

**EMISSOR DE CUPOM FISCAL DE SERVIÇO POR MEIO  
DE PONTO DE ATENDIMENTO**

Paulo Henrique de Medeiros

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
[paulomedeiros0212@gmail.com](mailto:paulomedeiros0212@gmail.com)

Francisco Assis Moreira do Nascimento

Professor Orientador

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
[assis@faccat.br](mailto:assis@faccat.br)

**Resumo**

Este artigo tem por objetivo apresentar um equipamento, que possibilita melhorias no processo de recolhimento de imposto, mais especificamente, do ISSQN (Imposto Sobre Serviço de Qualquer Natureza) por parte dos prestadores de serviço, incluindo empresas e profissionais autônomos. A ferramenta desenvolvida consiste da combinação de software (programa de computador) e hardware (equipamento), visando tornar mais rápida a emissão de notas fiscais de serviços eletrônicas e atribuir maior dinamismo ao processo, que muitas vezes é demorado e complexo. Com a utilização do protótipo desenvolvido e apresentado neste artigo, é possível emitir os documentos fiscais pertinentes, sem a necessidade de utilização de um computador, bastando apenas se possuir acesso à internet. Outra vantagem advinda da ferramenta é a simplificação da integração de documentos eletrônicos por ERP's de terceiros.

**Palavras-chave:** cupom fiscal, ISSQN, nota fiscal de serviço eletrônica.

**COUPON ISSUING FISCAL SERVICE IN POINT OF ATTENDANCE**

**Abstract**

*This paper has as objective to present an equipment oriented to improve the tax collecting process, more specifically, the collecting of a brazilian tax, called ISSQN (Imposto Sobre Qualquer Natureza), that should be paid by service providers, including companies and self-employed professionals. The developed tool consists of a combination of software (computer program) and hardware (mobile equipment), oriented to speed up the issuing of electronic invoices and give greater dynamism in the process, which is often time consuming and complex. Using the prototype developed and presented in this article, you can issue the relevant tax documents, without the need to use a computer, by simply having an internet access. Another advantage of the tool is due to the simplification of the integration of electronic documents by ERP systems.*

**Key-words:** fiscal invoice, ISSQN, electronic invoice of services.

## 1 INTRODUÇÃO

Uma das obrigações fiscais dos prestadores de serviços é o Imposto Sobre Serviços de Qualquer Natureza (ISSQN), que deve ser pago para as administrações tributárias municipais (PATROCÍNIO, 2011).

Para informarem os valores dos serviços que prestaram em cada mês, os prestadores de serviço devem entregar para a administração tributária municipal, declarações mensais, reportando cada serviço prestado e o correspondente valor de ISSQN a ser pago, ou emitir uma Nota Fiscal de Serviços eletrônica (NFS-e), em que constam estas mesmas informações, para cada tomador de serviço e, ao final do mês, gerar uma guia de recolhimento para pagamento do ISSQN devido referente a todas as NFS-es emitidas.

Com a crescente modernização das administrações tributárias, que priorizam as automatizações de todos os processos, a maioria dos municípios está passando a permitir o uso apenas de Notas Fiscais de Serviços eletrônicas como meio de informar aos municípios sobre os serviços prestados e o correspondente ISSQN a ser pago.

É dentro deste contexto que o presente trabalho propõe o desenvolvimento de uma ferramenta de emissão de cupom fiscal de serviços eletrônico para prestadores de serviço, que possuem grande fluxo de emissão de documentos eletrônicos (por exemplo, administradoras de estacionamentos, empresas de cópias, etc) ou qualquer outra empresa que opte por não utilizar um computador para emitir NFS-e.

A ferramenta desenvolvida combina componentes de hardware – equipamento emissor de cupom fiscal de serviços eletrônicos - e de software – portal de serviços Web para integração com municípios -, que automatiza e facilita o cumprimento da obrigação fiscal municipal ISSQN pelos prestadores de serviços.

Com o processo de modernização da administração tributária dos Municípios, a maior parte dos contribuintes será obrigada a aderir à emissão de NFS-e. Muitas dessas empresas não estão preparadas para essa modernização e acabam enfrentando grandes dificuldades nas mudanças internas para atender esta obrigatoriedade fiscal. Além desta mudança, normalmente, esses processos podem gerar um custo considerável na compra dos equipamentos necessários (computador, impressora e acesso à internet). Outro fator de impacto negativo para adesão ao sistema da NFS-e é a falta de familiaridade com a informática de determinados prestadores de serviço (por exemplo, mecânicas, empresas de instalações elétricas, encanadores e etc).

Muitos desses prestadores possuem um volume considerável de prestação de serviço durante o expediente, pois a emissão de NFS-e pode impactar na velocidade do atendimento, devido à exigência do preenchimento de diversos campos solicitados no site colocado à disposição por cada Município.

Ademais, cada empresa responsável pela sistematização da emissão de NFS-e de cada Município, normalmente, possui seu próprio padrão de conectividade, causando um enorme retrabalho para os responsáveis pelos ERP's dos prestadores de serviços, pois precisam dedicar horas de seus programadores para realizar a integração exclusiva de seu ERP para cada Prefeitura.

Esta pesquisa propõe uma solução de baixo custo e de rápida e fácil utilização para os prestadores de serviços emitirem seus documentos fiscais, podendo submetê-los para os Municípios em que atuam, sem a necessidade de um grande esforço para adequação de seus ERP's.

Diante do avanço da tecnologia, todos os Municípios que ainda não aderiram ao sistema da NFS-e, inevitavelmente, em um futuro próximo passarão a realizar a implantação deste sistema. A implantação proporciona diversos benefícios para o Município, tais como: a facilidade na fiscalização, controle e gestão, aumento da percepção de risco e, principalmente, a redução de sonegação (SEBRAE RJ).

Além do benefício para o fisco municipal, a adequação ao sistema da NFS-e é vista de forma positiva na maioria das empresas de médio e grande porte, pois simplifica o processo de faturamento e reduz custos, por não precisar de impressão do documento e a nota fiscal poder ser transmitida eletronicamente, não sendo necessário o envio do documento físico por correio, motoboy ou algum outro meio de transporte.

Diante dessa evolução, no setor fiscal, é importante que as empresas de pequeno e médio porte possam aderir a essa nova ferramenta sem impacto em seus processos internos, desse modo, é fundamental que alguma solução ou atitude seja tomada para que estas empresas não sejam prejudicadas.

Neste trabalho, o objetivo geral é desenvolver uma máquina POS (*Point of Sale* – Ponto de Venda) para que qualquer prestador de serviço, sem conhecimentos em informática, consiga operar e emitir seus cupons fiscais de serviços, e, além disso, desenvolver um ambiente Web capaz de receber todos os cupons fiscais de serviços transmitidos pelas máquinas POS e por softwares de terceiros, sendo capaz de se adaptar facilmente com qualquer modo de conectividade dos Municípios no território nacional.

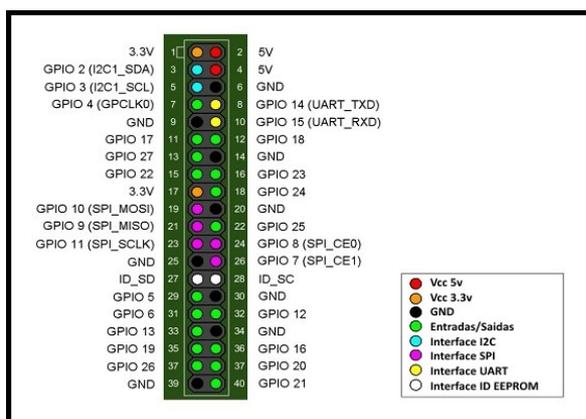
## 1.1 Conceitos e Tecnologias

### 1.1.1 Hardware

Para o desenvolvimento do protótipo foi utilizado o RaspberryPI B+, que segundo RICHARDON e WALLACE (2013) é um microcomputador de baixo custo desenvolvido pela empresa Raspberry. Ele é um computador de baixa capacidade de processamento capaz de rodar sistemas operacionais enxutos, com pouca necessidade de processamento e espaço em disco. Apesar de ser um computador que cabe na palma de uma mão, ele possui interfaces que permitem conectá-lo em monitores, TV's, placas de áudio, teclado e mouse, ou seja, possui os recursos periféricos tradicionais que todo computador possui, porém, vale destacar que ele não possui HD (*Hard Disk*), sendo assim, o armazenando dos dados é realizado em um micro cartão SD.

