



Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática

2008. 1

Um protótipo para emissão e gerenciamento de
notas fiscais eletrônicas
Trabalho de Graduação

Bruno Rodrigues (br@cin.ufpe.br)

Orientador: Fernando da Fonseca de Souza (fdfd@cin.ufpe)

Agradecimentos

Agradeço prioritariamente aos meus pais, que me apoiaram em todas as etapas da minha formação, acadêmica ou não, colocando os estudos antes de qualquer outra coisa. Agradeço a paciência de Flávia da Rosa Borges, por ter sido deixada de lado enquanto eu passava minhas madrugadas no Centro de Informática, que ela mesma chama, com razão, de “inferno”.

Agradeço também a todos os músicos que tornam meu dia-dia mais agradável e minhas dificuldades mais suportáveis. Dentre eles, os principais: Hermeto Pascoal, Uakti, Egberto Gismonti, Arrigo Barnabé (por me mostrar a música dodecafônica), Itamar Assumpção (e toda a vanguarda paulista), Jards Macalé, Walter Franco, Tom Zé, Frank Zappa, Kraftwerk, Jimi Hendrix, entre outros.

Um “Obrigado” a Jack Kerouac e todos os *beatniks*, que além de subversivos, escreveram algo diferente de manuais técnicos e tornaram meus passeios diários no “CDU/Várzea” mais interessantes, além de contribuírem com a minha formação pessoal. Um sincero agradecimento aos cineastas Alejandro Jodorowsky, David Cronenberg,, Michael Haneke, Lars Von Trier, Pedro Almodóvar, e todos os asiáticos que dispensam comentários.

Agradeço também aos professores que fizeram minha graduação valer a pena, especialmente aos professores Geber Ramalho, Fernando Fonseca, e Sílvio Meira, figuras humanas no meio de máquinas. Ser professor, por si só, já é algo digno de aplausos. Aproveito pra agradecer a Cleiton e Magare, os donos do Bar do Bigode, por terem criado o ambiente de maior socialização da UFPE.

Por fim, preciso agradecer à dançarina Carla Perez por me mostrar o que acontece com quem não estuda, e ao pagodeiro Alexandre Pires, por ter chorado como uma criança nos braços de George W. Bush, e ter me dado a certeza de que esse o mundo é um circo que não precisa ser levado tão a sério.

" Quanto mais nos elevamos, menores parecemos àqueles que não sabem voar."

Friedrich Wilhelm Nietzsche

RESUMO

Um novo modelo nacional de notas fiscais está sendo estudado, desde 2005, pelas Secretarias da Fazenda dos Estados Brasileiros. Esse novo modelo propõe a substituição da nota fiscal em papel pelo formato eletrônico, enviado pelos contribuintes através da internet, obedecendo a formatos e padrões pré-estabelecidos.

O objetivo desse trabalho de graduação é estudar e desenvolver um software capaz de manipular informações em bases de dados referentes às vendas de serviços ou produtos de um contribuinte qualquer, formatá-las em arquivos xml, possibilitando assim o encapsulamento dessa informação em pacotes assinados digitalmente para serem trafegados na internet, com destino aos servidores da Secretaria da Fazenda, via webservices. A aplicação em questão tem caráter de protótipo devido às limitações apresentadas pelos webservices em atividade nos servidores das Secretarias da Fazenda dos Estados, os quais impossibilitam testes reais feitos por uma pessoa física.

Palavras-chave: XML, Assinatura digital, Webservices, Ferramentas.

Sumário

1	<i>Introdução</i>	8
1.1	Problemática	9
1.2	Objetivos da Monografia	11
1.2.1	Objetivo Principal	11
1.2.2	Objetivos Específicos	11
1.2.3	Relevância	11
1.3	Organização da Monografia	12
2	<i>O projeto NFe</i>	13
2.1	Modelo operacional da NFe	13
2.2	Mensagens trafegadas no modelo NFe	14
2.2.1	Extensible Markup Language	14
2.2.2	Padrões técnicos das mensagens XML no projeto NFe	15
2.3	Comunicação com o contribuinte	15
2.3.1	Serviços	15
2.3.2	Webservices e SOAP	17
2.3.2.1	WSDL – Webservice Description Language	18
2.3.3	Webservices NFe	19
2.4	Certificação Digital	21
2.4.1	Criptografia.....	21
2.4.1.1	Criptografia Simétrica.....	22
2.4.1.2	Criptografia de chave pública	22
2.4.2	Assinatura Digital	23
2.4.3	Certificado Digital	25
2.4.4	Padrões de assinatura e certificado do Projeto NFe	25
3	<i>O protótipo para emissão de notas fiscais</i>	27
3.1	Tecnologias	27
3.1.1	Framework XStream.....	28
3.1.2	Framework XFire	29
3.1.3	Apache POI	30

3.2	Arquitetura.....	30
3.3	Funcionalidades e Telas.....	31
4	<i>Conclusão</i>	38
4.1	Dificuldades Encontradas.....	39
4.2	Trabalhos Futuros.....	39
	<i>Referências</i>	40
	<i>Apêndice A</i>	43
	<i>Apêndice B</i>	44

LISTA DE FIGURAS

Figura 1: Emissor de nota fiscal eletrônica criado pela SEFAZ-SP.....	12
Figura 2: Emissor de nota fiscal eletrônica criado pela SEFAZ-SP.....	12
Figura 3: Serviço síncrono[SEFAZ,2007].....	17
Figura 4: Serviço assíncrono[SEFAZ,2007].	18
Figura 5: Aplicação cliente acessando diretamente um Webservice.[CUNHA,2002]....	20
Figura 6: Exemplo de mensagem encapsulada em um envelope SOAP[SEFAZ,2007] .	21
Figura 7: Comunicação no envio de um lote NFe(serviço assíncrono) [SEFAZ,2007]..	22
Figura 8: Assinatura digital utilizando algoritmos de chave pública[IRIB,2005].....	25
Figura 9: Conferência da assinatura digital[IRIB,2005]	25
Figura 10: Framework XStream integrado com o recurso Annotations.....	29
Figura 11: Plugin do XFire para o Eclipse.	30
Figura 12: Chamadas Apache POI para acesso a dados em arquivos xls.....	30
Figura 13: Arquitetura MVC.	31
Figura 14: Diagrama de casos de uso	32
Figura 15: Tela inicial.....	33
Figura 16: Tela de preparaçãode mensagens para envio via webservice	33
Figura 17: Combobox com os tipos de mensagens para envio via webservice.....	34
Figura 18: Mensagem retornada pelo servidor de simulação	35

LISTA DE QUADROS

Quadro 1: Tipos de serviços disponíveis[SEFAZ,2005].....	21
---	----

1 Introdução

Um assunto em evidência no Brasil, é a integração de informações tributárias entre os estados, que segundo a constituição, possuem autonomia política própria. Isso significa que cada estado brasileiro é independente na sua administração fiscal.

Toda essa descentralização fiscal tem gerado transtornos administrativos, tornando difíceis a manipulação e a integração desses dados fiscais oriundos de diferentes estados, criando uma imagem burocrática e ineficiente do governo, diante dos contribuintes.

Essa questão se apresenta mais delicada em empresas que atuam em diferentes estados, e que são justamente as maiores empresas, e que mais contribuem. Essas empresas se deparam com diferentes regras em cada uma das unidades federativas nas quais atuam, e acabam sendo prejudicadas por essa falta de uniformidade, arcando com custos de alocação de recursos humanos para registro e gerenciamento de tais informações. Todo esse cenário, somado ao fato de que todos esses registros são armazenados em incalculáveis quantidades de papéis, desencadeou a introdução do Inciso XXII ao art. 37 da Constituição Federal, por meio da Emenda Constitucional nº 42, que determina que as diferentes administrações tributárias dos diferentes estados devem atuar de forma integrada, tornando público, inclusive, informações fiscais uns aos outros. Essa mudança no *status quo* do sistema fiscal só pode ser efetivada através da inserção de tecnologia, garantindo assim a consistência dos dados e a quebra dos mais diversos entraves burocráticos do modelo atual.

Foi partindo dessas premissas que em 2005 o governo criou o projeto de Nfe (Nota Fiscal eletrônica), fazendo com que toda informações fiscais sejam digitais e com total validade jurídica, anulando qualquer uma das preocupações acima citadas.

