

FRAMEWORK PARA DESENVOLVIMENTO DE PROGRAMA EMISSOR DE CUPOM FISCAL

¹Newton Edgar Ribas de Mattos¹, Reiner Franchesco Perozzo¹

¹ Centro Universitário Franciscano

Caixa Postal – 91.501-970 – Santa Maria – RS – Brasil

newton.edgar@gmail.com, reiner.perozzo@unifra.br

Abstract. *The objective of this work is to develop a framework in Java Language, to enable the development of tax coupon issuer program for models of printers manufactured by Logic Industry and Trade of Electronic Equipment. In the world of business automation, the program that sends commands to the fiscal printer, or print the coupon, is called PAF - Fiscal Application Program and the printer name is ECF - Issuing Fiscal Coupons. The development of this type of application aims to meet the tax law relating to undertakings that fall within the requirement to issue coupon tax, due to the steady increase in companies engaged in the sale of goods or services and has already reached the limit of compulsory . The programs developed using the framework obtained a reduction in development time, the final cost of the product, taking into account current legislation and can be used by any company that is required to issue the tax coupon.*

Resumo. *O objetivo deste trabalho é a elaboração de um Framework, na Linguagem Java, para possibilitar o desenvolvimento de programa emissor de cupom fiscal, para os modelos de impressoras fabricadas pela Bematech Indústria e Comércio de Equipamentos Eletrônicos. No mundo da automação comercial, o programa que envia os comandos para a impressora fiscal, ou seja, imprime o cupom, é chamado de PAF – Programa Aplicativo Fiscal e o nome da impressora é ECF – Emissor de Cupom Fiscal. O desenvolvimento deste tipo de aplicativo visa atender a legislação tributária relativo às empresas que se enquadram dentro da obrigatoriedade de emitir cupom fiscal, em função do aumento constante de empresas que atuam na venda de mercadorias ou prestação de serviços e que já alcançaram o limite da obrigatoriedade. Os programas desenvolvidos com a utilização do framework obtiveram uma redução no tempo de desenvolvimento, no custo final do produto, atendendo a Legislação vigente e podem ser utilizados por qualquer empresa que esteja obrigada a emitir o cupom fiscal.*

1. Introdução

O objetivo específico deste trabalho é o desenvolvimento de um *framework* a ser utilizado para desenvolver programas de emissão de cupom fiscal. O Emissor de Cupom Fiscal – ECF é o software que se comunica com Impressora Fiscal, possibilitando assim a sua utilização. O desenvolvimento deste tipo de programa requer habilidade em programação, seja a linguagem que for, representando investimento e tempo para sua conclusão.

Então o *framework* PAF-ECF, possibilita o desenvolvimento deste tipo de programa com meia dúzia de linhas de código, em Java, sem a preocupação de uma análise mais profunda na Legislação Tributária que norteia a estrutura de um programa emissor de cupom fiscal. Com o uso do *framework* PAF – ECF, o programa desenvolvido atende a Legislação vigente, imprimindo o cupom fiscal de acordo com a Fiscalização, para qualquer tipo de empresa.

As empresas brasileiras buscam inovações na área da tecnologia, adquirindo responsabilidades sociais, como participantes no desenvolvimento do Estado, cumprindo seus compromissos legais e tributários. O atendimento à legislação que trata da obrigatoriedade de uso do ECF, elemento integrante da automação comercial, demonstra que a empresa precisa conhecer e gerenciar os impostos que tem a recolher demonstrando para o seu cliente, real contribuinte do tributo, um atendimento rápido, seguro e transparente. Com a aquisição e implantação do ECF a empresa atende a legislação fiscal e ainda permite fazer a gestão dos seus negócios [Gonçalves, 2014].

Para que esta impressora funcione é necessário um programa específico, com determinadas características, evitando assim o desvio de registros que poderiam influenciar nos impostos a pagar. Este programa tem o nome de PAF – Programa Aplicativo Fiscal e é o único programa a se comunicar com a Impressora Fiscal [Mouta 2014].

Em virtude da dificuldade de desenvolvimento deste programa aplicativo fiscal, pela complexidade dos requisitos legais estabelecidos na legislação das Secretarias Fazendárias dos Estados brasileiros, e com diferenças de Estado para Estado, foi desenvolvido um *framework*, em Java, para facilitar a vida do programador, agilizando o processo de desenvolvimento de programas emissor de cupom fiscal.

Este trabalho consiste então num conjunto de classes escritas na Linguagem Java, disponibilizadas num *framework*, com a finalidade de facilitar o desenvolvimento de aplicativo emissor de cupom fiscal, dando possibilidade ao desenvolvedor, mais rapidez na entrega dos softwares, atendimento mais rápido do cliente.