Outro diferencial deste periférico são os quarenta pinos de conexão, que permitem acoplar facilmente qualquer outro módulo eletrônico por meio de conexões seriais e paralelas.

Figura 1 – Pinagem do Raspberry B+



Fonte: Elaborado por equipe FILIPEFLOP

Conforme ilustrado na Figura 1, o RaspberryPI B+ possui a maioria dos pinos de comunicação com sinal digital, possuindo apenas o estado baixo (0V) e alto (3,7V ou 5V). Porém, existem dois pinos exclusivos de comunicação serial, um pino é responsável pela leitura (GPIO15) e outro pela escrita (GPIO14). Toda pinagem opera na tensão de 5V de saída e com entrada de 3,7V, não permitindo que um GPIO possa ser ligado diretamente em outro devido a discrepância na tensão de saída e entrada.

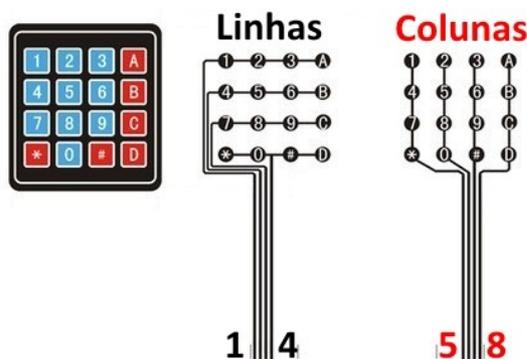
Para realizar a impressão foi utilizada uma impressora térmica, que vem sendo, frequentemente, adotada pelo comércio varejista, devido a sua agilidade na impressão e na sua relação custo/benefício satisfatória, além de não haver necessidade de realizar a troca ou recarga de cartuchos de tinta. Sua agilidade na impressão é possível em virtude de não possuir

reservatórios de tinta, já que o processo de impressão é realizado através do aquecimento do papel térmico, onde cada ponto aquecido fica na cor preta. Além disso, este tipo de impressora possui um diferencial em relação às impressoras comuns, pois, no final da impressão, possui uma funcionalidade de “guilhotina”, que é capaz de cortar o papel na medida certa.

Devido à necessidade de interação com o usuário, foi utilizado monitor touchscreen de 2,8 polegadas capaz de conectar-se diretamente na porta USB do microcomputador, sem necessidade de uma fonte externa.

Para realizar a entrada de informações no microcontrolador, foi utilizado um teclado matricial, que é organizado internamente no modo de matrix (conforme ilustrado na Figura 2), pois sua estrutura é composta por linhas e colunas. Esta estrutura foi desenvolvida com a finalidade de economizar portas de comunicação com qualquer tipo de leitor. Para realizar a leitura deste tipo de teclado é necessário utilizar uma técnica de multiplicação, onde o periférico “leitor” alimenta uma linha e verifica se alguma das colunas está com sinal positivo (indicando que a tecla está pressionada), posteriormente realiza o teste em outra linha e este ciclo se repete.

Figura 2: Especificação das linhas e colunas do teclado matricial



Fonte: Elaborado pela equipe FILIPEFLOP

### 1.1.2 Software

A linguagem de programação escolhida para o desenvolvimento do trabalho foi Java, que é uma das primeiras linguagens que suportam orientação a objetos. Sua origem ocorreu em 1991 através da empresa Sun Microsystems com base na sintaxe e semântica da linguagem C. A linguagem possui, como grande diferencial, a possibilidade de ser executada em qualquer sistema operacional que possua uma Java Virtual Machine (JVM) instalada e configurada, removendo esta responsabilidade de compatibilização do desenvolvedor. Diferentemente da linguagem C, o código fonte compilado não é transformado diretamente para código de máquina, e sim transformado para um *bytecode* que é executada pela JVM.

Dessa forma, todo o controle de compatibilidade entre os sistemas operacionais existentes fica sob responsabilidade da máquina virtual.

A tecnologia escolhida para a conectividade entre os sistemas foi os Serviços Web, que é praticamente a única saída quando se trata de bons padrões de desenvolvimento e integração. Sua simplicidade permite que softwares de grande porte troquem informações complexas com padrões simples de comunicação conforme descrito por RICHARDON e RUBY (2007). Esta comunicação é realizada na maioria dos casos por meio do protocolo HTTP, simplificando e aumentando a compatibilidade entre os sistemas.

O padrão mais utilizado para a troca de informações utilizando serviços Web é o XML (eXtensible Markup Language) que é utilizado mundialmente para a troca de informações por meio de dados organizados de forma hierárquica. A linguagem que conforme a tradução significa extensível, permite que qualquer projetista defina seus elementos. A especificação é realizada através da utilização de “tags”, que correspondem à definição da estrutura da linguagem criada. Estas tags possuem um início e um fim (exceto quando o conteúdo é vazio, neste caso, apenas uma tag pode sinalizar a abertura e o fechamento), sendo o conteúdo informado dentro deste intervalo considerado o valor da respectiva tag, por exemplo, uma tag “nome”, pode possuir o valor “João”.

#### 1.1.2.1 Cliente

Para a configuração e instalação do micro controlador Raspberry foi utilizado o RaspbianOS, que conforme descrito por NASCIMENTO (2015) foi o primeiro sistema operacional desenvolvido exclusivamente para RaspberryPI. Ele foi desenvolvido pela própria empresa Raspberry, e possui diversas funcionalidades e ferramentas educacionais voltadas para o aprendizado de linguagens de programação. Além disso, ele possui um ambiente visual, permitindo que qualquer usuário acesse seus aplicativos e diretórios de pastas sem necessidade de realizar comandos diretos no terminal. Este sistema operacional foi desenvolvido com base no Debian OS, sendo assim, possui compatibilidade parcial ou total com todos os softwares disponibilizados em repositórios das diversas distribuições Linux baseadas no Debian.

A distribuição Debian (NOVAIS, 2013) é um dos sistemas operacionais *Linux* mais utilizados e mais importantes do mundo. Sua primeira versão foi lançada em 1993, e desde seu lançamento passou a ser considerado o “pai” das distribuições, devido a sua facilidade no controle de pacotes e programas, leveza e segurança. É possível destacar que sua distribuição “filha” mais conhecida é o sistema operacional *Ubuntu*, que possui um ambiente gráfico

similar ao sistema operacional desenvolvido pela Microsoft, mundialmente utilizado – Microsoft Windows.

Devido os recursos limitados do microcontrolador, foi adotado o SQLite como banco de dados da aplicação que será instalada no hardware. Este banco de dados foi desenvolvido na linguagem C e possui a maioria das funções da linguagem SQL, porém, devido sua leveza há algumas limitações, como por exemplo a ausência de *foreign key*, *triggers*, *right join*, *full outer join*, *grant* e *revoke*. Devido à “Aplicação Cliente” não possuir diversas funcionalidades, nenhuma dessas ausências e restrições foi impactante para o desenvolvimento do projeto.

#### 1.1.2.2 Servidor

O servidor de aplicações Java utilizado foi o Glassfish, que segundo GONCALVES (2010), é um servidor de aplicação desenvolvido pela Oracle capaz de hospedar aplicações estruturadas em Java EE. Sua primeira versão foi desenvolvida em 2005, pela empresa Sun Microsystems (atualmente adquirida pela Oracle), sendo capaz de suportar apenas aplicações com estruturas Java EE 5. Atualmente sua última versão é a 4.0, que possui suporte a diversas novas tecnologias, como EJB, JMS, ferramentas de gestão de *performace* e etc. Além de hospedar a aplicação, o servidor possui como escopo garantir alta disponibilidade, tolerância a falhas, além de permitir técnicas de clusterização.

Como base de dados foi adotado o PostgreSQL, que surgiu no ano de 1986, na Universidade de Berkeley, na California, com o propósito de ser um banco de dados livre, sem necessidade de obtenção de licença para operacionalização. Apesar de ser um banco de dados gratuito, sua rapidez e robustez permitem que aplicações com grande volume de dados possam utilizá-lo sem enfrentar qualquer tipo de problema de desempenho e eficiência conforme referido por MILANI (2008). Além disso, a licença utilizada é BSD, permitindo que qualquer usuário consiga modificar o código fonte do banco de dados, de acordo com sua necessidade.