1.1 Problemática

Imaginemos o cenário onde a implantação do projeto de notas fiscais eletrônicas tenha acontecido com o sucesso esperado, com servidores ativos 24 horas por dia, pronto para receberem dados referentes às notas fiscais emitidas nas mais diferentes empresas do Estado, e se comportando de maneira esperada segundo suas especificações. Os problemas parecem estar resolvidos, até que surge a seguinte pergunta: e o contribuinte, como vai eliminar o antigo modelo de notas fiscais em papel, e começar a emitir essa informação digitalmente, e no formato técnico exigido?

A resposta pode parecer simples, se forem considerados apenas os contribuintes de pequeno porte. Essa categoria de contribuinte pode se adaptar ao novo modelo sem grandes implicações de custo, considerando que ela pode substituir o seu software de vendas de produtos ou serviços por um novo aplicativo que “gera” as notas fiscais eletrônicas e as envia para os servidores associados àquele Estado, de forma transparente para o vendedor.

Agora imagine uma grande empresa que investiu milhões de reais para automatizar o seu controle de vendas, além de tantas outras tarefas necessárias para o gerenciamento administrativo da mesma. Tal empresa não pode cogitar a possibilidade de descartar o seu sistema atual para atender às novas exigências da nota fiscal eletrônica, por esse motivo se torna inviável o uso de uma nova ferramenta.

Pensando nisso, a Secretaria da Fazenda (SEFAZ) do Estado de São Paulo desenvolveu um software emissor de notas fiscais eletrônicas, visando acelerar a inclusão das empresas ao novo modelo fiscal. O software em questão foi projetado pensando na inserção totalmente manual dos dados, tornando o processo igualmente custoso se comparado com o modelo da nota em papel, além de dificultar a inclusão das grandes empresas, pelo motivo citado no parágrafo anterior. Tal ferramenta se encontra disponível para *download* gratuito no site <http://www.nfe.fazenda.gov.br/portal/emissor.aspx>, porém, em caráter de testes,

percebido pela notável instabilidade apresentada pelo mesmo, além de atualizações quase diárias, fazendo com que o usuário se sinta inseguro.

Algumas telas do Emissor desenvolvido pela SEFAZ-SP são mostradas nas Figuras 1 e 2.

The screenshot shows the 'Cadastro de Emitente' (Issuer Registration) screen. The title bar reads 'Emissor de Nota Fiscal Eletrônica (NF-e) - VERSÃO DE TESTE'. The menu bar includes 'Notas Fiscais', 'Emitente', 'Cadastros', 'Sistema', and 'Ajuda'. The main area is titled 'Cadastro de Emitente' and contains the following fields:

- Emitente:**
 - * Nome/Razão Social: [Empty text box]
 - Nome Fantasia: [Empty text box]
 - * CNPJ: [Empty text box]
 - CNAE Fiscal: [Empty text box]
 - Inscrição Estadual: [Empty text box]
 - Inscrição Municipal: [Empty text box]
 - Inscrição Estadual (Subst. Tributário): [Empty text box]
- Endereço:**
 - * Logradouro: [Empty text box]
 - * Número: [Empty text box]
 - Complemento: [Empty text box]
 - * Bairro: [Empty text box]
 - CEP: [Empty text box]
 - País: [BRASIL (dropdown)]
 - * UF: [Empty dropdown]
 - * Município: [Empty dropdown]
 - Telefone: [Empty text box with area code placeholder ()]
- Logotipo:**
 - Placeholder: Sem logotipo
 - Buttons: Selecionar Imagem..., Deixar em Branco

At the bottom, there are 'Incluir' and 'Voltar' buttons, and a note: (*) Campo de preenchimento obrigatório.

Figura 1. Emissor de nota fiscal eletrônica criado pela SEFAZ-SP. Tela de cadastro de emitente.

The screenshot shows the 'Nota Fiscal' (Fiscal Note) screen. The title bar reads 'Emissor de Nota Fiscal Eletrônica (NF-e) - VERSÃO DE TESTE'. The menu bar is the same as in Figure 1. The main area is titled 'Nota Fiscal' and contains the following information:

- Status:** Em Digitação
- Chave de acesso:** [Empty text box]
- Número:** 00000004
- Versão XML:** 1.10
- Transporte:** [Empty dropdown]
- Cobrança:** [Empty dropdown]
- Informações Adicionais:** [Empty dropdown]
- Exportação e Compras:** [Empty dropdown]
- Certificado Digital:** [Empty dropdown]
- Dados da NF-e:** [Empty dropdown]
- Emitente:** [Empty dropdown]
- Destinatário/Remetente:** [Empty dropdown]
- Produtos e Serviços:** [Empty dropdown]
- Totais:** [Empty dropdown]
- Identificação:**
 - * Tipo do documento: [CNPJ (dropdown)]
 - * CNPJ/CPF: 91.878.450/6854-82
 - * Razão social/Nome: CLIENTE 2
 - Inscrição Estadual: [Empty text box]
 - Inscrição SUFRAMA: [Empty text box]
 - Isento do ICMS
- Endereço:**
 - * Logradouro: Rua Dantas Barreto
 - * Número: 598
 - Complemento: [Empty text box]
 - * Bairro: Sao Jose
 - CEP: 50000-000
 - País: [BRASIL (dropdown)]
 - * UF: [PE (dropdown)]
 - * Município: [Recife (dropdown)]
 - Fone: (81)34196656
- Local Retirada/Entrega:**
 - Local de retirada diferente do emitente
 - Endereço: [Empty text box]

Figura 2. Emissor de nota fiscal eletrônica criado pela SEFAZ-SP. Tela de cadastro de nota fiscal.

1.2 Objetivos da Monografia

Nesta seção serão apresentados o objetivo principal desse trabalho e o detalhamento dos objetivos específicos.

1.2.1 Objetivo Principal

O objetivo principal desta monografia é propor um estudo do novo modelo tributário a ser implantado nos diversos distritos brasileiros, gerando como resultado um software simulador de emissão de notas fiscais eletrônicas, que sugira uma solução para as grandes empresas que pretendem migrar para os novos padrões fiscais, sem a necessidade de remodelar ou recriar o seu sistema atual de controle de transações de vendas.

1.2.2 Objetivos Específicos

Neste contexto, este trabalho de graduação propõe a construção de um protótipo que possua uma interface para leitura de uma determinada base de dados, e que possa tratar esses dados gerando arquivos xml no formato especificado pelos manuais técnicos do projeto NFe, disponibilizado nos websites da SEFAZ dos estados, possibilitando assim a realização de emissão, cancelamento e consultas de notas fiscais

1.2.3 Relevância

O projeto escolhido teve como principal motivação a abrangência do conceito de nota fiscal, presente como objeto de controle econômico no país. Partindo desse fato, é notável a relevância de estudos voltados para a agilização, robustez e dinamismo da economia nacional.

1.3 Organização da Monografia

Além deste capítulo, esta monografia encontra-se organizada em mais 3 (três) capítulos:

Capítulo 2 contém uma visão geral dos principais conceitos sobre o projeto NFe, além de explicar detalhadamente toda a tecnologia envolvida na solução proposta pela especificação técnica do mesmo.

Capítulo 3 contém uma visão detalhada de algumas tecnologias que viabilizam a construção de um software gerenciador de notas fiscais eletrônicas, bem como a avaliação de um protótipo construído como resultado prático desse estudo.

Capítulo 4 contém as conclusões obtidas durante o desenvolvimento do trabalho, as dificuldades encontradas, assim como sugestões para possíveis trabalhos futuros.

2 O projeto NFe

O projeto Nota Fiscal Eletrônica (NFe) é um projeto iniciado no ano de 2005, e ainda não finalizado, estando em desenvolvimento sob a responsabilidade do Encontro Nacional de Coordenadores e Administradores Tributários Estaduais (ENCAT), em conjunto com a Receita Federal.

O projeto prevê a substituição das notas fiscais em papel por um documento fiscal eletrônico com validade jurídica garantida pela aplicação do conceito de Assinatura Digital. Todos os documentos técnicos e informações gerais estão disponibilizados no Portal Nacional da Nota Fiscal Eletrônica, no endereço <http://www.nfe.fazenda.gov.br/portal>.

As próximas sessões irão tratar dos quesitos técnicos envolvidos no projeto NFe, que pode ser resumido na figura apresentada no Apêndice B.

2.1 Modelo operacional da NFe

O ciclo de vida de uma nota fiscal eletrônica começa na empresa emitente, quando, depois de uma transação comercial, que pode ser a venda de um serviço ou produto, são enviados os dados obrigatórios, segundo especificações técnicas, através da internet para os servidores da SEFAZ do estado referente à transação. Porém, para garantir a autoria do emissor, esse documento eletrônico precisa ser assinado digitalmente antes do envio.

