Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JOÃO PEDRO SACHETI

PROCESSO DE EMISSÃO DE NOTA FISCAL ELETRÔNICA

Assis/SP
2017

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JOÃO PEDRO SACHETI

PROCESSO DE EMISSÃO DE NOTA FISCAL ELETRÔNICA

Projeto de pesquisa apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): João Pedro Sacheti
Orientador(a): Prof. Ms. Fernando César de Lima

Assis/SP
2017

FICHA CATALOGRÁFICA

SACHETI, João Pedro.
Processo de emissão de nota fiscal eletrônica / João Pedro Sacheti. Fundação Educacional do Município de Assis –FEMA – Assis, 2017.
Número de páginas.

1. Nota Fiscal eletrônica. 2. XML. 3. Certificado Digital. 4. SOAP. 5. Webservice

CDD:
Biblioteca da FEMA

PROCESSO DE EMISSÃO DE NOTA FISCAL ELETRÔNICA

JOÃO PEDRO SACHETI

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Professor Mestre Fernando César de Lima

Examinador: _____
Professor Guilherme de Cleva Farto

Assis/SP
2017

DEDICATÓRIA

Dedico este trabalho a todos os seres deste planeta. Que eles possam alcançar um estado puro de perfeita harmonia e equilíbrio.

“Toda grande caminhada começa com um simples passo. ” – Buda Shakyamuni

RESUMO

É sabido que parte da rotina de uma empresa regular no Brasil consiste na emissão de notas fiscais. Desde 2005 regulamentada pelo Ajuste Sinief 07/05, a Nota Fiscal eletrônica – Modelo 1A surgiu como uma forma de garantir segurança, integridade e simplicidade para o contribuinte e o Estado.

Durante a produção deste trabalho, foi elaborada uma pesquisa exploratória sobre as rotinas de validação, preenchimento, emissão e autorização de notas fiscais eletrônicas e seus respectivos procedimentos peculiares, de forma a familiarizar desenvolvedores que desejam implementar softwares de emissão de nota fiscal eletrônica

Palavras-chave: Nota fiscal eletrônica, XML, documento fiscal, legislação tributária, certificado digital, criptografia, SOAP, webservices

ABSTRACT

It is known that part of a regular Brazilian company is the emission of fiscal documents such as the “Nota Fiscal eletrônica”. Since 2005 regulated by the SINIEF document 07/05, the “Nota Fiscal eletrônica – Model 1A” was made to ensure safety, integrity and simplicity to the taxpayers and the State.

During the production of this article, it was made an exploratory research about the routines of validation, filling, emission and validation of the fiscal documents and its respective peculiar procedures, so that other developers may use this as a reference for implementing their own software for emitting such fiscal documents.

Keywords: Nota fiscal eletrônica, XML, fiscal document, tributary legislation, digital certificate, cryptography, SOAP, webservices

SUMÁRIO

1. INTRODUÇÃO

A nota fiscal é um documento que é emitido entre os pares de uma operação comercial tributável com a discriminação de todas as partes, mercadorias e outros detalhes pertinentes a situação. Com a modernização da tecnologia e a necessidade de ter um controle mais rígido e seguro sobre as notas fiscais, o Conselho Nacional de Política Fazendária (Confaz) emitiu em 2005 o Ajuste Sinief 07/05 que regulamentava a emissão de um novo tipo de documento fiscal, a Nota Fiscal Eletrônica.

A nota fiscal eletrônica foi elaborada como um documento em formato XML, assinado digitalmente, que possui validade jurídica e tem como objetivo principal substituir as notas fiscais de papel, que são sujeitas a perdas, desfigurações, destruição e ocupam muito espaço nos gabinetes de escritórios de contabilidade de todo o país.

Como forma de auxiliar a comunidade de desenvolvedores, esta pesquisa exploratória explica de forma simples e objetiva como funciona o processo de emissão de nota fiscal eletrônica no leiaute contemporâneo a essa pesquisa e exemplificar esse processo por meio da elaboração de uma simulação da emissão de uma nota completa, usando o ambiente de homologação da Secretaria da Fazenda do estado de São Paulo e um pequeno software desenvolvido em Java.

