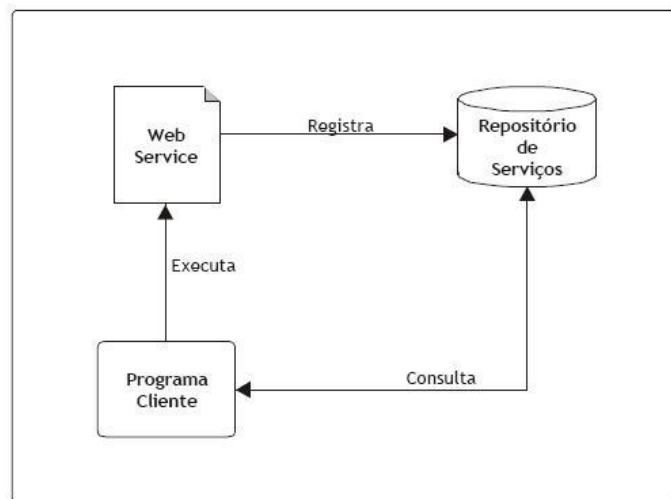


Figura 1: Funcionamento Simplificado de um Webservice. Fonte: Ayala et al. (2002).

Uma grande vantagem na utilização de *webservices* reside no fato de que a equipe de desenvolvimento poder focar seus esforços no sistema em si, praticamente sem se preocupar com o meio de comunicação entre os processos. Especialmente para as grandes corporações – que possuem uma infinidade de soluções concebidas por fornecedores diferentes ou mesmo desenvolvidas internamente nas mais variadas plataformas – esse conceito promete unificar, pela Web, todas as informações contidas nas aplicações, sem que haja necessidade de migração (W3C – a, 2008).

2.2. XML

O XML é uma linguagem derivada da SGML e foi idealizada por Jon Bosak, engenheiro da Sun Microsystems. O autor era conhecedor e usuário da SGML e apresentou ao W3 Consortium sua idéia de explorar o SGML em aplicações voltadas para Internet. Em 1996, foi criado o XML, inicialmente como uma versão simplificada do SGML, e, em fevereiro de 1998, o XML tornou-se uma especificação formal, reconhecida pelo W3 Consortium.

Para Anderson et al. (2001), XML é, conforme descrito, anteriormente, uma metalinguagem de marcação criada a partir de SGML. Na verdade, ela é uma forma restrita de SGML. Seus principais objetivos são:

- prover o intercâmbio de documentos por meio da Web de forma independente de sistemas operacionais ou formatos de arquivos;
- suportar uma grande gama de aplicações, permitindo a definição de elementos pelo usuário (ou aplicação) para estruturar o documento;
- facilitar a análise de documentos XML por programas;
- documentos XML devem ser legíveis por humanos;
- economia de tags de marcação não é importante;
- ter uma especificação formal para a marcação de documentos.

Como a XML não possui um conjunto pré-definido de *tags* ou elementos, ela é considerada uma meta-linguagem para descrição de linguagens de marcação. Desta forma, num documento XML, os elementos – aqui também chamados de *tags* – usados para definir o significado dos dados, podem ser definidos livremente de acordo com o domínio dos dados e da aplicação (DAUM, MERTEN, 2002).

Segundo Moulitis e Kirk (2000), a XML não fornece mecanismos para apresentação de dados nem para ligação entre documentos. Tais recursos são disponibilizados por outras especificações de linguagens como, por exemplo, XSL (Extensible Style Language) e Xlink. A primeira é utilizada na descrição de estilos, já a segunda especifica links entre documentos. O suporte que a XML oferece a separação entre conteúdo e apresentação é um aspecto de suma importância na linguagem. Torna-se imperativo ainda acrescentar que XML é uma linguagem simples, possui um conjunto de estruturas de dados ricas, permite a troca e exibição de conteúdo de bases de dados e pode ser utilizada como formato para troca de mensagens na comunicação entre aplicações. Dentre essas, a troca de mensagens na comunicação entre aplicações de empresas oferece um meio de comunicação de baixo custo para aplicações B2B (Business to Business), e essa é uma das áreas que também pode tirar proveito da tecnologia XML, contanto que protocolos seguros sejam usados para assegurar as comunicações (ANDERSON et al., 2001).