Passada essa etapa, o computador servidor da SEFAZ, o qual recebeu o documento, fará uma validação quanto ao formato e validade dos dados, e retornará uma mensagem de autorização para o emitente. O recebimento dessa autorização é obrigatório para que o trânsito da mercadoria de fato aconteça.

Estando a mercadoria devidamente autorizada, o emitente deverá imprimir um documento que conterá informações do produto, além de uma chave de acesso e código de barras, que permitirá a consulta da NFe pela internet, garantindo a confirmação das informações. A esse documento foi atribuído o nome de

Documento Auxiliar da Nota Fiscal Eletrônica – DANFE, sendo importante deixar claro que o mesmo não possui validade jurídica alguma, sendo apenas uma forma de comprovação gráfica de que a nota fiscal foi enviada.

Enquanto isso, de forma transparente para o emitente, a SEFAZ estadual enviará a NFe em questão para a Receita Federal, que guardará em repositório todas as notas de todos os estados.

2.2 Mensagens trafegadas no modelo NFe

A requisição de todo e qualquer serviço da NFe, bem como o retorno dessas solicitações, estarão no formato XML – Extensible Markup Language[W3C,1990].

2.2.1 Extensible Markup Language

XML é um tipo de linguagem de marcação criada com o intuito de facilitar o compartilhamento de informações através da internet. A sua concepção prezou pela legibilidade tanto para pessoas quanto para computadores, além do desacoplamento entre conteúdo e formatação.

O formato xml deixa o desenvolvedor livre para criações de novas *tags*, e também dispõe de arquivos de validação de estrutura chamados DTD(Data Definition Type). Os DTD's são documentos que contêm as regras que definem quais as tags que podem ser usadas em um documento XML e quais os valores válidos. Uma alternativa ao DTD é o *XML Schema*, linguagem baseado em XML usada com o mesmo propósito, apresentando algumas diferenças, como suporte a tipos de dados, além de serem extensíveis para adições futuras.

De XML surge um conceito importante chamado XML *namespace*, que nada mais é que a descrição da sintaxe de *namespace*, possibilitando a criação de prefixos para o nome das *tags*, evitando confusões de nomes iguais para *tags* diferentes. *XML schema* suportam namespaces.

2.2.2 Padrões técnicos das mensagens XML no projeto NFe

A versão do XML utilizado pelo projeto NFe é a 1.0, e a codificação dos caracteres obedecerá o UTF-8, sendo obrigatório a inserção da seguinte informação no início de cada arquivo:

```
<?xml version="1.0" encoding="UTF-8"?>
```

A indicação do namespace deve aparecer no elemento raiz:

```
<NFe xmlns="http://www.portalfiscal.inf.br/nfe" >
```

Os arquivos que definem os XML schemas, também chamados de XSD, estão disponíveis em <http://www.nfe.fazenda.gov.br/portal/schemas.aspx>.

2.3 Comunicação com o contribuinte

É detalhada nas subseções seguintes.

2.3.1 Serviços

Toda e qualquer comunicação entre contribuinte e SEFAZ será iniciado pelo aplicativo do contribuinte, que enviará a solicitação do serviço desejado.

Serviços se enquadram em duas categorias[SEFAZ,2007]:

- **Serviços síncronos**- serviço onde o processamento acontece na própria conexão da solicitação do mesmo, ou seja, o aplicativo emitente envia informações para um determinado serviço, esse serviço é processado na SEFAZ, e em seguida é retornado um resultado para o aplicativo emitente, seguindo do encerramento da conexão. Mostrado na Figura 3.



Figura 3. Serviço Síncrono [SEFAZ, 2007].

- **Serviços assíncronos**- serviço onde o processamento não acontece na própria conexão da solicitação do mesmo, ou seja, o aplicativo emissor envia informações para um determinado serviço, esse serviço não é processado imediatamente na SEFAZ, ficando numa “lista de espera”, aguardando a sua vez, enquanto a conexão é encerrada. Esse tipo de serviço acontece pela necessidade de uso de vários processos distribuídos. Mostrado na Figura 4.

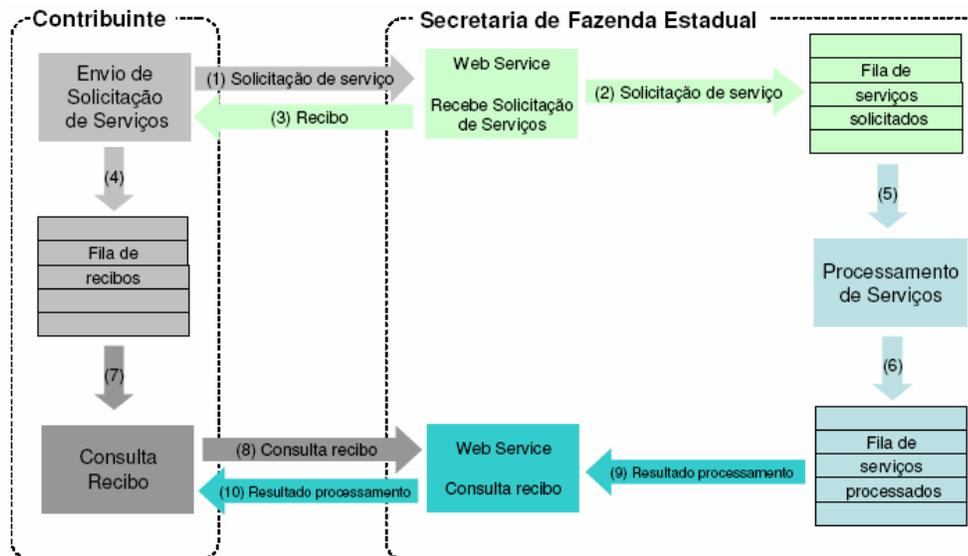


Figura 4. Serviço Assíncrono [SEFAZ, 2007].

Esses serviços, síncronos ou assíncronos, que “escutam” e “enviam” requisições são chamados *Webservices* [W3C,2002], e seguem um padrão de comunicação chamado SOAP (*Simple Object Access Protocol*) [W3C,2004].

2.3.2 Webservices e SOAP

Quando a internet começou a ganhar popularidade ao redor do mundo, os usuários tinham um papel unicamente passivo na rede, resumindo-se a simples clientes que requisitavam informações de servidores, os quais enviavam as informações solicitadas para exibição no computador pessoal do usuário. Para isso, era necessária uma linguagem padronizada que permitisse a exibição desse conteúdo trafegado, compatível com qualquer navegador web. A linguagem em questão foi a HTML (HyperText Markup Language) [W3C,1995].

Porém, novas tecnologias surgiram com o passar dos anos, e cada vez mais aparecem aplicações baseadas em tecnologias diferentes entre si, e que ao mesmo tempo precisam trocar informações umas com as outras.

Partindo dessa necessidade, nasce conceito de *webservice*, que foi definido pela W3C como sistemas de softwares projetados para serem interoperáveis dentro de uma rede.

Para facilitar o entendimento do conceito, convém citar um exemplo comum no dia-dia de quem usa a internet: imagine que o usuário está se cadastrando em algum portal de comércio eletrônico, e em algum momento do cadastro é solicitado que digite o número de CPF. Distraidamente, esse usuário digita algum número errado, e o sistema lança um aviso de CPF inválido. Como um simples portal de comércio eletrônico pode validar números de CPF?

Obviamente, o portal de vendas não vai se dar ao trabalho de armazenar em sua base de dados todos os números de CPF do país, com o propósito único de

garantir que não serão cadastradas pessoas com CPF falsos. Na verdade, o portal apenas requisitou automaticamente um serviço disponível nos servidores da Receita Federal, usado para consulta de CPF. Esse serviço disponibilizado é chamado de *webservice*, e não precisa ser necessariamente implementado na mesma tecnologia do site do portal de vendas, porém, precisa de uma troca de mensagens em um “idioma” que seja compreendido pelas duas aplicações. A esse “idioma” se dá o nome de protocolo, e que no caso da SEFAZ, é baseado em um dos mais usados protocolos para *webservices*, conhecido como SOAP. Este é um protocolo que utiliza documentos em formato xml, e que aderem a uma especificação também descrita pelo W3C.

2.3.2.1 WSDL – Webservice Description Language

Para acessar algum serviço na internet, a aplicação cliente precisa saber de algumas informações a serem passadas para o *webservice*, como por exemplo, os parâmetros que um determinado serviço recebe. No exemplo do cadastro no portal de *e-commerce*, provavelmente o serviço de consulta do número do CPF deve receber apenas um parâmetro, no caso, o próprio número.