2. JUSTIFICATIVA

Atualmente os desenvolvedores nacionais passam por uma dificuldade constante devido à ausência de esclarecimentos no processo de emissão da nota fiscal eletrônica, seja por deficiências tanto de conhecimento fiscal quanto das ferramentas e técnicas empregadas na emissão do documento fiscal.

Este trabalho se apoia na necessidade de suprir essa deficiência oferecendo uma explicação conceitual, mais abrangente e unificada tanto dos padrões de emissão quanto das ferramentas utilizadas na transmissão deste documento fiscal.

3. PROPOSTA DE TRABALHO

Durante a pesquisa e elaboração deste trabalho, são explorados os conceitos de webservice, preenchimento do documento fiscal, assinaturas digitais utilizando certificados no padrão ICP-BRASIL, construção do arquivo de XML com o uso da especificação JAXB, validação e assinatura digital deste arquivo XML, transmissão dos dados usando *Secure Socket Layer* e *webservices* SOAP fornecidos pela Secretaria da Fazenda autorizadora e aferimento da autorização da emissão do documento fiscal, bem como seu documento auxiliar de nota fiscal eletrônica (DANFE). Como forma de exemplificar os conceitos aqui abordados, um programa anexo será desenvolvido de forma a auxiliar e abordar de forma prática todo ou partes do processo de emissão do documento fiscal, entretanto, tal programa tem como objetivo somente servir de exemplo e não como uma biblioteca que pode ser inclusa em um projeto. Não é do escopo deste trabalho especificar e detalhar as regras fiscais e contábeis relativas ao preenchimento da nota fiscal eletrônica, mas sim servir de base para implementação de sistemas compatíveis com a emissão da nota fiscal eletrônica. Também não serão abordados os temas de contingência, cancelamento e inutilização de nota fiscal eletrônica, limitando-se a tratar somente do processo de autorização de um lote de nota fiscal eletrônica.

4. REFERENCIAL BIBLIOGRÁFICO

Segundo o manual de orientação do contribuinte uma nota fiscal eletrônica é um documento digital que é juridicamente validado por uma assinatura digital e um protocolo de autorização. O objetivo deste documento fiscal é discriminar circulação de mercadorias ou serviços. (RECEITA FEDERAL, 2015)

Outros autores (PEREIRA; LOCKS, 2008, p. 7) esclarecem que é objetivo da nota fiscal eletrônica substituir a emissão de notas fiscais em papel e simplificar o processo fiscal das empresas brasileiras

Uma das principais tecnologias utilizadas para a emissão de uma nota fiscal eletrônica é aa *Extensible Markup Language*, o XML. As consultas feitas em artigos deixam

explícitos também a sua importância para na internet por ser um meio uniforme de compartilhar (SAYIH; BRÜGGEMANN-KLEIN; 2016) e armazenar (FAN et al, 2016) dados entre aplicações Web. Trata-se de um padrão desenvolvido pela World Wide Web Consortium (W3C) para a criação de documentos humanamente legíveis e com interoperabilidade com outros formatos como o HTML (*Hyper Text Markup Language*) (BRAY et al, 2008).

Para garantir a segurança e integridade dos dados contidos num XML, utiliza-se um Certificado digital. Eles são uma forma de garantir a credibilidade em sistemas de acesso restrito onde é necessário autenticação bilateral por meio de credenciais digitais (SEAMONS et al., 2002). Podem também ter a funcionalidade de servir como base para funções matemáticas que criptografam informações ou garantir que a informação não foi adulterada durante o transporte (MERKLE, 1990)

A transmissão dos dados para a Secretaria da Fazenda é feita por meio de um *Web Service*. Um *Web Service*, segundo definição da W3C, é um sistema designado para suportar a comunicação entre máquinas de forma neutra com uma interface que é processável por um programa de modo que as operações realizadas entre um cliente hipotético não dependam da plataforma de um servidor hipotético. (BOOTH et al., 2004)

Um de seus usos é o pedido de execução de uma determinada atividade de um servidor por um cliente utilizando como canal de comunicação o protocolo HTTP e o uso de XML(SUDA, 2003)

5. FUNDAMENTAÇÃO TEÓRICA

Para iniciar os estudos delineados nessa pesquisa exploratória, precisamos primeiramente estabelecer uma fundamentação teórica básica em alguns assuntos vitais para o entendimento do processo de autorização de uma nota fiscal eletrônica.