Para o desenvolvimento da interface com o usuário foi utilizando o Spring MVC, que segundo PACHECO (2007), é um *framework* utilizado para aplicações Web que utiliza o padrão MVC de desenvolvimento, com três camadas - “Model”, “View” e “Controller” - que correspondem, respectivamente, à camada de modelo de dados, camada de visualização e camada de controle para as regras de negócios e visualização. Sua principal vantagem é facilitar no desenvolvimento por conta do desenvolvedor não precisar se preocupar com o

tratamento de todos os protocolos HTTP's necessários para a comunicação entre cliente e servidor.

## **1.2 Sistemas Correlatos**

Segundo a empresa SANTANDER GETNET, a simplificação do processo de faturamento e das obrigações fiscais de uma empresa deve ocorrer de forma simples e objetiva, preferencialmente, realizadas em uma única transação. Diante desta definição, a empresa criou soluções baseadas em TEF (Transferência Eletrônica de Fundos), que permite que seus clientes emitam suas notas fiscais eletrônicas estaduais diretamente na máquina de cartão de crédito, não precisando que o contribuinte acesse um computador para realizar a emissão.

Esta solução consiste em uma máquina de cartão de crédito chamada ICMP, que necessita do emparelhamento com outro dispositivo com sistema operacional Android ou iOS, onde o hardware funciona apenas como um “escravo” do aplicativo que roda no dispositivo *mobile*.

Devido a não padronização de comunicação dos sistemas de NFS-e municipais, a empresa GETNET não suporta emissão de nota fiscal de serviço, limitando-se apenas as notas fiscais estaduais.

Outro sistema correlato é o aplicativo “NFS-e Mobile” que atualmente está sendo disponibilizado através de um aplicativo *mobile* para a realização das emissões de NFS-e. Estes aplicativos normalmente são disponibilizados na plataforma *Android* com a intenção de abranger uma camada maior de contribuintes, devido o baixo custo de aquisição de um celular com este sistema operacional (KELLERMANN, 2014).

Apesar do aplicativo auxiliar os contribuintes na transmissão das notas fiscais, devido sua portabilidade e a facilidade na utilização, ele possui uma importante limitação de apenas transmitir as notas fiscais por e-mail para o tomador do serviço, não permitindo que o usuário imprima o documento fiscal em uma impressora comum ou térmica.

## **2 EMISSOR DE CUPOM FISCAL DE SERVIÇOS POR PONTO DE ATENDIMENTO**

Um protótipo capaz de processar a emissão de cupons fiscais de serviços eletrônicos (CFS-e) foi desenvolvido, sem a necessidade de equipamentos auxiliares (computador, mouse, teclado convencional, etc). Dois softwares foram criados, sendo um exclusivo para executar e administrar o protótipo criado, e outro software Web responsável por comunicar o

sistema do próprio protótipo e de outros softwares de terceiros com os sistemas de emissão de notas fiscais de serviço dos Municípios.

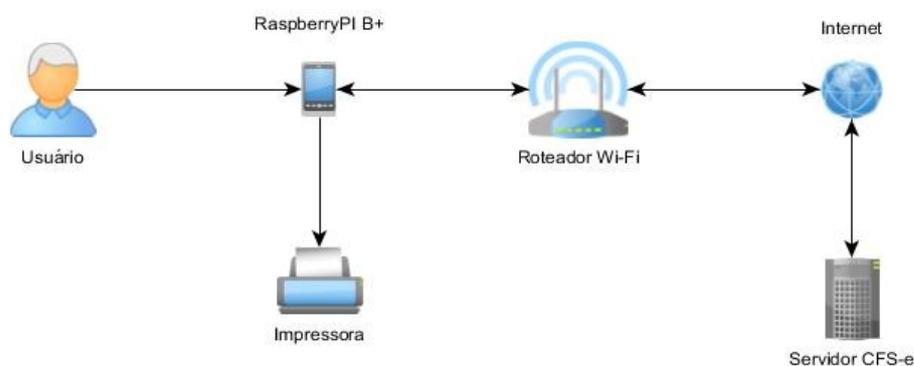
## 2.1 Visão Geral

Nesta seção será apresentada uma visão geral sobre o protótipo projetado e desenvolvido e também sobre as soluções de softwares criadas para operar na máquina cliente e no servidor de aplicação.

### 2.1.1 Protótipo

A Figura 3 mostra o fluxo de informações entre os componentes do protótipo do sistema, que possui um pequeno visor para permitir a comodidade e a portabilidade, similar a uma máquina de cartão de crédito, e, além disso, este visor está acoplado a um microcontrolador (RaspberryPI B+), que também é conectado a um teclado matricial, uma impressora (térmica ou comum) e um meio de conexão com a internet (Wi-Fi ou Ethernet).

Figura 3 – Diagrama de funcionamento do protótipo

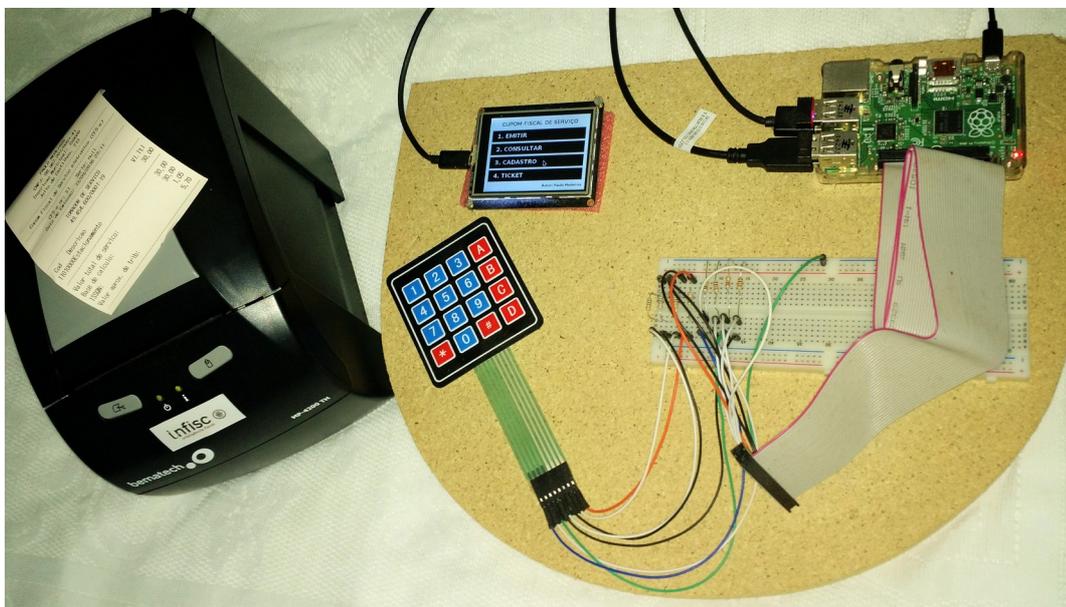


Fonte: Autor

Um software é responsável por recepcionar todas as interações do usuário com o servidor Web, diante disso, é capaz de interpretar todas as requisições informadas por meio do teclado matricial e processar suas informações para que o usuário visualize as informações no visor. O sistema permite que o prestador de serviço (usuário) consiga realizar emissão, consulta e cancelamento de seus cupons fiscais de serviços, e todas essas operações são encaminhadas para o servidor Web que é responsável por todas as interações com o sistema de cupom fiscal de serviços do Município do prestador de serviços.

O software também conta com a integração com qualquer impressora que o usuário deseje acoplar, preferencialmente uma impressora térmica, que é de melhor custo benefício e agilidade na impressão.