XML tornou-se um padrão internacional em 1998. Desde então várias linguagens de marcação seguindo o padrão XML foram criadas e estão sendo padronizadas no âmbito do w3.org. Recentemente, pequenas adaptações a HTML 4.01 para seguir os requisitos de XML, deram origem à linguagem XHTML (RAY, 2001).

2.3. *JasperReports*

Uma das tecnologias utilizada para desenvolvimento do projeto foi o JasperReports, devido esta aplicação ser segura, poderosa, *open source* e flexível para geração de relatórios. A aplicação é executada e acoplada ao visual designer iReport, que permite tirar total vantagem do poder do JasperReports sem necessidade de conhecimento profundo do formato XML nativo do JasperReports.

O JasperReports é uma biblioteca Java que permite definir um relatório e depois executá-lo contra uma fonte de dados. A definição é feita em XML e pode ser editada manualmente. Contudo normalmente é usada a ferramenta iReport que é um editor gráfico para o arquivo XML. O arquivo XML é depois compilado num arquivo com extensão .jasper que é um arquivo .class normal gerado através do código objeto do Java, apenas com a extensão modificada.

Este arquivo é depois executado contra uma fonte de dados, que na maioria dos casos é um banco de dados. Existem várias fontes de dados e a biblioteca oferece mecanismos para que qualquer um crie uma fonte de dados compatível.

A biblioteca pode ser usada para ver o relatório em tela ou gerar um PDF. Esta característica junto com a facilidade de uso e custo gratuito das ferramentas fazem do JasperReports uma biblioteca de geração de relatórios completa.

O design do relatório, incluindo a localização dos campos a serem preenchidos e seus respectivos nomes, para futuro mapeamento, são definidos em um arquivo XML através de tags XML que obedecem a uma estrutura, vocabulário e restrições, declarados em um arquivo DTD (jasperreports.dtd). Usando XML, o designer pode definir textos estáticos, imagens, linhas, formas geométricas, como retângulos e elipses, e suas localizações dentro do relatório. Pode-se, ainda, e principalmente, definir os campos que serão preenchidos dinamicamente a partir de uma base de dados. O arquivo XML precisa ser compilado, gerando um arquivo .jasper, isto é, contendo a versão compilada do código XML. Isto implica na compilação de todas as expressões Java definidas no arquivo XML, sendo realizadas várias verificações em tempo de compilação. (PAIXÃO, 2005)

Diferentes objetos JasperReports são usados para representar as etapas do processo de geração de relatório. O **JasperDesign** representa a definição do relatório em que, a partir do *template* XML, é criado um JasperDesign. O **JasperReport** representa o JasperDesign compilado, em que o processo de compilação verifica o *design* do relatório e compila o *design* em um objeto JasperReport. Por sua vez, o **JasperPrint** representa o relatório gerado, onde é criado um JasperPrint a partir de um JasperReport, contendo o relatório preenchido.

2.4. iReport

O iReport é um construtor visual de relatórios para o JasperReports escrito 100% na linguagem Java. Esta ferramenta permite aos usuários editar visualmente complexos relatórios com gráficos, imagens e sub-relatórios. Os dados para impressão podem ser recuperados através de várias maneiras, incluindo múltiplas conexões JDBC, TableModels, JavaBeans, XML, MDX, EJBQL, Hibernate, entre outros.

Desenvolver projetos de relatório para a biblioteca JasperReports é uma tarefa que exige conhecimento em XML, assim como das regras e *tags* definidas no DTD (JasperReports.dtd) da biblioteca JasperReports que estrutura o documento XML e que representa o projeto do relatório. Segundo Paixão (2005), esse problema constituiu-se inicialmente em um obstáculo para a utilização da biblioteca JasperReports, uma vez que era necessário o conhecimento da linguagem XML e das regras contida no DTD da biblioteca JasperReports (JasperReports.dtd). Para resolver esse problema foram criadas ferramentas GUI (Interface Gráfica com Usuário) para o desenvolvimento do projeto do relatório e suas configurações. Existem várias ferramentas que oferecem uma interface gráfica para o desenvolvimento do projeto de relatório para a biblioteca JasperReports, sendo que o iReport é uma dessas.