5.1. XML – EXTENSIBLE MARKUP LANGUAGE

A *Extensible Markup Language*, comumente abreviado para XML, é uma especificação elaborada em 1996 pela W3C (*World Wide Web Consortium)*, um órgão internacional que

escreve padronizações para as tecnologias básicas usadas na web. Essa especificação descreve uma linguagem simples para a escrita de documentos que são humanamente legíveis ao passo que podem ser interpretados por algoritmos computacionais para a extração de dados contidos no documento (BRAY et. al, 2008). O quadro 1 exibe um exemplo de arquivo XML:

```
<pessoa id=”5”>
        <nome>João Pedro Sacheti</nome>
        <nascimento>07/02/1996</nascimento>
</pessoa>
```

Quadro 1 - Exemplo de arquivo XML

Ao escrever um arquivo XML, as informações são inseridas dentro de *tags* que podemos chamar de “elementos”. O elemento é a demarcação de conteúdo do arquivo XML. Todo elemento deve receber uma rotulação que deve estar entre os operadores “menor que” (<) e “maior que” (>). Ao criar um elemento podemos informar o conteúdo textual ou outros elementos a fim de descrever a informação que se deseja. Ao finalizar a escrita dessas informações, deve-se fechar o elemento correspondente por meio da inserção de outro elemento com os operadores “menor que” e “barra” (</) junto com o operador “maior que” (>) de modo que os elementos correspondentes formam um par que indica o começo e outro que indica o fim. Tais elementos também podem ter atributos próprios no formato chave-valor. No quadro 1, o elemento “pessoa” possui o atributo “id” de valor 5.

Arquivos XML seguem algumas diretrizes básicas quanto a qualidade de sua estrutura: para ser considerado correto um arquivo XML deve ser bem formatado e opcionalmente válido. Um arquivo é considerado bem formatado se seguir as seguintes regras:

- Possuir um elemento raiz, isto é, um elemento que não está contido dentro de nenhum outro.
- Os elementos possuem início e fim
- Todos os valores de atributo estão inseridos com aspas

O quadro 2 ilustra um arquivo XML que não está formatado:

```
<mensagem>Bom dia! </teste>
```

Quadro 2 - Exemplo de XML mal formatado: elemento mensagem é aberto, mas não é fechado

Além disso, um XML pode estar opcionalmente associado a um arquivo de XSD (*XML Schema Definition*, definição de esquema XML). Um arquivo XSD é um metadado que especifica a ordem, elementos e atributos válidos, formatos de dados suportados entre outros e serve para garantir que um XML está válido. Ao ser avaliado por um processador de XML, um arquivo XML primeiro é inspecionado quanto a sua formatação: caso haja erros de formatação, o processador de XML interrompe seu processo e rejeita o arquivo de entrada. Caso o arquivo esteja bem formatado e associado a um esquema, o processador fará a avaliação das regras contidas dentro do esquema em contraste com a estrutura do XML informado e então aceitará ou rejeitará o arquivo.

5.2. LINGUAGEM DE PROGRAMAÇÃO JAVA

O Java é uma popular linguagem de programação orientada a objetos com suporte a concorrência, baseada em classes e amplamente utilizada para desenvolvimento de aplicações corporativas. Foi criada em 1996 por James Gosling, então funcionário da extinta Sun Microsystems.