Figura 4 – Protótipo emissor de cupom fiscal de serviço



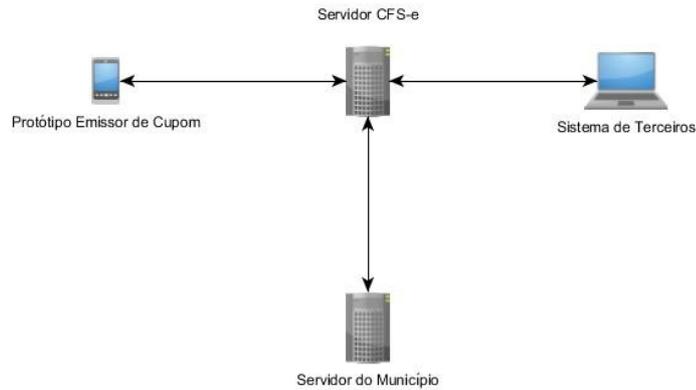
Fonte: Autor

### 2.1.2 Servidor Web

A aplicação Web é responsável pela recepção de requisições via serviços Web a partir do protótipo do equipamento emissor de CFS-e. Estas requisições serão processadas, armazenadas e comunicadas com o servidor CFS-e do Município, responsável pela recepção oficial dos cupons fiscais de serviços do Município. Assim, pode-se dizer que o software Web funciona como um *middleware*, pois é capaz de se comunicar com diversos Municípios, onde cada um pode possuir um padrão diferente de comunicação (estrutura de tags de XML, comunicação SOAP e Restfull, etc).

Além de realizar o tratamento das requisições do protótipo do equipamento emissor de CFS-e, também existe um padrão de comunicação disponibilizado para software de terceiros, que pode utilizar o “Servidor CFS-e” para realizar a integração com qualquer Município de seus clientes (conforme ilustra a Figura 5), não necessitando o desenvolvimento específico para cada novo Município, pois o Servidor CFS-e é responsável por esta comunicação.

Figura 5 – Diagrama de conectividade do projeto



Fonte: Autor

Por meio de interfaces visuais, o usuário também pode realizar o *login* em uma página Web e atualizar seu cadastro, consultar de seus cupons fiscais e demais ferramentas e funcionalidades que permitam que o prestador de serviços acompanhe o andamento de seu faturamento.

## 2.2 Funcionalidades

Para que o Portal de Integração de notas fiscais funcione, o administrador do sistema deve realizar o cadastro dos Municípios suportados e já configurados. A configuração inicia com poucas informações, como nome, UF e código do IBGE e situação da integração (ativa ou desativa), conforme exibido na Figura 6. Sempre que um novo Município for incorporado ao portal, não é necessário realizar grandes ajustes no sistema, pois as configurações foram feitas para funcionar de modo dinâmico, onde o objetivo é moldar todos os layouts de conectividade sem ajustes no código fonte.

Figura 6 – Cadastro de cidades

### Cadastro de Cidades

Nome	UF	Código IBGE	Código UF				
Sapucaia do Sul	RS	4320008	43	Configurar Web Service	Configurar XML	Desativar Integração	Excluir
Campo Bom	RS	111111	43	Configurar Web Service	Configurar XML	Ativar Integração	Excluir
Taquara	RS	10101010	43	Configurar Web Service	Configurar XML	Ativar Integração	Excluir

Fonte: Autor

A configuração de um novo Município é separada em dois momentos, o primeiro é a configuração dos endereços de conectividade, onde é necessário informar as URL's de acesso para envio dos cupons fiscais e também de cancelamento. Além disso, também é importante que o usuário configure qual tag do XML que corresponde ao resultado do processamento e qual valor deve corresponder ao resultado positivo da integração. Por exemplo, como demonstrado na Figura 7, a tag que corresponde a situação da integração é “sit” e em caso do processo de integração ter ocorrido com sucesso, seu conteúdo é 100. Porém, caso o processo ocorra com rejeição, o detalhe do processamento é informado dentro da tag “motivos”.

Figura 7 – Configuração de Web Service

### Web Service - Cidade: Sapucaia do Sul

**URL Envio de Cupom**

**Namespace Envio de Cupom**

**Método Envio de Cupom**

**Tag com resultado da critica**

**Valor positivo para processamento do envio do cupom fiscal**

**Tag com motivos de erro no processamento**

**URL de consulta do CF**

```

<?xml version="1.0" encoding="UTF-8" ?>
<resultadoLoteCupom versao="1.0">
  <CNPJ>08967207000141</CNPJ>
  <cLote>463</cLote>
  <CFSe>
    <chvAcessoCFSe>4308967207000141980CF000000002000000002</chvAcessoCFSe>
    <sit>100</sit>
    <motivos>
      <mot>Cupom fiscal 2 aceito. Protocolo nr. 761</mot>
    </motivos>
  </CFSe>
</resultadoLoteCupom>

<?xml version="1.0" encoding="utf-8" ?>
<resultadoLoteCupom versao="1.0">
  <CNPJ>08967207000141</CNPJ>
  <cLote>463</cLote>
  <CFSe>
    <chvAcessoCFSe>4308967207000141980S000000001000000001</chvAcessoCFSe>
    <sit>200</sit>
    <motivos>
      <mot>Error1113 Chave de acesso inválida</mot>
    </motivos>
  </CFSe>
</resultadoLoteCupom>

```

Fonte: Autor

O segundo momento para a configuração de um novo Município é realizar o cadastro do layout do XML de conectividade. Para a realização deste procedimento é necessário que o administrador tenha em mãos o manual de conectividade do Município onde é possível obter todos os detalhes da estrutura do XML. Na Figura 8, é possível verificar uma configuração realizada à esquerda e de seu resultado à direita.

Figura 8: Configuração de XML

Fonte: Autor

É necessário que o usuário adicione cada uma das tags constantes na estrutura do layout do XML, definida no manual de conectividade, informando sua hierarquia, nome, tipo, ordem e também de onde seu conteúdo deve ser extraído.

Figura 9: Configuração de TAG

Nome <input type="text" value="nCFS-e"/>	<input type="checkbox"/> Repetir por item de serviço
Prioridade <input type="text" value="2"/>	Tipo <input type="text" value="Número (Integer/Decimal)"/>
<input type="checkbox"/> Tag agrupadora	Pattern Valor <input type="text" value="0000000"/>
<input type="checkbox"/> Fixo	Classe <input type="text" value="Cupom Fiscal"/>
<input type="checkbox"/> Parametro	Atributo <input type="text" value="Número do Cupom"/>
Valor Fixo / Parâmetro <input type="text"/>	Tag Pai <input type="text" value="Id"/>
Expressão para geração de valor <input type="text"/>	

Fonte: Autor

Após as devidas configurações de conectividade do Município, o administrador pode adicionar clientes (usuário), que podem utilizar o Portal de Integração, juntamente com a

máquina física de emissão de cupom fiscal. No primeiro acesso, o cliente deve preencher todos os seus dados, conforme o cadastro municipal, para que o sistema da Prefeitura não rejeite suas notas fiscais. Estas informações consistem de dados cadastrais primordiais, como endereço, CNPJ, inscrição municipal, e-mail e serviços autorizados. Também é necessário que o usuário informe o certificado digital no tipo A1 para realizar a integração, pois a assinatura digital é fundamental para realizar a integração com qualquer tipo de Município. E por fim, caso o usuário possua o ramo de estacionamento, ele também pode configurar os valores que podem ser cobrados por períodos de horas, permitindo que o sistema apure automaticamente o valor do serviço através da utilização de tickets (que será descrito abaixo).

Figura 10: Telas de cadastro

### Dados Cadastrais

**Razão Social**  
Paulo Medeiros

**CNPJ/CPF**  
08967207000141

**Inscrição Municipal**  
150000

**Inscrição Estadual**  
ISENTO

**E-mail**  
paulo@infisc.com.br

**Logradouro**  
Julio de Castilhos

**Número Logradouro**  
713

### Serviços Autorizados

Adicionar

Código	Descrição	Aliquota	Percentual Aproximado de Tributos	Lei 116	Estacionamento		
11010000	Estacionamento	3.50	19.00	1101	Sim	Editar	Excluir

### Certificado Digital

Adicionar novo certificado

**Certificado:** C:\opt\cfs\certificados\1467512231856.pfx

**Data de Cadastro:** 02/07/2016

**Alias (nome legível):** infisc.inteligencia.em.tecnologia.ltda.epp08967207000141

**Validade:** 16/06/2017

### Configuração do Ticket de Estacionamento (R\$)

Adicionar

Hora Inicial	Hora Final	Valor	
0:00h	1:00h	2,00	Excluir
1:00h	3:00h	5,00	Excluir
3:00h	8:00h	10,00	Excluir
8:00h	-	20,00	Excluir

Fonte: Autor

Por fim, o usuário pode realizar a consulta de todos os seus cupons transmitidos através do menu “Cupom Fiscal”, permitindo visualizar através do navegador todos os seus documentos transmitidos, podendo visualizar no site do Município e também validar o motivo de qualquer eventual rejeição por parte do fisco municipal.