Como o iReport tem como finalidade desenvolver projetos de relatórios de uma forma simples e rápida, através de uma interface gráfica, não há a necessidade de se ter nenhum conhecimento em XML e das regras definidas em seu DTD. Desta forma, o usuário pode criar projetos de relatório utilizando a interface gráfica do iReport sem muitos esforços, o que favorece a produtividade durante o desenvolvimento de um *software*. Também é permitida a visualização do relatório e realização de testes antes de utilizá-lo em uma aplicação. O que diferencia o iReport de outras ferramentas, que têm como finalidade oferecer uma interface gráfica para a construção do projeto de relatório para a biblioteca JasperReports, é que o iReport é de fácil utilização e, além de tudo, é gratuito.

3. ESTUDO DE CASO – MÓDULO RELATOR

O Módulo Relator é uma ferramenta desenvolvida cuja finalidade principal é recuperar informações contidas na base de dados do SIEP – Gerencial, solicitando os dados necessários para o módulo Seletor e retornando para o cliente solicitante. O Seletor também é um *webservice* que provê serviços de consultas à base de dados do SIEP-Gerencial, utilizando uma linguagem própria de consulta no formato XML, nomeada de *XQL*, elaborada pelo núcleo de pesquisa do CEFET-MT.

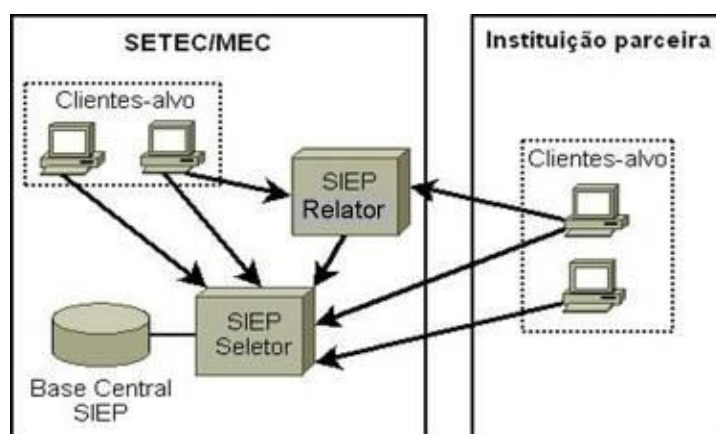


Figura 2: Funcionamento Simplificado do módulo Relator

A figura acima ilustra as etapas para geração de relatórios utilizando o *webservice* Relator: um cliente faz uma requisição ao módulo Relator, utilizando o formato XQL. O relator então envia a requisição para o SELETOR, que faz a comunicação com a base de dados, extrai os dados solicitados e envia um arquivo de resposta no formato XML. A partir do resultado, o módulo Relator realiza a formatação e o dimensionamento dos dados dinamicamente, gera o relatório e o torna disponível para o requisitante autorizado. A figura abaixo ilustra de forma simplificada, o funcionamento do módulo Relator, e como ele se comunica com outros clientes e com o *webservice* Seletor.

A figura abaixo mostra o diagrama de caso de uso com a relação existente entre os módulos Seletor e o módulo Relator. A distribuição das classes neste módulo está dividida através dos funcionamentos e responsabilidades das classes para geração do relatório solicitado.

Caso de Uso - Siep - Módulo Relator

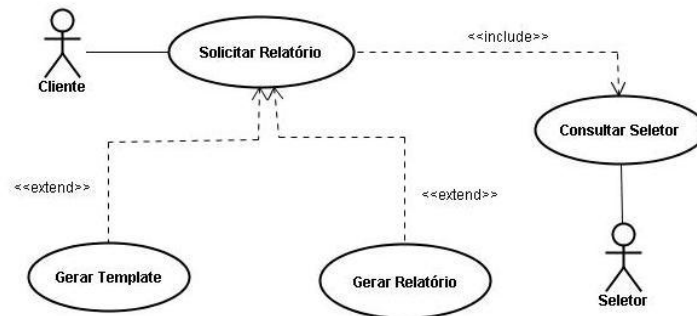


Figura 3 – Diagrama de Caso de Uso

A figura abaixo demonstra as classes que possuem a função de definir a lógica de negócios do Módulo Relator.