Java foi criado com a arquitetura *Write once, run everywhere,* pois funciona utilizando uma máquina virtual: um software que interpreta as instruções dentro de um executável Java e executa essas instruções nativamente dentro do sistema operacional nativo. É também uma linguagem de alto nível que utiliza referências vinculadas a um *Garbage Collector*, um software que funciona dentro da máquina virtual Java e coleta referências que não são mais utilizadas durante a execução de um programa (GOSLING et al., 2015). Além disso, Java possui uma tipagem estritamente forte e estática e consegue detectar a maior parte de seus erros em tempo de compilação: tal característica combinada com a ausência de outras características inseguras (como index negativo em vetores, que cria comportamento imprevisível em certos softwares) faz do Java ser a linguagem de escolha para o desenvolvimento deste trabalho.

5.2.1. APACHE AXIS2

O Apache Axis2 é um software livre oferecido pela Apache Software Foundation, uma fundação sem fins lucrativos que mantém diversos projetos com uma vasta comunidade de desenvolvedores no mundo todo.

O Apache Axis2 é utilizado para gerenciar a comunicação entre *web services* de forma simples dentro da plataforma Java. Com ele é possível gerar uma classe Java que abstrai o comportamento de um cliente de web service tal que o usuário apenas foque na regra de negócio da aplicação que receberá as informações via web service.(GRAHAM et al., 2004)

5.2.2. JAVA ARCHITECTURE FOR XML BINDING – JAXB

A JAXB, sigla para *Java Architecture for XML Binding*, é uma tecnologia da plataforma Java delineada na especificação JSR-222 em 2009. De acordo com essa especificação, é possível fazer o mapeamento de esquemas XML para classes Java com tipos compatíveis, e vice-versa.(KAWAGUCHI; VAJJHALA; FIALLI, 2009)

Essa especificação também define um modelo unificado para bibliotecas que desejam implementar ferramentas de manipulação, leitura e criação de XML dentro da plataforma Java. Tal especificação conta com uma implementação de referência que se encontra na biblioteca padrão da linguagem Java desde sua versão 6.

Dentro da especificação, são definidas quatro ferramentas básicas:

- *Marshaller*: é a ferramenta responsável por transformar uma instância de classe Java em um XML compatível
- *Unmarshaller*: é a ferramenta responsável por transformar um XML com uma classe Java compatível
- *Binder*: cria um vínculo entre uma instância de classe Java com um arquivo XML de modo que possam ser validados e manipulados de forma simétrica
- *Schema compiler* e *Schema generator:* são um par de ferramentas que compilam um esquema para classes Java ou transpilam uma classe Java para um esquema XML, respectivamente.

5.2.3. Java Database Connectivity – JDBC

A JDBC (*Java Database Connectivity)* é uma especificação que surgiu inicialmente em 1997 para suprir a necessidade dos desenvolvedores Java acessarem bancos de dados de forma programática, neutra e consistente com o padrão SQL. Ela oferece as interfaces básicas de comunicação entra a aplicação, a plataforma Java e o banco de dados em questão por meio de um driver implementado pelo banco de dados que se deseja usar (ANDERSEN, 2011). Esse formato garante que a JDBC possa se comunicar com qualquer banco de dados sem ter que alterar sua estrutura interna, pois todos os detalhes da implementação e características únicas de um determinado sistema de banco de dados sejam implementados por seus drivers, cabendo a JDBC apenas oferecer uma interface para que o desenvolvedor que utiliza o banco de dados em questão escreva suas consultas por meio de um programa escrito em Java, garantindo uma integração transparente entre os dados e a aplicação.

5.3. POSTGRESQL 9.6

O banco de dados relacional PostgreSQL é um projeto open-source mantido por sua comunidade, originalmente baseado no POSTGRES, um banco de dados desenvolvido internamente na Universidade de Berkley. Sua especificação é compatível com o padrão de consultas SQL (*Structured Query Language*) com suporte a diversos recursos modernos, como consultas complexas, integridade transacional, triggers, funções, agregações, etc.