Figura 11 – Consulta de cupons fiscais

Numero	Série	Data de Emissão	CNPJ/CPF Tomador	Valor Serviço	ISSQN	Situação	XML de Envio	XML de Cancelamento	Site do Município
50	CF	13/09/2016	025.613.470-79	5,00	0,18	Processado	XML Envio XML Resultado Envio		Abriu
51	CF	13/09/2016	49.454.600/0001-19	30,00	1,05	Processado	XML Envio XML Resultado Envio		Abriu
52	CF	13/09/2016	000.000.000-00	2,00	0,07	Rejeitado - Error1181 Item Rejeitado: Cupom fiscal 52 rejeitada. Por favor, verifique a lista de motivos Error1174 Data emissão fora do período permitido (2016-09-13)	XML Envio XML Resultado Envio		Abriu

Fonte: Autor

O software desenvolvido, para ser executado no protótipo, inicia sua configuração exibindo uma tela informando o código de validação da máquina. Este código deve ser cadastrado no software Web, no menu “Máquina”. Após o usuário realizar o devido cadastro da máquina na página Web, o software inicia o processo de configuração, realizando o download do certificado digital e todas as demais informações cadastrais. Assim que este processo é concluído, o menu principal do sistema é exibido, conforme exibido na Figura 12.

Figura 12 – Tela inicial programa cliente

Fonte: Autor

Por meio do teclado matricial, o usuário pode navegar no menu, onde cada número corresponde a uma opção. Por exemplo, ao digitar o número “1”, o sistema inicia o processo de emissão de cupom fiscal.

Para os prestadores de serviço do ramo de estacionamento, o sistema dispõe uma funcionalidade de geração de ticket, onde o usuário informa a placa do carro e o protótipo

realiza a impressão de um documento não fiscal que registra a hora e a data de entrada do veículo.

Figura 13 – Emissão de Ticket

**PLACA**

\_\_\_\_\_

A. CONFIRMAR

B. VOLTAR

Fonte: Autor

O processo de emissão possui diversos passos, onde o prestador de serviço inicia informando o tipo do tomador, podendo ser pessoa física, jurídica ou até mesmo “Tomador não Informado”. Após esta escolha, o sistema exige o CPF ou CNPJ, de acordo com a opção escolhida. Posteriormente, o usuário pode utilizar um ticket já impresso ou não, caso opte por informar um ticket, as telas de escolha do serviço prestado, quantidade e valor não são exibidas para preenchimento, pois o sistema realiza os cálculos automaticamente, indo diretamente para a tela de resumo do cupom fiscal.

Figura 14 – Emissão de cupons fiscais

<b>CPF</b> _____-_____-_____ A. CONFIRMAR B. VOLTAR	<b>TICKET</b> _____ A. CONFIRMAR B. VOLTAR C. SEM TICKET	<b>SERVIÇOS</b> 1. Estacionamento 2. Voltar
<b>QUANTIDADE</b> ____1 A. CONFIRMAR B. VOLTAR	<b>VALOR</b> ____10,00 A. CONFIRMAR B. VOLTAR	<b>RESUMO</b> Tomador: 025.613.470-79 Qtd Serviços: 1 Valor: 10,00 ISSQN: 0,35 A. CONFIRMAR B. VOLTAR

Fonte: Autor

Para que o usuário consiga realizar o acompanhamento do processamento dos cupons fiscais, é possível realizar a consulta de seus cupons, onde os cupons vermelhos são os que foram rejeitados pelo Município e os verdes os que foram aceitos. A partir da tela do cupom

fiscal, é possível realizar novamente a impressão ou também no caso de rejeição, imprimir o motivo pelo qual o Município recusou o documento.

Figura 15 – Consulta de cupom fiscal

CUPONS FISCAIS	CUPOM FISCAL	CUPOM FISCAL
1. Número: 54 - 21/09/2016 - Vir: 100,00 2. Número: 53 - 21/09/2016 - Vir: 15,00 9. Voltar	Número: 54 Emissão: 21/09/2016 Tomador: 025.613.470-79 Valor: 100,00 ISSQN: 3,50 Situação: Processado 1. Imprimir 3. Cancelar B. Voltar	Número: 53 Emissão: 21/09/2016 Tomador: 025.613.470-79 Valor: 15,00 ISSQN: 0,53 Situação: Rejeitado 2. Imprimir Rejeição B. Voltar

Fonte: Autor

### 2.3 Características

O sistema desenvolvido permite que as empresas responsáveis por ERP's não necessitem se preocupar em ajustar seus sistemas toda vez que exista a necessidade de se integrar com um novo Município, pois a ferramenta desenvolvida permite que os sistemas de gestão utilizem um padrão único de conectividade, ficando toda a responsabilidade de integração com o Portal de Integração.

Além de auxiliar as empresas de software, o protótipo também auxilia aqueles prestadores de serviço que não possuem contato direto com informática, pois a máquina é de fácil operacionalização, através de menus simples e objetivos, que permitem ao usuário operar de forma ágil e confiante. Ademais, é importante ressaltar que não existe necessidade do usuário adquirir um computador para realizar a emissão de seus cupons, baixando o custo para aderir as obrigações acessórias municipais.

### 3. DESENVOLVIMENTO

O desenvolvimento do sistema foi realizado utilizando um modelo de processo de software incremental. A escolha desse modelo foi motivada por conta de se tratar de um projeto de pesquisa e por isso, nem todos os requisitos podem ser inteiramente previstos desde o início, podendo eventualmente surgirem novos requisitos ou reformulações ao longo da realização do projeto. Em cada incremento, foi adotado um processo sequencial (cascata), pois o projeto de pesquisa possui apenas um colaborador responsável por todas as etapas do desenvolvimento, desse modo, se tornará mais fácil o controle de prazos e andamento. Neste modelo de processo de software, cada incremento foi composto pelas fases de análise, projeto, codificação, testes e implantação.

### 3.1 Análise

A fase de análise foi iniciada realizando um levantamento de todas as funcionalidades que seriam necessárias para resolver ao máximo os problemas identificados. Com isso, foi realizado o levantamento de requisitos dos dois sistemas, sendo o primeiro para o software que está rodando no protótipo e outro para o sistema responsável por realizar toda a integração entre o protótipo, ERP e sistemas de NFS-e dos Municípios. Ambos os levantamentos estão descritos nas Tabelas 1 e 2.

Tabela 1 – Requisitos Funcionais do Sistema - Protótipo

<b>Requisito Funcional</b>	<b>Descrição do Requisito</b>
RF001	Permitir que o usuário emita cupom fiscal de serviço;
RF002	Permitir que o usuário consulte seus cupons emitidos;
RF003	Permitir que o usuário envie o cupom fiscal por e-mail (juntamente com o cupom e a NFS-e) (funcionalidade secundária);
RF004	Permitir que o usuário realize o controle de cobrança de um estacionamento rotativo;
RF005	Permitir que o usuário realize as configurações do certificado digital;
RF006	Permitir que o usuário cancele algum cupom já emitido;
RF007	Permitir que o usuário emita cupons fiscais mesmo estando sem comunicação com a integração ou servidor de integração;
RF008	Permitir que o usuário realize a impressão do cupom fiscal;