Para resolver esse problema, foi criado um documento baseado novamente em xml, chamado WSDL, que descreve um conjunto de mensagens SOAP e a forma como essas mensagens são trocadas[IWEB,2003].

Tendo acesso ao WSDL de um determinado *webservice*, basta construir uma aplicação, em qualquer tecnologia, que acesse os serviços da maneira descrita, e a comunicação está estabelecida.

A Figura 5 mostra a conexão de um cliente a um *webservice*.

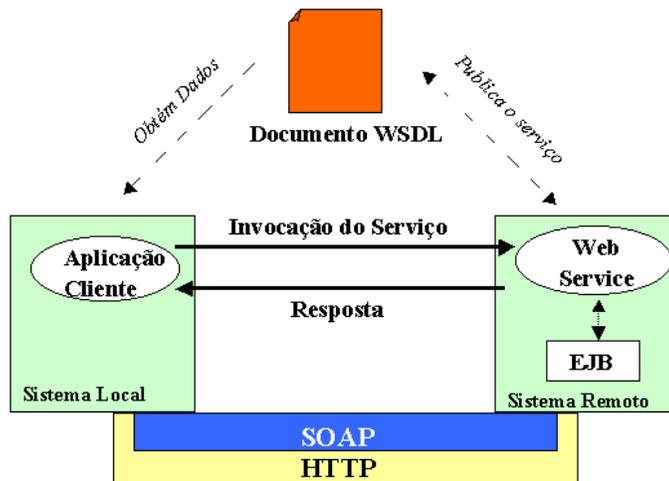


Figura 5. Aplicação cliente acessando diretamente um Webservice [CUNHA,2002].

2.3.3 Webservices NFe

A comunicação entre o contribuinte e a Secretaria da Fazenda do estado ao qual ele pertence se dá através da troca de mensagens entre os *Webservices* seguindo o padrão SOAP, com troca de mensagens XML no padrão *Style/Encoding: Document/Literal, wrapped*. A opção “*wrapped*” significa a passagem de mais de um parâmetro aos métodos disponíveis. Os parâmetros usados para qualquer chamada aos *webservices* da SEFAZ, possuem dois parâmetros do tipo *string*, com os nomes de: “*nfeCabecMsg*” e “*nfeDadosMsg*”.

A Figura 6 mostra um exemplo de uma mensagem dentro de um envelope SOAP:

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP:Body>
<nfeRecepcaoLote xmlns="http://www.portalfiscal.inf.br/nfe/wsd/NfeRecepcao">
<!-- XML Área de Cabeçalho -->
<nfeCabecMsg>
  <?xml version="1.0" encoding="UTF-8"?>
  <cabecMsg xmlns="http://www.portalfiscal.inf.br/nfe" versao="1.02">
    <versaoDados>1.07</versaoDados>
  </cabecMsg>
</nfeCabecMsg>
<!-- XML Área de Dados -->
<nfeDadosMsg>
  <?xml version="1.0" encoding="UTF-8"?>
  <enviNFe xmlns="http://www.portalfiscal.inf.br/nfe" versao="1.07">
    <idLote>0000000000000001</idLote>
    <NFe xmlns="http://www.portalfiscal.inf.br/nfe"
      <infNFe Id="NFe43060600000000000191550990000070080007055472" versao="1.07">
      ...
    </NFe>
  </enviNFe>
</nfeDadosMsg>
</nfeRecepcaoLote>
</SOAP:Body>
</SOAP:Envelope>

```

Figura 6. exemplo de mensagem encapsulada em um envelope SOAP[SEFAZ,2007]

A área do cabeçalho (nfeCabecMsg), é utilizada unicamente para controle, contendo informações da versão dos XML schemas que validam o conteúdo no corpo da mensagem (nfeDadosMsg).

A SEFAZ estabeleceu a existência de um *webservice* para cada serviço, que pode ser classificado quanto ao seu tipo, como no Quadro 1:

Quadro 1.Tipos de serviços disponíveis[SEFAZ,2005].

Serviço	Implementação
Recepção de NF-e	Assíncrona
Cancelamento de NF-e	Síncrona
Inutilização de Numeração de NF-e	Síncrona
Consulta da situação atual da NF-e	Síncrona
Consulta do status do serviço	Síncrona

A Figura 7 mostra a comunicação para envio de um lote de notas fiscais eletrônicas.

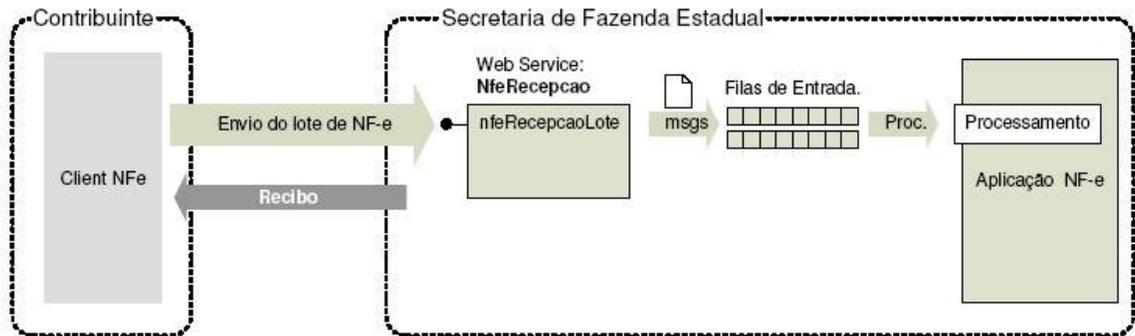


Figura 7. Comunicação no envio de um lote NFe(serviço assíncrono)[SEFAZ,2007]

As URL's (Uniform Resource Locator), ou seja, endereço dos serviços na internet, encontram-se disponíveis nos sites das SEFAZ dos estados. Acessando a URL pode ser obtido o WSDL (Webservices Description Language) de cada Webservice.

2.4 Certificação Digital

A rede mundial de computadores cada vez agiliza o envio de dados e documentos para as diversas partes do mundo, de forma quase instantânea. Porém, se faz necessário alguma estratégia para garantir que esses dados trafegados sejam confidenciais, autênticos e íntegros. Dessa necessidade surgiu a idéia de Certificação Digital[X.509,1988], conceito que vem amadurecendo desde a década de 80, oriundo do estudo da criptografia.

2.4.1 Criptografia

A palavra criptografia carrega em sua etimologia a junção da palavra “código” com a palavra “escrita”, e estuda as formas de se transformar informações legíveis por pessoas ou máquinas em informações cifradas, ilegíveis,

que podem ser decifradas ao conteúdo original através de uma “chave” [IRIB,2005].

Apesar da criptografia abranger bem mais do que a codificação e decodificação de informação, este tópico se limitará a essa área de estudo.

Para criptografar uma determinada informação, um computador recebe a informação e um número chave, usado no algoritmo de transformação dos dados. O processo que permite decifrar essa informação só é possível, novamente, com o uso de um número chave, o que leva a conclusão de que, para manter um texto qualquer sob sigilo, basta que o mesmo seja criptografado, e o número de chave seja guardado em segredo.

Existem duas formas de se fazer esse tipo de criptografia nos dias atuais: a *criptografia simétrica* e a *criptografia de chave pública*.

2.4.1.1 Criptografia Simétrica

Nesse caso, o número de chave usado na cifragem da informação é o mesmo usado na decifragem da mesma, fazendo com que essa chave seja compartilhada com quem vai cifrar e que vai recuperar de volta a mensagem original. A forma como esse número de chave será compartilhado entre essas duas entidades deve ser segura, visto que se uma terceira entidade tem conhecimento desse número, as informações podem ser facilmente lidas [IRIB,2005].

2.4.1.2 Criptografia de chave pública

No caso da criptografia de chave pública, são usadas duas chaves diferentes, uma privada e a outra pública. A operação realizada com uma delas pode ser revertida com a outra. A chave pública, como o nome já diz, deve ficar disponível para o acesso de quem tiver interesse, enquanto que a chave privada deve ser guardada em sigilo por quem a gerou. [IRIB,2005].

Esse tipo de criptografia é atualmente o mais usado, e se apresenta bastante eficiente por garantir não apenas o sigilo das informações, mas garante também a autoria de quem está criptografando as informações.

Para garantir a autenticidade, o autor de um determinado documento submete o mesmo à uma cifragem utilizando sua chave privada, e pode disponibilizar, inclusive na internet, sua chave pública, para quem quiser decifrar o documento.

Sabendo-se que a chave privada que gerou aquele documento agora decifrado é de conhecimento apenas do autor daquele documento, esse processo garante a sua autoria.

Fazendo-se um processo inverso, podemos garantir a confidencialidade do documento. Nesse caso, o remetente utiliza a chave pública do destinatário no processo de cifragem, e envia o documento para o destino, sabendo que apenas com a chave privada do destinatário, mantida sigilosamente por ele, a informação será lida.

2.4.2 Assinatura Digital

A técnica de autenticação viabilizada pelos algoritmos de chave pública, quando são usados em conjunto com um algoritmo chamado Hash[LUHN,1953], temos a Assinatura Digital[ICP-2001]. O Código Hash é o resultado de uma função chamada *função de hash*, que recebe um determinado conjunto de informações, como um texto, por exemplo, e retorna um código único para aquele conteúdo. Caso um texto de milhares linhas seja alterado com a adição ou mudança de apenas um caractere, o código gerado pela função de hash será completamente diferente do anterior. Uma característica importante das funções hash é que elas são unidirecionais, ou seja, não se pode recuperar o conteúdo de um documento a partir do código gerado pela função hash aplicada ao mesmo.

Partindo desse conceito, o usuário X, que queira utilizar da assinatura digital no envio de um documento, submeterá esse documento à uma função

hash, que retornará o código hash referente àquele conteúdo. Em seguida, esse usuário utilizará de um algoritmo de chave pública para cifrar essa informação, usando sua chave privada, gerando assim a sua Assinatura Digital. Depois desse procedimento, qualquer destinatário com acesso à chave pública da pessoa X, decifra o conteúdo criptografado e compara o resultado com o hash do documento. As Figura 8 e Figura 9 mostram algumas imagens para ilustrar o procedimento.



Figura 8. Assinatura digital utilizando algoritmos de chave pública [IRIB,2005].

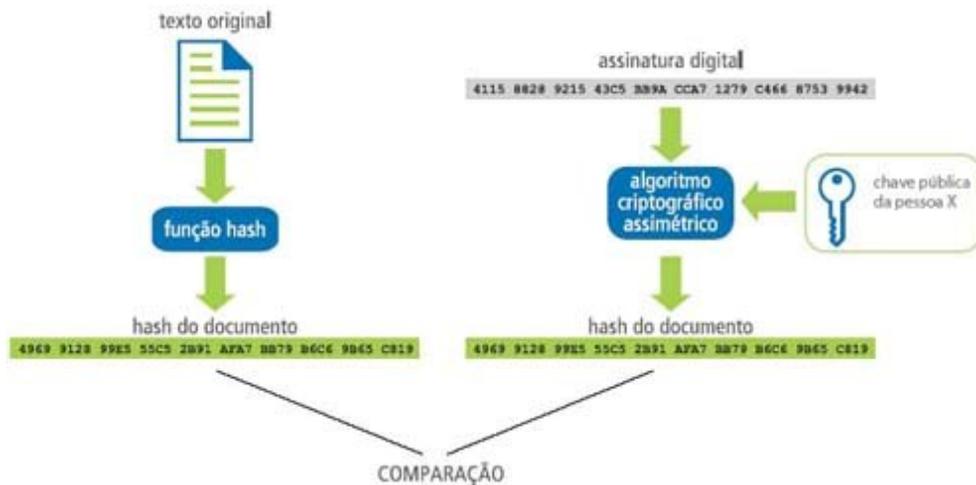


Figura 9. Conferência da assinatura digital[IRIB,2005].

A Medida Provisória (MP) nº 2.200-2, de 24 de agosto de 2001, atribuiu validade jurídica à assinatura digital.

2.4.3 Certificado Digital

O certificado digital é um arquivo que associa uma entidade ou pessoa a uma chave pública. Esse documento é assinado digitalmente por alguma entidade autorizada para esse fim, e possui informações importantes como chave pública, nome da entidade ou pessoa associada a essa chave, nome e assinatura da empresa que assinou o certificado e número de série[IRIB,2005].

Esse tipo de certificado é usado em sistemas bancários pela internet, para garantir ao cliente que o servidor o qual ele está trocando informações é de fato o seu banco. O cliente também pode usar o seu certificado digital para garantir o sentido contrário.

Além de bancos, alguns órgãos governamentais também têm se favorecido com esse conceito, como a Secretaria da Fazenda do Estado de Pernambuco, primeiro estado brasileiro a se utilizar de certificação digital nesse âmbito [IRIB,2005].

A autoridade certificadora raiz brasileira é a ICP(Infra-estrutura de Chaves Públicas[MP, 2001]), e sua principal finalidade é definir um conjunto de técnicas que caracterizam um sistema de certificação digital de chaves públicas.

2.4.4 Padrões de assinatura e certificado do Projeto NFe

As subseções seguintes abordam os formatos técnicos especificados no projeto Nfe para garantir Certificado e Assinatura digital.

O certificado digital no Projeto NFe deverá ser emitido pela ICP-Brasil, sendo do tipo A1 ou A3, contendo o CNPJ do titular da certificação digital.

As mensagens enviada para a SEFAZ, como já foi dito anteriormente, são arquivos no formato XML. Esses arquivos devem ser assinados digitalmente com o certificado que indique o CNPJ do estabelecimento matriz ou emissor.

Algumas informações presentes no certificado digital devem ser dispensadas do arquivo XML. São elas[SEFAZ,2007]:

<X509SubjectName>

<X509IssuerSerial>

<X509IssuerName>

<X509SerialNumber>

<X509SKI>

Deverão ser evitados o uso das seguintes *tags*, pelo mesmo motivo citado acima:

<KeyValue>

<RSAKeyValue>

<Modulus>

<Exponent>

O padrão do formato detalhado pode ser encontrado na página web da W3: <http://www.w3.org/TR/xmlsig-core/>, e o *schema* XML segue no apêndice A desse documento, seguindo a versão mais atual (1.01).

3 O protótipo para emissão de notas fiscais

O resultado prático desse trabalho consiste na construção de um protótipo para simulação de geração e envio de documentos fiscais eletrônico, seguindo especificações técnicas propostas pelo portal da SEFAZ, respeitando limitações encontradas em simulações reais, fazendo com que o servidor da SEFAZ seja simulado localmente a título de demonstração.

3.1 Tecnologias

Toda a implementação realizada nesse trabalho foi realizada na linguagem JAVA[SUN,1995], na versão 1.6. A escolha da linguagem de programação foi motivada na experiência do desenvolvedor, além da produtividade conseguinte das linguagens de alto nível. O ambiente de codificação utilizado foi o Eclipse 3.2[ECLIPSE,2004] . Apesar da IDE(ambiente de desenvolvimento integrado) em questão estar na versão 3.3, a versão anterior foi escolhida devido à possibilidade de integração com o *plugin* Visual Editor[ECLIPSE, 2002], usado para construção das interfaces gráficas.

Toda a base de dados foi simulada em arquivos do tipo planilhas do Excel[MS,1985, tornando mais genérico o acesso aos dados necessários à existência das notas fiscais eletrônicas.

Visando a máxima produtividade de implementação, alguns *frameworks* de desenvolvimento convenientes, detalhados nas subseções seguintes, foram utilizados para poupar trabalho e garantir robustez ao resultado final.

3.1.1 Framework XStream

A manipulação de documentos no formato XML foi auxiliada pelo uso do framework XStream 1.3[CODEHAUS,2008], biblioteca gratuita com ricas funcionalidades no tratamento de arquivos XML, responsável por mapear objetos Java em estruturas de marcação. O XStream garante alta performance de processamento e utiliza de recursos poderosos como *Annotations*, técnica de inserção de metadados dentro do código Java, recurso introduzido a partir da versão 1.5 da linguagem.

A Figura 10 mostra um pequeno exemplo do uso de *Annotations* como metadados para facilitar o mapeamento do *XStream*, indicando o nome que cada atributo vai assumir na estrutura XML, bem como os campos a serem omitidos no documento:

```
@XStreamAlias("retEnviNFe")
public class RetornoEnvioNFe extends Entidade{
    @XStreamOmitField
    public static final int AMBIENTE_PRODUCAO = 1;
    @XStreamOmitField
    public static final int AMBIENTE_HOMOLOGACAO = 2;