Foi em 1996 que o projeto POSTGRES se transformou no projeto PostgreSQL, passando por uma reformulação tanto de nome, versionamento e renovação no time de desenvolvimento. A partir daí, passou a integrar novos recursos tanto no seu servidor de banco de dados como no cliente que realiza as consultas, se tornando o renomado banco de dados dos tempos atuais.(THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2017)

5.4. ASSINATURA DIGITAL

A assinatura digital é um meio de garantir a integridade e autenticidade de um arquivo que está assinado. Esse objetivo é adquirido por meio de um certificado digital, que contém informações de uma entidade, uma chave privada criptográfica e uma chave pública criptográfica.

Ao assinar um arquivo, cria-se um *Hash*, que é um “resumo” dos dados binários contidos no arquivo. A chave privada então é utilizada para digerir esse *Hash* e gerar uma informação semelhante, porém contendo os dados de quem assinou. *Hash* e *Digest* (a digestão do Hash, ou seja, a assinatura) são anexados ao arquivo original junto com os dados de quem gerou, além da data de geração da assinatura.

Quando o receptor receber o arquivo assinado digitalmente, ele poderá gerar o mesmo *Hash* e comparar com o *Digest* do arquivo por meio da chave criptográfica pública. Caso os dados não coincidam, significa que o arquivo foi adulterado e, portanto, sua assinatura é invalidada. Caso os dados coincidam, o receptor tem a garantia de integridade e autenticidade do documento assinado, permitindo então a irretratabilidade do arquivo e conferindo-lhe condição de legítimo.

5.4.1. Assinatura digital de arquivos XML

A assinatura digital em arquivos XML funciona de forma muito semelhante à de outros arquivos e é utilizada também para garantir a integridade e autenticidade do arquivo XML assinado. Essa técnica utiliza um padrão criado pela W3C que especifica elementos especiais que conterão os valores da assinatura digital e do *Hash* do elemento a ser assinado. Chamamos essa técnica de “Assinatura envelopada”, pois ela é inclusa dentro do próprio arquivo a ser assinado. (EASTELAKE et al., 2013)

Para a criação de uma assinatura digital de um arquivo XML, são necessárias as seguintes etapas:

1. Definir o elemento a ser assinado digitalmente e identifica-lo por meio de uma URI (*Uniform Resource Identifier*)
2. Normalizar os dados inseridos dentro do elemento por meio de transformadores. As informações sobre esses transformadores serão inclusas num elemento especial junto com a assinatura

3. Calcular o *hash* do elemento a ser assinado e já normalizado
4. Inserir o valor deste hash em formato texto dentro de um elemento especial chamado “*Reference*”
5. Aplicar uma assinatura no hash gerado e inserir as informações relativas ao algoritmo utilizado para a criação da assinatura em um elemento para este fim.
6. Inserir o binário da assinatura em formato texto dentro de um elemento para este fim, “SignatureValue”
7. Incluir os dados relativos ao certificado utilizado para a assinatura, como dados do proprietário e chave pública
8. Agrupar todos os novos elementos gerados relativos a assinatura em um único elemento, o “Signature”.
9. Inserir o elemento contendo todos os dados da assinatura, seu hash e valor da assinatura na estrutura do arquivo assinado. (GERIC; VIDACIC, 2012)

O quadro 3 ilustra a estrutura básica dos elementos utilizados numa assinatura digital em arquivo XML:

```
<Signature>
  <SignedInfo>
    <CanonicalizationMethod />
    <SignatureMethod />
    <Reference>
      <Transforms>
      <DigestMethod>
      <DigestValue>
    </Reference>
    <Reference />
  </SignedInfo>
  <SignatureValue />
  <KeyInfo />
</Signature>
```

Quadro 3 - Estrutura dos elementos da Assinatura Digital

6. DESENVOLVIMENTO DO TRABALHO DE EXEMPLO

Para fundamentar e exemplificar o processo de emissão, um aplicativo Java será desenvolvido com a seguinte estrutura:

1. Classes de modelo (*beans*): serão responsáveis por armazenar as informações em memória de objetos da vida real: nota fiscal, produto, cliente, emitente, etc.
2. Classes de validação: serão responsáveis por garantir a integridade dos dados antes que eles possam ser enviados para a secretaria da Fazenda
3. Classes de geração do arquivo XML: serão responsáveis por coletar as informações e montar o arquivo XML correspondente ao processo corrente
4. Classes de emissão do documento XML: serão responsáveis por criar a comunicação com o servidor da secretaria da Fazenda, enviar o arquivo gerado e então receber uma resposta do servidor.
5. Classes de análise de resultado: serão responsáveis por analisar o tipo de resposta da Secretaria da Fazenda, seja uma resposta de autorização ou um erro no processo de emissão
6. Classe de controle de processo: será responsável por orquestrar e organizar o funcionamento do processo de modo que ele ocorra na ordem correta, e que, caso haja um erro, o programa consiga se recuperar e reportar ao usuário o problema ocorrido.

Tal aplicação será disponibilizada na plataforma GitHub e licenciada sob os termos da licença Apache 2.0. Essa licença permitirá que futuros desenvolvedores usem o software de maneira permissiva, mas que exige a declaração dos autores originais e demais propriedades intelectuais.

7. TRABALHOS FUTUROS

Como parte dos trabalhos futuros, poderão ser elaborados novas implementações que se adequem as futuras versões da nota fiscal eletrônica, bem como uma API completa

que simplifique o trabalho dos desenvolvedores além de mantê-la compatível com as futuras especificações emanadas pelo Ministério da Fazenda.

8. REFERENCIAS

ANDERSEN, L. (ORACLE C. **JSR 211: JDBC 4.1 Specification**. Disponível em: <http://download.oracle.com/otn-pub/jcp/jdbc-4_1-mrel-spec/jdbc4.1-fr-spec.pdf>. Acesso em: 5 mar. 2017.

BOOTH, D. et al. Web Services Architecture. **W3C Note**, v. 22, n. February, p. 1--98, 2004.

BRASIL. **Sistema Nota Fiscal Eletrônica Manual de Orientação do Contribuinte** Brasília, Receita Federal, , 2015. Disponível em: <http://www.nfe.fazenda.gov.br/portal/exibirArquivo.aspx?conteudo=URCYvjVMIzI=>

EASTELAKE, D. et al. **XML Signature Syntax and Processing Version 1.1**. Disponível em: <https://www.w3.org/TR/2013/REC-xmldsig-core1-20130411/>. Acesso em: 5 mar. 2017.

FAN, H. Distributed XPath Query Processing over Large XML Data based on MapReduce framework. p. 1447–1453, 2016.

GERIC, S.; VIDACIC, T. XML digital signature and its role in information system security. **The 35th International Convention MIPRO**, p. 1520–1525, 2012.

GOSLING, J. et al. **The Java ® Language Specification Java SE 8 Edition**. Disponível em: <https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf>. Acesso em: 19 fev. 2017.

GRAHAM, S. et al. **Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI**. Pearson Education, 2004.

KAWAGUCHI, K.; VAJJHALA, S.; FIALLI, J. **The Java Architecture for XML Binding**. Santa Clara, CA., 2008

MERKLE, R. C. A certified digital signature. **Advances in Cryptology—CRYPTO'89 Proceedings**, p. 218–238, 1990.

PEREIRA, S.; LOCKS, R. Governança Eletrônica na administração pública: estudo de caso sobre a nota fiscal eletrônica–NF-e. **18º Congresso Brasileiro de Contabilidade**, p.

1–12, 2008.

SAYIH, M.; BRÜGGEMANN-KLEIN, A. Using XML Technology in the Development Process of Web Applications. p. 89–93, 2016.

SEAMONS, K. E. et al. **Requirements for Policy Languages for Trust Negotiation -- Seamons Winslett.pdf**. Third international Workshop on Policies for Distributed Systems and Networks. **Anais**...IEEE Computer Society, 2002

SUDA, B. **SOAP Web Services**. University of Edumburgh, 2003.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL 9.6.2 Documentation**. Disponível em: <https://www.postgresql.org/docs/9.6/static/index.html>. Acesso em: 5 mar. 2017.