Fonte: Autor

Tabela 2 – Requisitos Funcionais do Sistema – Portal de Integração

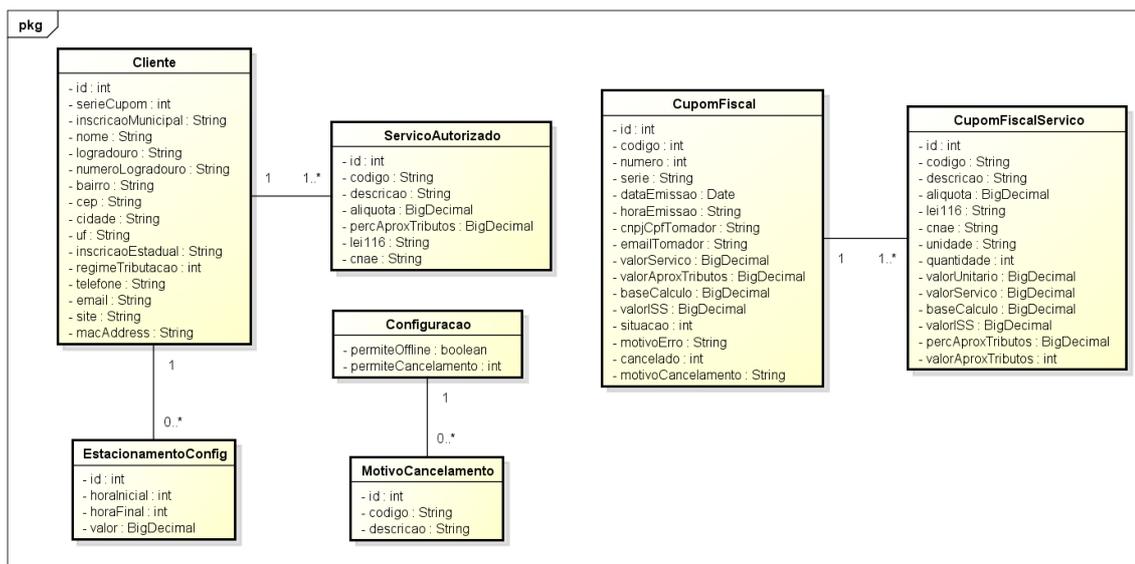
<b>Requisito Funcional</b>	<b>Descrição do Requisito</b>
RF001	Permitir que a aplicação recepcione os XML's enviados pelo programa cliente;
RF002	Permitir que o usuário realize a configuração do certificado digital para a integração com o Município destino;
RF003	Permitir que o administrador do sistema realize as configurações necessárias para cada integração do cupom com cada Município (XML, URL's e etc);
RF004	Permitir que o administrador cadastre todos os clientes que tenham adquirido a máquina de emissão de cupom, vinculando-o com o Município respectivo de integração;

RF005	Permitir que o usuário consulte todos os seus cupons fiscais com suas NFS-es respectivas;
RF006	Permitir que o usuário visualize “dashboards” e gráficos com os dados dos cupons emitidos (funcionalidade secundária);
RF007	Permitir que o sistema transmita para o Município as requisições recebidas pelo programa cliente;
RF008	Permitir que o administrador visualize os logs de todas as operações realizadas no sistema Web (funcionalidade secundária);
RF009	Permitir que o usuário envie a NFS-e gerada por um cupom para o cliente (funcionalidade secundária);
RF010	Permitir que o usuário visualize todos os XML's comunicados com o Município (funcionalidade secundária);
RF011	Permitir que o usuário bloqueie a máquina emissora de cupom fiscal (no caso de furto/roubo);
RF012	Permitir que o usuário altere seus dados cadastrais;
RF013	Permitir que o usuário cadastre descrições personalizadas para imprimir juntamente com o cupom (funcionalidade secundária).
RF014	Permitir que o usuário do ramo de estacionamento configure os parâmetros de precificação conforme desejado;

Fonte: Autor

Além do levantamento de requisitos também foram desenvolvidos dois modelos de domínio, sendo um para software do protótipo e outro para o Portal de Integração, onde ambos estão exibidos abaixo.

Figura 16 – Diagrama de domínio do Software Cliente



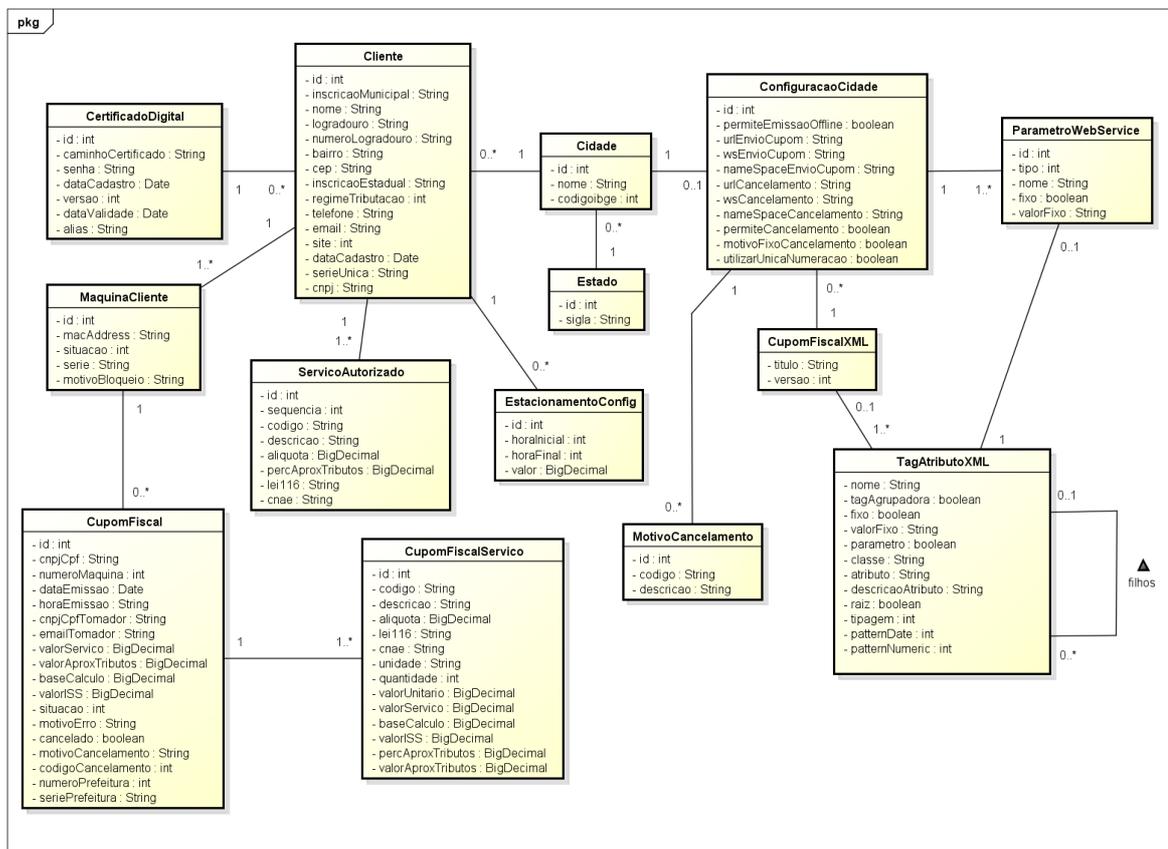
powered by Astah

Fonte: Autor

Neste diagrama de domínio mostrado na Figura 16, é possível identificar que somente as informações cadastrais do contribuinte são armazenadas no programa cliente, não

sendo necessário armazenar nenhuma informação de configuração para conversão e tratamento das informações.