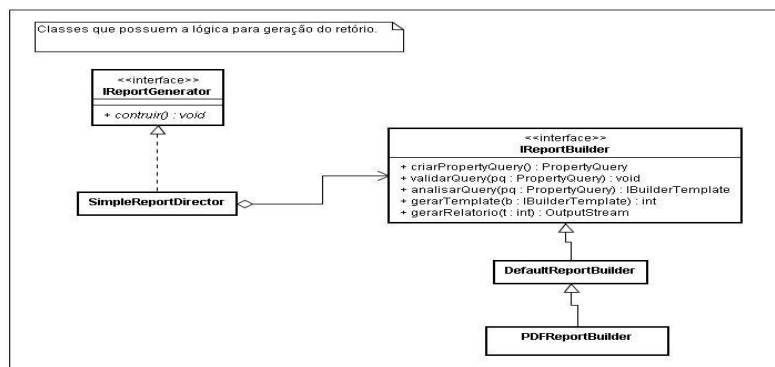


Figura 4 – Diagrama de classes que definem a lógica do módulo

O pacote para lógica de geração de relatório contém classes Java que processam os arquivos de dados a serem mostrados em relatórios, instanciando as classes necessárias do pacote assim como utilizando os arquivos modelos necessários do pacote *Templates*.

A imagem abaixo mostra as classes que possuem a função de definir a estrutura de *Templates*, para o relatório.

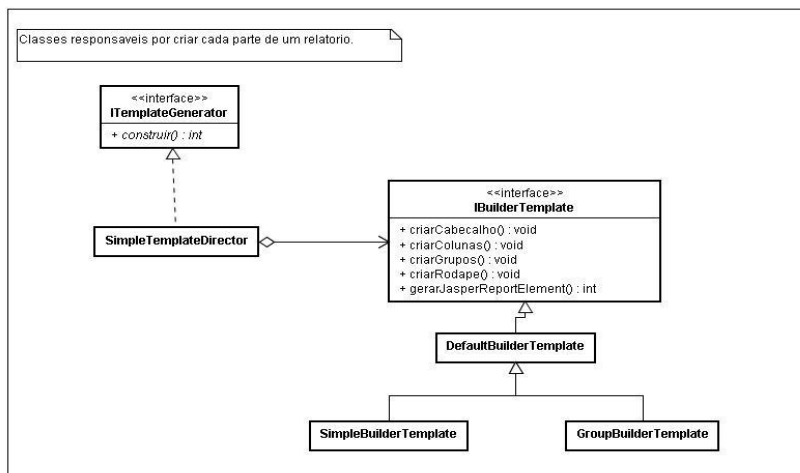


Figura 5 - Classes responsáveis pela estrutura do Template do relatório.

O pacote templates XML contém arquivos em formato XML seguindo o esquema definido para a criação de documentos em PDF utilizando a API JasperReport. Esta API foi criada com a finalidade de possibilitar a geração e manipulação de documentos em formato PDF, assim como permitir a exportação para outros formatos. Os arquivos existentes dentro deste pacote estão classificados conforme a orientação do documento, paisagem ou retrato, bem como o tipo do relatório: se o relatório possui agrupamentos, se possui funções ou se possuem ambas. As funções aqui mencionadas são funções de agrupamento do SQL (sum, count, max e entre outras).

Esta é a visão geral do módulo Relator, cuja versão atual contempla somente a criação de relatórios em formato PDF, mas que até o final do projeto também disponibilizará a saída em outros formatos, assim como permitirá ao cliente formatar visualmente, definindo, por exemplo: o tamanho da fonte e larguras das colunas um relatório a ser mostrado.

4. CONCLUSÃO

A utilização de ferramentas de *software* livre no desenvolvimento de aplicações governamentais é uma prática adotada recentemente no país, com resultados que demonstram a viabilidade tanto técnica quanto econômica deste tipo de tecnologia. As tecnologias utilizadas neste trabalho funcionaram sem nenhum tipo de dificuldade, favorecendo de forma positiva o desenvolvimento das funcionalidades a serem executadas pela aplicação. Além disso, a utilização da idéia de *webservice* na concepção do módulo Relator favorecerá a interoperabilidade entre vários clientes, uma vez que estes só precisam conhecer a interface do *webservice* para receber como retorno um relatório.