    @XStreamAlias("tpAmb")
    private TipoDoAmbiente tipoDoAmbiente;
    @XStreamAlias("verAplic")
    private String versaoAplicativoReceptor;
    @XStreamAlias("cStat")
    private int codigoStatusResposta;
    @XStreamAlias("xMotivo")
    private String descricaoStatusResposta;
    @XStreamAlias("cUF")
    private int codigoUFreceptor;
```

Figura 10. Framework XStream integrado com o recurso Annotations.

3.1.2 Framework XFire

A implementação da comunicação foi alavancada pelo uso do Framework XFire 1.2[ENVOI,2008], biblioteca *opensource* que abstrai a construção dos pacotes SOAP, recebendo apenas o conteúdo dos documentos XML, e atuando no preparo completo dos pacotes que irão trafegar na internet, baseados em especificações passadas como parâmetro durante suas chamadas. O grande ponto facilitador trazido pelo XFire é a integração com a IDE Eclipse, disponibilizando gratuitamente um *plugin* bastante fácil e intuitivo. Tal *plugin* recebe a URL da WSDL, e garante a geração de todo o código dos *stubs* necessários para chamada dos serviços remotos.

A Figura 11 ilustra a simplicidade de uso do *plugin*:



Figura 11. Plugin do XFire para o Eclipse.

3.1.3 Apache POI

A *Java API To Access Microsoft Format Files – Java POI 3.1*[APACHE,2008] foi a biblioteca utilizada para possibilitar que a aplicação acessasse a base de dados em arquivos do tipo .xls, criados pelo editor de planilhas Microsoft Office Excel.

O apache POI, além de gratuito, torna trivial o acesso a arquivos, planilhas e células, através de simples e intuitivas chamadas de métodos, como mostrado na Figura 12.