Difícilmente uma empresa terá poder intelectual, financeiro, mão de obra disponível e tempo para competir com a comunidade mundial. Em outras palavras, o desenvolvedor de software de uma corporação deve conhecer o máximo possível das opções de componentes e *frameworks* existentes no mercado para não cair no velho e já conhecido buraco de tentar “reinventar a roda” à sua maneira [Oliveira, 2009].

2. Referencial Teórico

Nesta seção será apresentado uma descrição sobre impressora fiscal que emite os cupons, chamada de ECF, sobre o PAF que é o Programa Aplicativo Fiscal, responsável pela comunicação com a impressora, para impressão de cupom fiscal e sobre *frameworks* de desenvolvimento de *software*.

2.1. ECF – Emissor de Cupom Fiscal

Segundo definição do CONVÊNIO ICMS 9, de 3 de abril de 2009, ECF é o equipamento de automação comercial e fiscal com capacidade para emitir, armazenar e disponibilizar documentos fiscais e não fiscais e realizar controles de natureza fiscal referentes a operações de circulação de mercadorias ou a prestações de serviços, implementado na forma de impressora com finalidade específica (ECF-IF) e dotado de Modulo Fiscal Blindado (MFB) que recebe comandos do Programa Aplicativo Fiscal - Emissor de Cupom Fiscal (PAF-ECF) externo [Gonçalves, 2014].

As empresas que exercem as atividades de venda ou revenda de bens a varejo ou prestadoras de serviços e cujo faturamento anual ultrapasse o limite estabelecido por Normas em cada Estado, estão obrigadas ao uso de equipamento Emissor de Cupom Fiscal de acordo com a Lei nº 9.532, de 10 de dezembro de 1997, art. 61 [Mouta, 2014].

O ECF é normalizado através de Convênios celebrados entre a União, os Estados e Distrito Federal, pelo Conselho Nacional de Política Fazendária (CONFAZ), integrando-se à legislação estadual por meio de decretos e outras normas complementares. Porém, os assuntos e aspectos que não são tratados ou detalhados pelos Convênios são regulamentados diretamente no Regulamento do ICMS, por meio de Decreto Estadual ou em Portarias expedidas pela Subsecretaria da Receita Estadual – SER [Gonçalves, 2014].

Portanto, o uso de ECF é uma obrigação prevista em lei federal, regulamentada por convênio nacional e estabelecida nos Estados, por força de decreto do Poder Executivo.

Emissor de Cupom Fiscal é o equipamento de automação comercial com capacidade de emitir documentos fiscais e realizar controles dos valores referentes a operações de circulação de mercadorias ou a prestações de serviços. [Mouta, 2014].

Sendo o ECF um equipamento de controle fiscal, a sua fabricação, comercialização e utilização estão sujeitas às regras estabelecidas na legislação do ICMS, dentre as quais se podem destacar:

- Deve ser produzido atendendo aos requisitos técnicos de *hardware* e *software* estabelecidos em Convênios celebrados pelo Conselho de Política Fazendária – CONFAZ;
- Deve ser registrado na Comissão Técnica Permanente do ICMS – COTEPE/ICMS e na Secretaria de Fazenda do Estado;
- Somente pode ser utilizado mediante autorização expedida pela Secretaria de Fazenda do Estado;
- Para ser utilizado, deve estar lacrado com lacre especificado e controlado pelo fisco;

- Somente pode receber manutenção por meio de empresa de assistência técnica credenciada pela Secretaria de Fazenda do Estado.

2.2. PAF – Programa Aplicativo Fiscal

Os equipamentos de automação comercial, até o ano de 1994, se resumiam à Máquina Registradora (MR) e ao Terminal Ponto de Venda (PDV), os quais tinham funcionamento totalmente independente, ou seja, todas as funções e rotinas executadas pelo equipamento estavam implementadas em *software* interno ao equipamento. Dessa forma, a análise e homologação do equipamento realizado pela COTEPE/ICMS abrangiam todas as rotinas e funções por ele executadas [Gonçalves, 2014].

Neste mesmo ano com o lançamento no mercado e a regulamentação pelo Convênio ICMS 156/94 da Impressora Fiscal (ECF-IF), que para o seu funcionamento depende de comandos enviados por programa aplicativo externo, os procedimentos de verificação e registros realizados pela COTEPE/ICMS deixaram de abranger parte das rotinas e funções antes executadas pelo *software* interno do equipamento, pois tais rotinas e funções passaram a ser executadas pelo programa aplicativo externo.