Figura 17 – Diagrama de domínio do Software Web



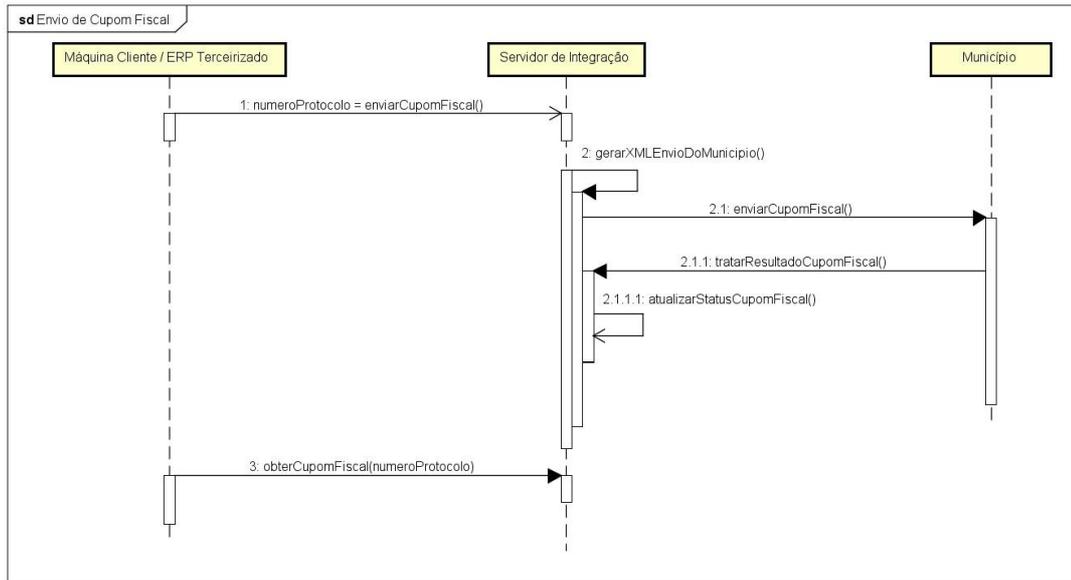
Fonte: Autor

Neste diagrama exposto na Figura 17, vale destacar a importância da classe “TagAtributoXML”, que possui vínculo com os parâmetros de XML. Para cada tag necessária para a geração do XML, um registro é inserido nesta tabela. Em alguns Municípios, existem parâmetros de serviços Web, sendo assim, é possível gerar uma estrutura de configuração para cada parâmetro, tornando o programa mais dinâmico e aumentando a quantidade de Municípios aceitos pelo sistema sem necessidade de adequação no código fonte.

### 3.2 Projeto

A partir dos requisitos levantados, foi necessário realizar dois diagramas de sequência para projetar a maneira que o software iria proceder em relação às integrações entre os sistemas. Um diagrama foi realizado para modelar a integração de envio de cupom fiscal (Figura 18) e o outro foi projetado para o cancelamento.

Figura 18 – Diagrama de seqüência do envio de cupom fiscal



powered by Astah

Fonte: Autor

Na Figura 18, é possível identificar que o processo de emissão funciona de forma assíncrona, onde o programa cliente ou ERP de terceiro envia o cupom fiscal para posteriormente buscar o retorno do processamento.

Além do diagrama realizado, antes de realizar o desenvolvimento, todos os XMLs fixos de comunicação foram definidos, os quais estão listados abaixo na Figura 19.

Figura 19 – XML's de comunicação

- 01-01-obterCadastro.xml
- 02-01-respostaCadastroErro.xml
- 02-02-respostaCadastroSucesso.xml
- 03-01-alteracaoCadastral.xml
- 04-01-respostaAlteracaoCadastral.xml
- 05-01-envioCupom.xml
- 06-01-respostaCupom.xml
- 07-01-cancelarCupom.xml
- 08-01-respostaCancelarCupom.xml

Fonte: Autor

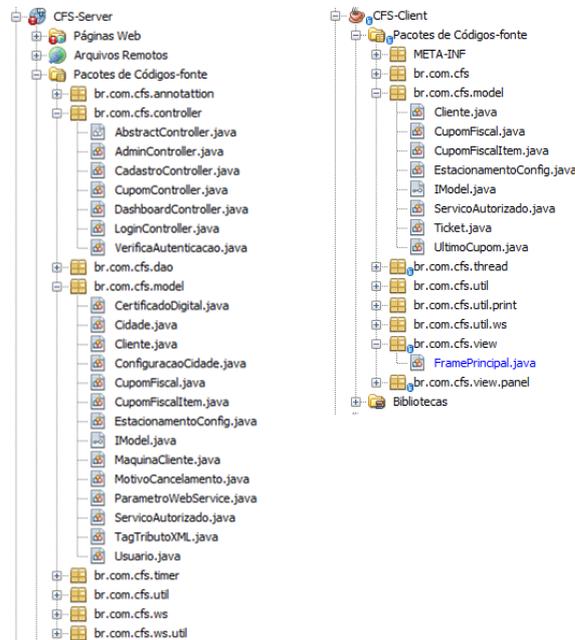
É possível identificar na Figura 19 que os XML's foram listados em seqüência de execução, por exemplo, o primeiro passo é "01 - obter cadastro", podendo retornar "02 - 01 - resposta com sucesso" ou "02 - 01 - resposta com erro".

### 3.3. Codificação

A codificação de ambos os softwares, para equipamento emissor de CFS-e e do servidor CFS-e, foi realizada utilizando o conceito de orientação a objetos, e para isso, foi

adotada a linguagem de programação Java. O ambiente de codificação foi a IDE NetBeans 8.0, que foi responsável pela entrada dos códigos fontes e pela saída com o código compilado para a JVM.

Figura 20 – Código fonte no NetBeans



Fonte: Autor

Na Figura 20, é possível identificar as classes Java e seus respectivos pacotes. Na parte da esquerda estão as classes utilizadas pelo Servidor Web e na direita estão as classes do programa cliente.

Para realizar o controle de versão e manutenção do sistema foi utilizada a ferramenta GIT, através do ambiente online BitBucket.

A integração do software do programa cliente com o teclado matricial foi realizada através dos pinos GPIO (*General Purpose Input/Output*), que são portas programáveis de entrada e saída. Através do próprio código Java, é possível realizar esta integração, onde se controla a leitura e escrita dos pinos, onde 0V é considerado LOW e 3,3V é considerado HIGH.

Figura 21 – Comunicação Java com GPIO

```
this.pinSaida0 = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_00, PinState.LOW);
this.pinSaida2 = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_02, PinState.LOW);
this.pinSaida3 = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_03, PinState.LOW);
this.pinSaida12 = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_12, PinState.LOW);

this.pinLeitura4 = gpio.provisionDigitalInputPin(RaspiPin.GPIO_04, PinPullResistance.PULL_DOWN);
this.pinLeitura5 = gpio.provisionDigitalInputPin(RaspiPin.GPIO_05, PinPullResistance.PULL_DOWN);
this.pinLeitura6 = gpio.provisionDigitalInputPin(RaspiPin.GPIO_06, PinPullResistance.PULL_DOWN);
this.pinLeitura10 = gpio.provisionDigitalInputPin(RaspiPin.GPIO_10, PinPullResistance.PULL_DOWN);
```

Fonte: Autor

É possível identificar na Figura 23 que os pinos de saída iniciam todos “desligados”, enquanto os pinos de entrada são iniciados com valor padrão “desligado”. Estas informações são importantes para que a API do GPIO funcione corretamente.

### **3.4. Testes**

A aplicação foi submetida a diversos testes funcionais, mantendo o foco na emissão de cupons fiscais a partir da máquina cliente e na conversão dos dados recebidos pelo Portal de Integração na transformação dos XML's pré-configurados. Com os testes foram identificadas as seguintes correções que impactavam na emissão do cupom fiscal:

- a) A partir de uma determinada quantidade de cupons emitidos na máquina cliente, o banco de dados HSQLBD era corrompido, perdendo alguns dos cupons emitidos. Para efetuar a correção, foi adotada a utilização do banco de dados SQLite, criando uma camada de persistência nativa, sem a utilização de JPA;
- b) Quando um cupom era emitido realizando a importação de um ticket já impresso com horário inicial superior às 24h, o cálculo automático do valor de serviço não era realizado corretamente, gerando a rejeição do mesmo pelo site do Município. O cálculo do valor unitário foi ajustado e os cupons passaram a ser aceitos pelo sistema municipal;
- c) Após o vencimento do certificado digital, os cupons passavam a ser rejeitados pelo site do Município, não gerando nenhum alerta para o usuário. Foi criado um mecanismo que realizar a verificação a data de vencimento do certificado e alerta periodicamente sobre o vencimento, não permitindo emitir cupons fiscais com o certificado vencido.

### **3.5. Implantação**

O software do programa cliente foi configurado no Linux instalado no Raspberry (Raspbian OS), juntamente com toda a configuração de pinagem para que o software identifique quais portas do GPIO correspondem ao teclado matricial. O software foi instalado no diretório /opt e um atalho foi criado na área de trabalho do Raspberry para que o usuário tenha fácil acesso.

O sistema Web foi implantado em um servidor GlassFish rodando na porta 8080, através do subendereço “CFS-Server”, por enquanto, o servidor é iniciado somente em

ambiente interno ou local, sendo assim, só pode ser acessado através de endereço interno da rede ou *localhost*.