```
public String getCelulaTextual(short linha, short coluna){  
    ExcelRow row = this.sheet.getRow(linha);  
    return row.getCell(coluna).getStringCellValue();  
}
```

Figura 12.chamadas Apache POI para acesso a dados em arquivos xls.

3.2 Arquitetura

A arquitetura do sistema teve como principal foco a modularidade, e a separação clara entre as camadas de software. O principal motivo dessa preocupação se deve ao fato de que o sistema foi projetado para sofrer mudanças depois de sua implantação em versão inicial.

A arquitetura básica do projeto foi baseada no modelo *Model-view-controller* (MVC)[REENSKAUG,1979], prezando pela separação entre dados (*Model*) e o *layout* (*View*). Isso garante que mudanças em uma dessas camadas não são “enxergadas” pela outra.

O *model-view-controller* resolve este problema com a inserção de uma camada intermediária chamada *controller*, que separa as tarefas de acesso a dados e a lógica de negócio, como mostrado na Figura 13.

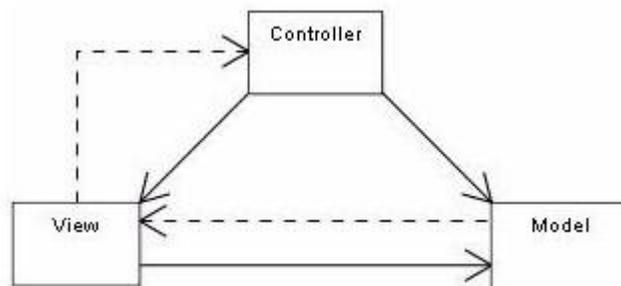


Figura 13. Arquitetura MVC.

A camada *model* é a camada onde estão classe básicas e dados do sistema. Essa camada representa o repositório de dados do sistema, guardados em planilhas, e abstraídos no nível de código em classes Java.

A camada de apresentação, também chamada de *view*, é representada pelas classes que descrevem a interface gráfica do software e se comunica com a camada de dados através de uma camada intermediária controladora.

É justamente nessa camada controladora (*controller*) que estão descritas as regras de negócio, além das chamadas aos *frameworks* citados anteriormente.

3.3 Funcionalidades e Telas

O software emissor de notas fiscais eletrônicas, resultado prático desse projeto de graduação, implementou com maior prioridade a recuperação e tratamento de dados. Em outras palavras, o sistema deu foco às transformações de dados coletados em documentos XML em formato especificado, bem como o encapsulamento em pacotes SOAP.

A Figura 14 mostra um diagrama UML de casos de uso para melhor esclarecer as funcionalidades:

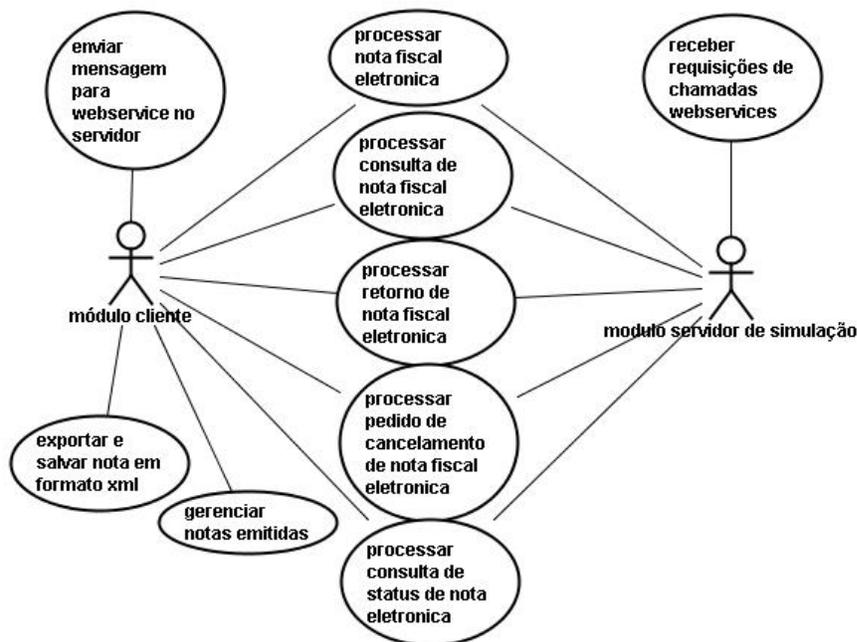


Figura 14. Diagrama de casos de uso

As funcionalidades da figura 14 são detalhadas abaixo:

Processar nota fiscal eletrônica- Essa funcionalidade garante a geração do xml referente à nota fiscal eletrônica validado pelo *XML schema* referente a esse tipo de mensagem. Esse processamento acontece depois que os dados que compõem a mensagem da nota fiscal, de acordo com manual de integração do SEFAZ, são lidos da planilha existente no caminho indicado pelo usuário.

Depois que esses dados são carregados, acontece o *parsing* que irá transformar os dados lidos em um arquivo xml.

Caso o ator desse caso de uso seja o servidor de testes, o processamento se resume à validação do xml recebido.

Exportar nota em formato xml- Essa funcionalidade permite que o usuário salve localmente o arquivo xml resultado do *parsing* realizado no processamento da nota fiscal eletrônica.

Enviar mensagem para webservice no servidor- Essa funcionalidade permite que o usuário encapsule o xml resultante do processamento da nota fiscal

em um pacote SOAP, e o envie como parâmetro para o webservice ativo em um servidor de testes local.

Receber requisições de chamadas webservices- Essa funcionalidade garante o recebimento da mensagem por parte do servidor de testes, o processamento dessa mensagem, e o envio da mensagem de retorno referente ao serviço chamado. Essa mensagem de retorno pode ser um erro ou um aviso de sucesso, quer irá no formato xml, também dentro de um pacote SOAP.

Processar consulta de nota fiscal eletrônica- Essa funcionalidade garante a geração do xml referente ao pedido de consulta de uma nota fiscal eletrônica enviada previamente, validado pelo *XML schema* referente a esse tipo de mensagem.

Caso o ator desse caso de uso seja o servidor de testes, o processamento se resume à validação do xml recebido.

Processar consulta de status de nota fiscal eletrônica- Essa funcionalidade garante a geração do xml referente ao pedido de consulta do status do processamento uma nota fiscal eletrônica enviada previamente, validado pelo *XML schema* referente a esse tipo de mensagem. Esse tipo de consulta é necessário porque o webservice do servidor não retorna um resultado imediato quando recebe uma nova nota fiscal eletrônica.

Caso o ator desse caso de uso seja o servidor de testes, o processamento se resume à validação do xml recebido.

Processar pedido de cancelamento da nota fiscal eletrônica- Essa funcionalidade garante a geração do xml referente ao pedido de cancelamento de uma nota fiscal eletrônica enviada previamente, validado pelo *XML schema* referente a esse tipo de mensagem.

Caso o ator desse caso de uso seja o servidor de testes, o processamento se resume à validação do xml recebido.

Gerenciar notas fiscais eletrônica- Essa funcionalidade garante a visualização do status das notas fiscais emitidas, bem como o seu status, permitindo que o usuário selecione uma determinada nota para ser salva localmente em um arquivo xml.

A tela principal do aplicativo, mostrada na Figura 15, contém uma tabela responsável pela funcionalidade de gerenciar as notas já emitidas, exibindo-as atreladas a um status.

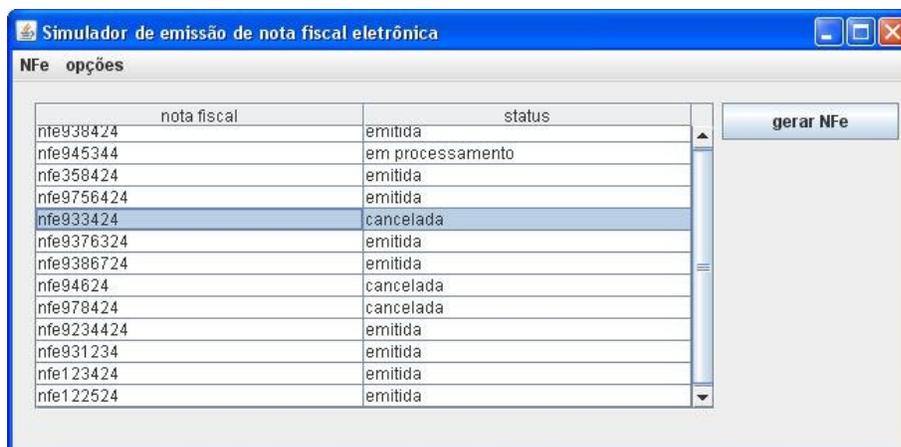


Figura 15. Tela inicial

A tela principal, além de listar todas as NFe emitidas e o seu status de processamento, apresenta um botão que leva para a funcionalidade de geração de um novo xml a ser enviado ao *webservice* indicado.

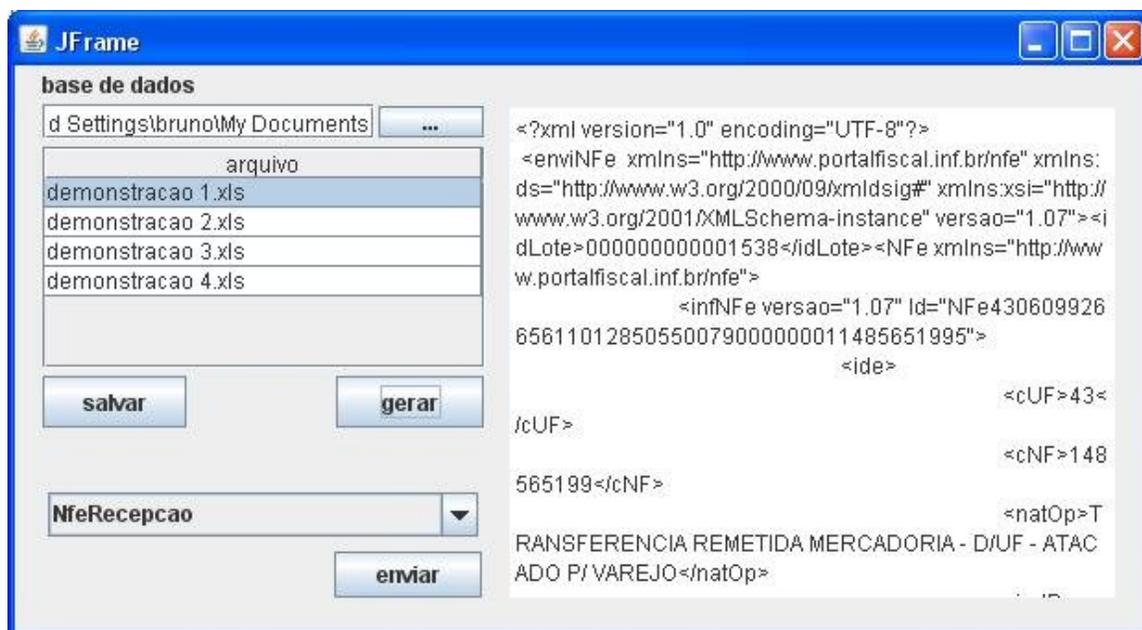


Figura 16. Tela de preparação de mensagens para envio via webservice

A figura 16 mostra a interface gráfica usada para a geração de mensagens XML.

O primeiro passo a ser realizado nessa tela é indicar o diretório referente à base de dados, ou seja, o diretório onde se encontram as planilhas com os dados mínimos para a geração das mensagens. Selecionado o diretório, os arquivos encontrados são listados em um componente *JTable*, no qual pode ser selecionado apenas uma das linhas mostradas. No exato momento em que algum arquivo é selecionado, o botão com a legenda “gerar”, até então desativado, torna-se disponível.

No momento em que o botão “gerar” é pressionado, a planilha é lida, e todo o conteúdo passará por *parsing* em um documento XML no formato previsto, e de acordo com o tipo de servidor a qual ela irá ser submetida, escolhido no *combobox* que pode ser visto na parte inferior esquerda da tela, detalhado na Figura 17.



Figura 17. Combobox com os tipos de mensagens para envio via webservice

Baseado no tipo de serviço escolhido, o sistema irá gerar a mensagem prevista para a ocasião. Terminado esse passo, o sistema dispõe de duas opções: salvar o xml gerado localmente, ou enviar para o servidor de simulação implementado para testes.

A opção de salvar o arquivo localmente funciona como uma forma de importar os dados no formato final, pronto para ser submetido. Já a funcionalidade de enviar, irá submeter o conteúdo para um servidor local, sem qualquer validade jurídica, usado unicamente para processar o documento recebido quanto a sua estrutura. Caso o documento recebido pelo servidor atenda aos formatos previstos, ele será processado, e imediatamente será devolvida a mensagem de confirmação e aceitação do conteúdo, como mostrado na Figura 18.



Figura 18. Mensagem retornada pelo servidor de simulação

4 Conclusão

Este trabalho mostrou a viabilidade de se implantar o Projeto NFe, proposto pela Receita Federal, com o uso de aplicativos que auxiliem todos os tipos de empresas no processo de tratamento de dados e geração de notas fiscais eletrônicas.

Além disso, a construção do protótipo de emissão comprova que as tecnologias atuais estão amadurecidas o suficiente para facilitar a construção de aplicações do gênero.

Outro fator de grande importância é o uso de ferramentas gratuitas, tornando o desenvolvimento viável para empresas de pequeno e médio portes, sem grandes custos. Aliado a isso, *frameworks* de desenvolvimento, também gratuitos, ajudam a alavancar o processo da construção do software, garantindo maior abstração e menor tempo de programação.

Todas essas facilidades implicam numa grande oportunidade para empresas de tecnologia da informação que podem se aproveitar de um conceito ainda incipiente no país para ganhar espaço no mercado, com produtos de alta qualidade e baixo custo de desenvolvimento.