Diante disto, o Fisco das Unidades Federadas passaram a sentir a necessidade de regulamentar e de estabelecer controles sobre os programas aplicativos destinados a enviar comandos de funcionamento ao ECF-IF e sobre as empresas desenvolvedoras destes programas. Tais programas foram denominados pela legislação como Programa Aplicativo Fiscal – Emissor de Cupom Fiscal (PAF-ECF).

O PAF é então o programa que faz a interface com o ECF-IF.

Segundo definição do CONVÊNIO ICMS 9, de 3 de abril de 2009, PAF-ECF é o programa aplicativo desenvolvido para possibilitar o envio de comandos ao *Software* Básico do ECF, sem capacidade de alterá-lo ou ignorá-lo, para utilização do contribuinte usuário do ECF [Gonçalves, 2014].

2.3. Frameworks

Diante da diversidade e das necessidades existentes em diferentes domínios de aplicações computacionais, o desenvolvimento de *software* pode ter a sua complexidade associada ao conhecimento prévio do desenvolvedor e às falhas de projeto relacionadas com a arquitetura do sistema [Andrade, 2006]. Para que uma arquitetura possa ser utilizada corretamente num determinado domínio de aplicação são exigidos do projetista conhecimento do problema que está abordando e experiência no desenvolvimento de sistemas computacionais.

Desse modo, uma solução interessante utilizada em desenvolvimento de *software* são os *frameworks*, que podem ser definidos como uma biblioteca de classes, concretas e abstratas, com padrões de interação entre objetos [Rogers, 1997]. Também podem ser definidos como um conjunto de classes cooperantes para a construção de projetos reutilizáveis [Gamma et al., 2000] ou, ainda, como hierarquias de classes genéricas para um domínio de aplicações [Shalloway & Trott, 2004].

Os *frameworks* implementam uma arquitetura que pode ser utilizada de modo a tornar o desenvolvimento de sistemas computacionais mais rápidos e produtivos, em relação ao desenvolvimento tradicional, facilitando a captura de decisões de projetos

que são comuns ao domínio de uma aplicação e permitindo uma redução do tempo gasto com o desenvolvimento de novas soluções. Uma das principais contribuições de um *framework* é a definição de sua arquitetura projetada para suportar todas as aplicações do domínio de maneira flexível e extensível [Gamma et al., 2000].

A sua utilização pode contemplar um conjunto de classes implementadas em uma linguagem de programação específica, projetadas para auxiliar no desenvolvimento de *software*.

Dentro desse mesmo contexto, os *frameworks* podem ser uma solução para um conjunto de problemas em comum, com uso de classes e interfaces, que disponibilizam objetos com capacidade de capturar funcionalidades comuns em várias aplicações. Assim sendo, um *framework* pode ser considerado sob certo ponto de vista, como uma solução quase completa [Oliveira, 2009].

Diferente de uma biblioteca de classes, a caracterização de um *framework* pode ser dada pelo relacionamento existente entre as classes, com padrões interação entre objetos e, principalmente, com o objetivo de se reutilizar projeto e não apenas código.

Assim sendo, um *framework* deve ser extensível, documentado e, principalmente, reusável. O uso de *frameworks* pode oferecer benefícios claros em termos de redução de custos no desenvolvimento de novos projetos dentro de um mesmo domínio de aplicação [Oliveira, 2009].

3. Proposta

Diante do número crescente de empresas que são registradas atualmente no Brasil, para venda de mercadorias e prestação de serviços, aliado às empresas que atingem os limites da obrigatoriedade ao uso do ECF (o que representa um mercado cada vez maior na aquisição de aplicativos fiscais) [Mouta, 2014]. Este trabalho propõe a construção de um *framework* responsável por auxiliar o desenvolvimento de *software* na área de impressão de cupom fiscal.

Uma das principais contribuições deste trabalho é que o *framework* abstrai a complexidade relacionada com o conhecimento prévio da Legislação Tributária e os requisitos funcionais definidos pela Legislação (que são numerosos). Além dessa abstração, os desenvolvedores de *software* que forem utilizar o *framework* poderão canalizar seus esforços diretamente na construção da aplicação, uma vez que toda a estrutura de classes, métodos, regras e funcionalidades inerentes ao processo de emissão de cupom fiscal já estão mapeadas e contempladas no *framework*.

Na Figura 1 é apresentada uma visão geral da proposta do trabalho em que, teoricamente, um programador amparado por um *framework* teria o seu trabalho de desenvolvimento de *software* facilitado.