Assim, verifica-se com este trabalho que as necessidades governamentais podem ser perfeitamente atendidas com a utilização de *software* livre ao mesmo tempo em que favorece a capacitação tecnológica e científica de professores e alunos da Rede Federal de Ensino Tecnológico nesse tipo de tecnologia, funcionando, como importante política indutora de criação de grupos de pesquisa na área de informática.

REFERÊNCIAS

ANDERSON, Richard et al. **Professional XML**. 1. ed. Rio de Janeiro: Ciência Moderna, 2001. 1266p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023** : Informação e documentação: Referências: Elaboração. Rio de Janeiro, 2002a.

_____. **NBR10520**: Informação e documentação: Citações em documentos: Apresentação. Rio de Janeiro, 2002b.

BARATO, Jarbas Novelino. Educação Profissional. SENAC São Paulo, 2004

BARCELLOS, M. Almeida. **Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares**. UFMGs. Ci. Inf., Brasília, v. 31, n. 2, p. 5-13, maio/ago. 2002

BEA, IBM, Microsoft, AG, S., and Siebel (2003). **Specification: Business process execution language for webservices version 1.1**. Disponível em: <<http://www-106.ibm.com/developerworks/library/ws-bpel/>>. Acesso em: agosto 2008.

BONDUKI, N. **Origens da habitação social no Brasil**. 4. ed. São Paulo: Estação Liberdade, 2004.

CARDOSO, R. **Uma introdução à história do design**. 2. ed. rev. e amp. São Paulo; Edgard Blücher, 2004.

DAY, R.A. **Como escrever e publicar um artigo científico**. 5. ed. São Paulo: Santos Editora, 2001. 275 p.

DAUM, Berthold; MERTER, Udo. **Arquitetura de sistemas com XML**. 1. ed. Rio de Janeiro: Campus, 2002. 441p.

DEITEL, Harvey M. **Java – Como Programar**. Prentice Hall Brasil, 2007.

GREGO, Maurício. **De carona com Webservices**. Revista INFO Corporate, São Paulo: Abril, n.1, p. 28-37, dez. 2002.

GOTTSCHALK, K., Graham, S., Kreger, H., and Shell, J. (2002). **Introduction to webservices architecture**. IBM Systems journal, 41:170–177.

LEWIS John; CHASE, Joe; DEPASQUALE, Peter. **Java Foundations**. Addison Wesley, 2007.

MANFREDI, Silvia Maria. **Educação Profissional no Brasil**. Cortez, 2003.

MORO, M.M. **Dicas para escrever artigos científicos**. Disponível em:
<<http://www.cs.ucr.edu/~mirella/Dicas.html>> Acesso em: 12 fev 2007.

MOULTIS, Natanya Pitts; KIRK, Cheryl. XML: **black book**. 1. ed. São Paulo: MAKRON Books, 2000. 627p.

PAIXÃO, Carlos Feliz. Geração de Relatório com JasperReports e iReport, Grupo de Usuários Java – <<http://www.guj.com.br>>. Acesso em: agosto de 2008

RAY, Erick. **Aprendendo XML**. 1. ed. Rio de Janeiro: Campus, 2001. 372p.

SAVITCH, Walter; MOCK, Kenrick. **Absolute Java**. Addison Wesley, 2007.

SAMPAIO, Cleuton. Guia **do Java Enterprise Edition 5**. Brasport, 2007.

WOHED, P., van der Aalst, W. M. P., Dumas, M., and ter Hofstede, A. H. M. (2002). **Pattern based analysis of bpel4ws**. Technical Report FIT-TR-2002-04, Queensland University of Technology (QUT).

W3 CONSORTIUM. Extensible Markup Language (XML): activity statement. Disponível em:
<<http://www.w3c.org>>. Acesso em: agosto de 2008.

PAIXÃO, Carlos Feliz. Geração de Relatório com JasperReports e iReport, Grupo de Usuários Java 2005– <<http://www.guj.com.br>>. Acesso em: agosto de 2008