Este documento, em conjunto com a demonstração prática, evidencia que a implementação das notas fiscais eletrônicas terá sucesso dentro de poucos anos, em pequenas e grandes empresas, elevando a administração fiscal do país para um patamar menos burocrático e mais robusto.

4.1 Dificuldades Encontradas

Algumas dificuldades foram enfrentadas ao longo do desenvolvimento do trabalho:

- Ausência de qualquer *webservice* ativo para testes, exigindo que as emissões trabalhem apenas com números de CNPJ reais, cadastrados na SEFAZ, com certificado digital emitido não expirado, além de processar vendas de produtos ou serviços, de fato;
- Péssima comunicação com a equipe de desenvolvimento do projeto NFe, não conseguindo respostas via email;
- Comunicação limitada com os telefones disponíveis no Portal da SEFAZ, se resumindo a esclarecimentos sobre o novo modelo de notas fiscais, não apresentando qualquer esclarecimento técnico;
- Tempo de desenvolvimento reduzido, prejudicando o escopo do projeto, visto que existe uma dependência do prazo com a curva de aprendizado dos conceitos e tecnologias.

4.2 Trabalhos Futuros

Alguns trabalhos futuros relacionados a esse projeto consistem em geração de DANFE(Documento Auxiliar da Nota Fiscal Eletrônica), geração de relatórios, melhor interface gráfica, e principalmente, uma interface com a base de dados mais genérica e robusta, contemplando bases reais.

Referências

[APACHE,2008] *Apache POI - Java API To Access Microsoft Format Files*. Site oficial disponível em <http://poi.apache.org/>.

Acessado em junho de 2008.

[CODEHAUS,2008] Site oficial do framework XStream:Xstream, 2007. Disponível em: <http://xstream.codehaus.org/>. Último acesso em junho de 2008.

[CUNHA,2002] CUNHA,D., 2002.*Web Services, SOAP e Aplicações Web*. Disponível em: http://www.oficinadanet.com.br/index.php?acao=colunas_show&id=448. Acesso em: 01 jun. 2008.

[ECLIPSE, 2002] *Visual Editor Project*, site oficial. Disponível em: <http://www.eclipse.org/vep/WebContent/main.php>. Acessado em junho 2008.

Acessado em junho de 2008.

[ECLIPSE,2004]Site oficial da ferramenta Eclipse. Eclipse - plataforma aberta. Disponível em: <http://www.eclipse.org>. Acessado em junho de 2008.

[ENVOI,2008]CodeHaus XFire home: site oficial XFire, 2007.

Disponível em: <http://xfire.codehaus.org/>. Último acesso em junho de 2008.

[ICP-2001] Portal ICP-Brasil. Disponível em: <https://www.icpbrasil.gov.br>.

Último acesso em junho de 2008.

[IRIB,2005] Instituto de Registro Imobiliário Brasileiro: *Cartilha de certificação digital*. Disponível em:

<https://www.oficioeletronico.com.br/Downloads/CartilhaCertificacaoDigital.pdf>.

Último acesso em junho de 2008.

[IWEB,2003] Site i-web: *webservices*.

Disponível em: <http://www.iweb.com.br/iweb/pdfs/20031008-webservices-01.pdf>.

Acesso em: 01 jun. 2008.

[LUHN,1953] IBM:*Communications of the ACM*. Disponível em: <http://www.ibm.com>

Último acesso em junho de 2008.

[MP, 2001] Receita Federal: *MEDIDA PROVISÓRIA No 2.200-2, DE 24 DE AGOSTO DE 2001*. Disponível em: http://www.receita.fazenda.gov.br/Acsrf/MP_ICP_22002.pdf

Acesso em junho de. 2008.

- [REENSKAUG,1979] Trygve Reenskaug, *MODELS - VIEWS - CONTROLLERS*. Technical note, Xerox PARC.
- [SEFAZ,2007]Portal Nacional da Nota Fiscal Eletrônica, *Manual de integração com o contribuinte*, versão 2.0.4, 2007. Disponível em: <http://www.nfe.fazenda.gov.br/portal/docs/>. Último acesso em junho de 2008.
- [SEFAZ,2005]Portal Nacional da Nota Fiscal Eletrônica, *Projeto conceitual do Sistema NFe*, versão 22_7_06, 2006. Disponível em: <http://www.nfe.fazenda.gov.br/portal/diversos.aspx>. Último acesso em junho de 2008.
- [SEFAZ,2005]Portal Nacional da Nota Fiscal Eletrônica, *Manual de utilização da Sefaz Virtual do ambiente nacional*, 2006. Disponível em:<http://www.nfe.fazenda.gov.br/portal/diversos.aspx>. Último acesso em junho de 2008.
- [SUN,1995]Site Oficial da Tecnologia Java: *The Source for Java Developers*. Disponível em: <http://java.sun.com>. Acessado em junho 2008.
- [W3C,1990]*Extensible Markup Language (XML)*: W3C recommendation Disponível em: <http://www.w3c.org>. Acesso em: 01 jun. 2008.
- [W3C,1995] *HTML*: W3C activity Disponível em: <http://www.w3.org/MarkUp>. Acesso em: 01 jun. 2008.
- [W3C,2002] Webservice: *W3C web services activity* Disponível em: <http://www.w3.org/2002/ws/> . Acesso em: 01 jun. 2008.
- [W3C,2004] *SOAP*: W3C activity Disponível em: <http://www.w3.org/TR/soap> . Acesso em: 01 jun. 2008.
- [WIKIPEDIA,2008]Wikipedia, the Free Encyclopedia: DTD, 2008. Disponível em: http://pt.wikipedia.org/wiki/Document_Type_Definition. Último acesso em junho de 2008.
- [X.509,1988] X.509: *Internet X.509 Public Key Infrastructure Certificate Management Protocols*. Disponível em: <http://www.faqs.org/rfcs/rfc2510.html>. Acesso em: 01 jun. 2008.

Apêndice A

Schema XML: xmldsig-core-schema_v1.01.xsd

#	Campo	Ele	Pal	Tipo	Ocor.	Tam.	Dec.	Descrição/Observação
XS01	Signature	Raiz	-	-	-	-		
XS02	SignedInfo	G	XS01	-	1-1			Grupo da Informação da assinatura
XS03	CanonicalizationMethod	G	XS02	-	1-1			Grupo do Método de Canonicalização
XS04	Algorithm	A	XS03	C	1-1			Atributo Algorithm de CanonicalizationMethod: http://www.w3.org/TR/2001/REC-xml-c14n-20010315
XS05	SignatureMethod	G	XS02	-	1-1			Grupo do Método de Assinatura
XS06	Algorithm	A	XS05	C	1-1			Atributo Algorithm de SignedMethod: http://www.w3.org/2000/09/xmldsig#rsa-sha1
XS07	Reference	G	XS02	-	1-1			Grupo de Reference
XS08	URI	A	XS07	C	1-1			Atributo URI da tag Reference
XS10	Transforms	G	XS07	-	1-1			Grupo do algorithm de Transform
XS11	unique_Transf_Algorithm	RC	XS10	-	1-1			Regra para o atributo Algorithm do Transform ser único.
XS12	Transform	G	XS10	-	2-2			Grupo de Transform
XS13	Algorithm	A	XS12	C	1-1			Atributos válidos Algorithm do Transform: http://www.w3.org/TR/2001/REC-xml-c14n-20010315 http://www.w3.org/2000/09/xmldsig#enveloped-signature
XS14	XPath	E	XS12	C	0-N			XPath
XS15	DigestMethod	G	XS07	-	1-1			Grupo do Método de DigestMethod
XS16	Algorithm	A	XS15	C	1-1			Atributo Algorithm de DigestMethod: http://www.w3.org/2000/09/xmldsig#sha1
XS17	DigestValue	E	XS07	C	1-1			Digest Value (Hash SHA-1 – Base64)
XS18	SignatureValue	G	XS01	-	1-1			Grupo do Signature Value
XS19	KeyInfo	G	XS01	-	1-1			Grupo do KeyInfo
XS20	X509Data	G	XS19	-	1-1			Grupo X509
XS21	X509Certificate	E	XS20	C	1-1			Certificado Digital x509 em Base64

Apêndice B

Característica	Descrição
Web Services	Padrão definido pelo WS-I Basic Profile 1.1 (http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html).
Meio lógico de comunicação	Web Services, disponibilizados pelo Portal da Secretaria de Fazenda Estadual.
Meio físico de comunicação	Internet
Protocolo Internet	SSL versão 3.0, com autenticação mútua através de certificados digitais.
Padrão de troca de mensagens	SOAP versão 1.2.
Padrão da mensagem	XML no padrão Style/Encoding: Document/Literal, wrapped.
Padrão de certificado digital	<p>X.509 versão 3, emitido por Autoridade Certificadora credenciada pela Infra-estrutura de Chaves Públicas Brasileira – ICP-Brasil, do tipo A1 ou A3, devendo conter o CNPJ do proprietário do certificado digital.</p> <p>Para assinatura de mensagens, utilizar o certificado digital do estabelecimento matriz ou do estabelecimento emissor da NF-e.</p> <p>Para transmissão, utilizar o certificado digital do responsável pela transmissão.</p>
Padrão de assinatura digital	XML Digital Signature, Enveloped, com certificado digital X.509 versão 3, com chave privada de 1024 bits, com padrões de criptografia assimétrica RSA, algoritmo message digest SHA-1 e utilização das transformações Enveloped e C14N.
Validação de assinatura digital	Será validada além da integridade e autoria, a cadeia de confiança com a validação das LCRs.
Padrões de preenchimento XML	<ul style="list-style-type: none"> • Campos não obrigatórios do Schema que não possuam conteúdo terão suas tags suprimidas no arquivo XML. • Máscara de números decimais e datas estão definidas no Schema XML. • Nos campos numéricos inteiro, não incluir a vírgula ou ponto decimal. • Nos campos numéricos com casas decimais, utilizar o “ponto decimal” na separação da parte inteira.

Assinaturas

Este Trabalho foi apresentado à graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de bacharel do aluno Bruno Rodrigues.

Fernando da Fonseca de Souza (orientador)

Bruno Rodrigues (aluno)