#### 4. RESULTADOS

Por meio do trabalho desenvolvido, foi possível realizar a integração com dois Municípios do Rio Grande do Sul. O Município de Caxias do Sul não possui a funcionalidade de emissão de Cupom Fiscal de Serviço, sendo assim, foi realizada a configuração de transmissão no modo de NFS-e, onde a configuração do XML da nota fiscal é mais complexa que a configuração de um simples cupom fiscal.

O outro Município homologado foi o de Sapucaia do Sul, que possui já a funcionalidade de CFS-e. Neste caso, se encaixou perfeitamente no projeto proposto.

Figura 22 – NFS-e de Caxias do Sul e CFS-e de Sapucaia do Sul

**PDF DOCUMENT: NFS-e de Caxias do Sul**

**PDF DOCUMENT: CFS-e de Sapucaia do Sul**

Fonte: Autor

Na Figura 22, é possível identificar uma NFS-e transmitida pelo sistema de integração no ambiente de Caxias do Sul à esquerda e um Cupom Fiscal de Serviço emitido no Município de Sapucaia do Sul à direita.

## 5. TRABALHO FUTURO

Diante do tempo de execução do projeto e do escopo pré-estabelecido, alguns requisitos funcionais e melhorias não foram considerados para a elaboração da versão inicial do sistema. Dentre estes itens, é possível citar:

- a) A disponibilização de um *dashboard* para que o contribuinte visualize em forma de gráficos a evolução do seu faturamento e a partir destas informações possa adotar estratégias para melhorar o desempenho da prestação de serviço;
- b) Mecanismo de autocadastro do usuário, sem a necessidade que o administrador do sistema realize seu cadastro;
- c) Gerenciamento de pagamento pela utilização do sistema, não necessitando que o administrador realize o bloqueio manual do usuário, caso o mesmo não efetue o pagamento da mensalidade;
- d) Funcionalidade para recuperação de senha para os usuários que esquecerem sua senha;
- e) Implantar o sistema em um *datacenter* e configurar um endereço BR para ser acessado via DNS.

## 6. CONCLUSÃO

Este trabalho de pesquisa e desenvolvimento foca seu objetivo na simplificação do processo de cumprimento das obrigações acessórias relacionadas com a transmissão de notas e cupons fiscais de forma eletrônica.

A partir do projeto desenvolvido, foi possível comprovar que o sistema atende aos seus objetivos, pois é possível constatar que a integração com os Municípios de Sapucaia do Sul e Caxias do Sul foram possíveis através da configuração simples de XML pelo usuário, não necessitando de alterações no código fonte do sistema. Diante disto, um processo que normalmente é burocrático e que pode durar semanas, foi solucionado em poucos minutos.

Portanto, é importante ressaltar que, apesar do hardware estar em fase de protótipo, sua operacionalização foi realizada conforme esperado, pois os testes executados com ambos Municípios foram feitos a partir da própria máquina emissora de cupom, onde foi possível realizar todo o processo de integração e impressão dos documentos fiscais.

## REFERÊNCIAS

- LDTR. **Ligando Display Touchscreen no RaspberryPI**. Disponível em: <<http://blog.filipeflop.com/embarcados/display-raspberry-pi-touchscreen.html>>. Acesso em: 1 out. de 2015.
- CUTMA. **Como usar o teclado matricial 4x4 com Arduino**. Disponível em: <<http://blog.filipeflop.com/arduino/teclado-matricial-4x4-arduino.html>>. Acesso em: 18 set. de 2015.
- CUAWR. **Como utilizar um adaptador WiFi com o RaspberryPI**. Disponível em: <<http://blog.filipeflop.com/embarcados/como-utilizar-adaptador-wifi-raspberry-pi.html>>. Acesso em: 20 set. de 2015.
- PPRP. **Primeiros Passos com o Raspberry PI**. Disponível em: <<http://blog.filipeflop.com/embarcados/tutorial-raspberry-pi-linux.html>>. Acesso em: 20 set. de 2015.
- GPIO. **GPIO: Raspberry PI Models A and B**. Disponível em: <<https://www.raspberrypi.org/documentation/usage/gpio/>>. Acesso em: 20 set. de 2015.
- DDTC. **Divisores de tensão e de Corrente**. Disponível em: <[http://www.ufrgs.br/eng04030/Aulas/teoria/cap\\_04/diviteco.htm](http://www.ufrgs.br/eng04030/Aulas/teoria/cap_04/diviteco.htm)>. Acesso em: 3 nov. de 2015.
- RSPFO. **RaspberryPI Foundation**. Disponível em: <<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>>. Acesso em: 2 nov. de 2015.
- HOJT. **The History of Java Technology**. Disponível em: <<http://www.oracle.com/technetwork/java/javase/overview/>>. Acesso em: 3 nov. de 2015.
- CFIT. **Como funciona a impressora térmica**. Disponível em: <<http://www.gomaq.com.br/blog/impressao/como-funciona-a-impressora-termica>>. Acesso em: 3 nov. de 2015.
- SOUZA, Vitor E. Silva. **Novidades do GlassFish 3.1 – Artigo Java Magazine 91**. Espírito Santo, 2014.
- APIC. **Artigo SQL Magazine 6 - PostgreSQL Introdução e Conceitos**. Disponível em <<http://www.devmedia.com.br/artigo-sql-magazine-6-postgresql-introducao-e-conceitos/7185>>. Acesso em 16 out. de 2015.
- MARTINS, Marcelo. **Criando um Webservice RESTful em Java**. São Paulo, 2010.
- SMVC. **Spring MVC**. Disponível em <<https://www.caelum.com.br/apostila-java-web/spring-mvc/#11-1-porque-precisamos-de-frameworks-mvc>>. Acesso em 3 out. 2015.
- EXML. **Extensible Markup Language (XML)**. Disponível em <<http://www.w3.org/XML/>>. Acesso em 3 out. 2015.
- RICHARDSON, Matt; WALLACE Shawn. **Primeiros Passos com Raspberry PI**. 1ª ed. São Paulo, São Paulo, 2013.
- SHARWOOD, Simon. **New Raspberry PI B+**. Disponível em [http://www.theregister.co.uk/2014/07/14/raspberry\\_pi\\_b\\_debuts\\_with\\_four\\_usb\\_ports/](http://www.theregister.co.uk/2014/07/14/raspberry_pi_b_debuts_with_four_usb_ports/). Acesso em 9 nov. 2015.
- GONÇALVES, Antonio. **Beginning Java EE 6 Platform with GlassFish 3**. 2º Ed. New York, NY, 2010.
- PATROCÍNIO, José Antônio. **ISSQN - Teoria, Jurisprudência, e Prática**. 1º Ed. Guarilhos, SP, 2011.
- MILANI, André. **PostgreSQL Guia do Programador**. 1º Ed. Paraná, 2008.
- NASCIMENTO, Felipe. **Instalando o Raspbian no Raspberry PI**. Disponível em: <<http://www.felipenascimento.com.br/instalando-o-raspbian-no-raspberry-pi/>>. Acesso em 9 nov. 2015.

RICHARDON, Leonard; RUBY, Sam. **RESTFul, Serviços Web**. 1º Ed. Rio de Janeiro, 2007.

PACHECO, Diego. **Spring Framework (2.0) - Framework para Desenvolvimento de Aplicações em Java**. 1º Ed. 2007.

NOVAIS, Claudio. **Debian, o sistema operacional mais usado em servidores**. Disponível em: <<http://www.forum.ubuntued.info/viewtopic.php?f=37&t=4210>>. Acesso em 9 nov. 2015.

KELLERMANN, Fabiana B. **Emissão de NFS-e pelo celular já é realidade em Campo Bom**. Disponível em <<http://novo.campobom.rs.gov.br/noticia-4152/emissao-de-nfs-e-pelo-celular-ja-e-realidade-em-campo-bom>>. Acesso em 11 nov 2015.

SEBRAE RJ. **Benefícios da NFS-e**. Disponível em [http://www.sebraerj.com.br/docs/NFe\\_SERVICOS.pdf](http://www.sebraerj.com.br/docs/NFe_SERVICOS.pdf). Acesso em 15 nov. 2015.

SANTANDER GETNET. **Solução de TEF**. Disponível em <<https://www.santandergetnet.com.br/site/Tecnologia/solucoes-tef>>. Acesso em 15 nov. 2015.