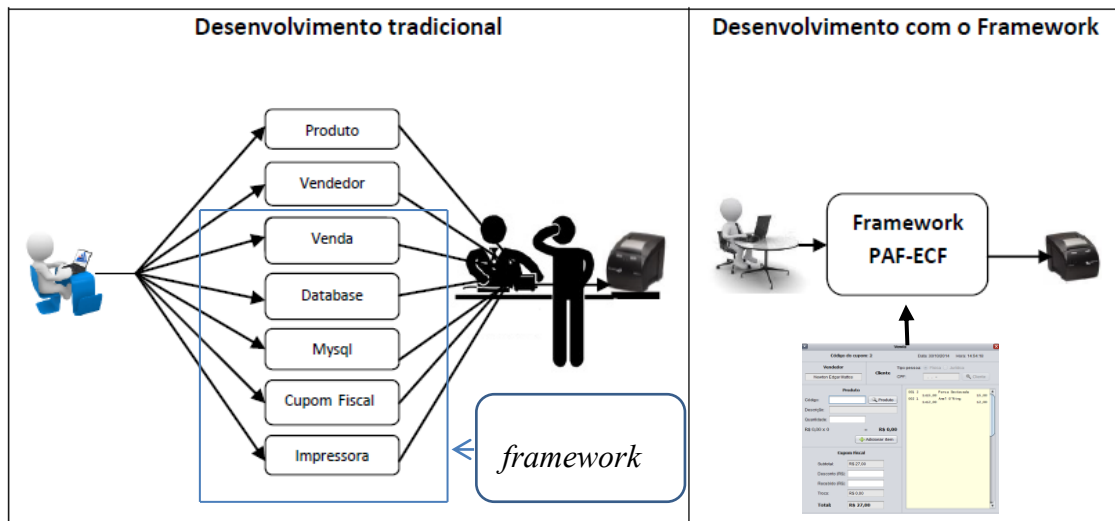


Figura 1 – Visão geral do *Framework*

3.1 Projeto

Com a apresentação da visão geral da proposta, destaca-se que o desenvolvimento deste trabalho baseia-se no uso das metodologias ágeis, as quais visam auxiliar organizações a obterem um dinamismo para acompanhar as mudanças de requisitos em um projeto [Miller 2009]. Essas metodologias se destacam das tradicionais, principalmente, por oferecerem maior prioridade à implementação das funcionalidades através código ao invés da produção da documentação escrita, oferecendo respostas rápidas às possíveis mudanças [Erdogmus, Morisio, Torchiano 2005].

Dessa forma, desenvolvimento do *Framework* PAF-ECF é norteado pela metodologia ICONIX (criado pela *ICONIX Software Engineering*) e que consiste em uma metodologia que visa à simplificação dos processos do ciclo de vida no desenvolvimento. Essa metodologia aborda um conjunto de métodos de orientação a objetos de desenvolvimento prático e imediato [Bona 2002].

O ICONIX trabalha a partir de um protótipo de uma interface em que se desenvolvem os diagramas de caso de uso baseados nos requisitos levantados [Silva et al, 2007]. De acordo com a metodologia, a primeira fase para o desenvolvimento de *software* é composta no levantamento de requisitos. Esses requisitos específicos do *Framework* PAF-ECF são os mesmos definidos pela Comissão Técnica Permanente do ICMS, através do Ato COTEPE/ICMS 9 de 13 de março de 2013 disponível em http://www1.fazenda.gov.br/confaz/confaz/atos/atos_cotepe/2013/AC009_13.htm. São 53 requisitos funcionais definidos nesse Ato.

Além dos requisitos existentes no Ato em questão, são apresentadas as seguintes funcionalidades que o *framework* proposto contempla: gerenciamento de vendedores, clientes e produtos (com cadastros e consultas em bases de dados); gerenciamento de vendas e emissão de cupom fiscal. Numa tentativa de auxiliar na compreensão do projeto, a Figura 2 ilustra um protótipo de interface utilizado como Aplicação Emissor

de Cupom Fiscal, ou seja: a uma Aplicação ECF que serve como exemplo daquilo que é esperado após a utilização do *framework* proposto pelos desenvolvedores de *software*.

Código	Quantidade	Descrição	Valor
001 2	1x15,00	Porca Sextavada	15,00
002 1	1x12,00	Anel O'Ring	12,00

Figura 2 – Tela do Emissor de Cupom Fiscal

É com base nesse protótipo (Figura 2) que o *framework* proposto neste trabalho é projetado. Dessa forma, parte-se do princípio daquilo que se deseja como resultado final e constrói-se o a parte estrutural do *framework*.

Dentro desse contexto, o *framework* desenvolvido possui um conjunto de classes que foram projetadas de modo a atender às necessidades dos desenvolvedores de aplicações ECF, fornecendo subsídios e implementações que permitem a reutilização de projetos neste âmbito.

A Figura 3 apresenta o Diagrama de Classes do *framework* PAF-ECF.

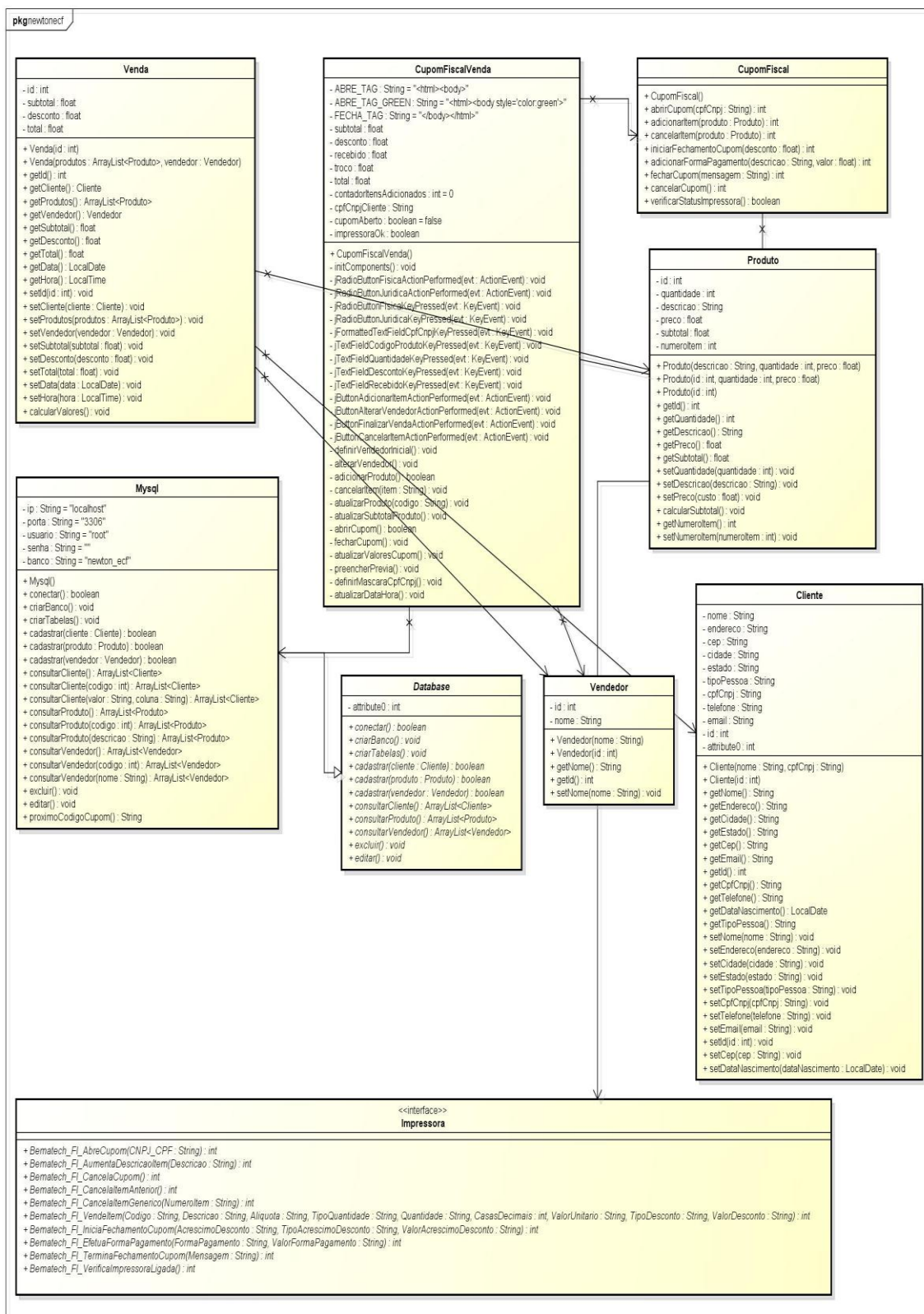


Figura 3 – Digrama de Classes do Framework PAF-ECF

Conforme ilustrado na Figura 3, o diagrama possui todas as classes que foram construídas para o *framework*, de modo a atender cada uma das necessidades

identificadas na interface gráfica prototipada. Esse mapeamento pode ser observado na Figura 4, em que algumas classes estão diretamente relacionadas com as representações gráficas apresentadas na interface.

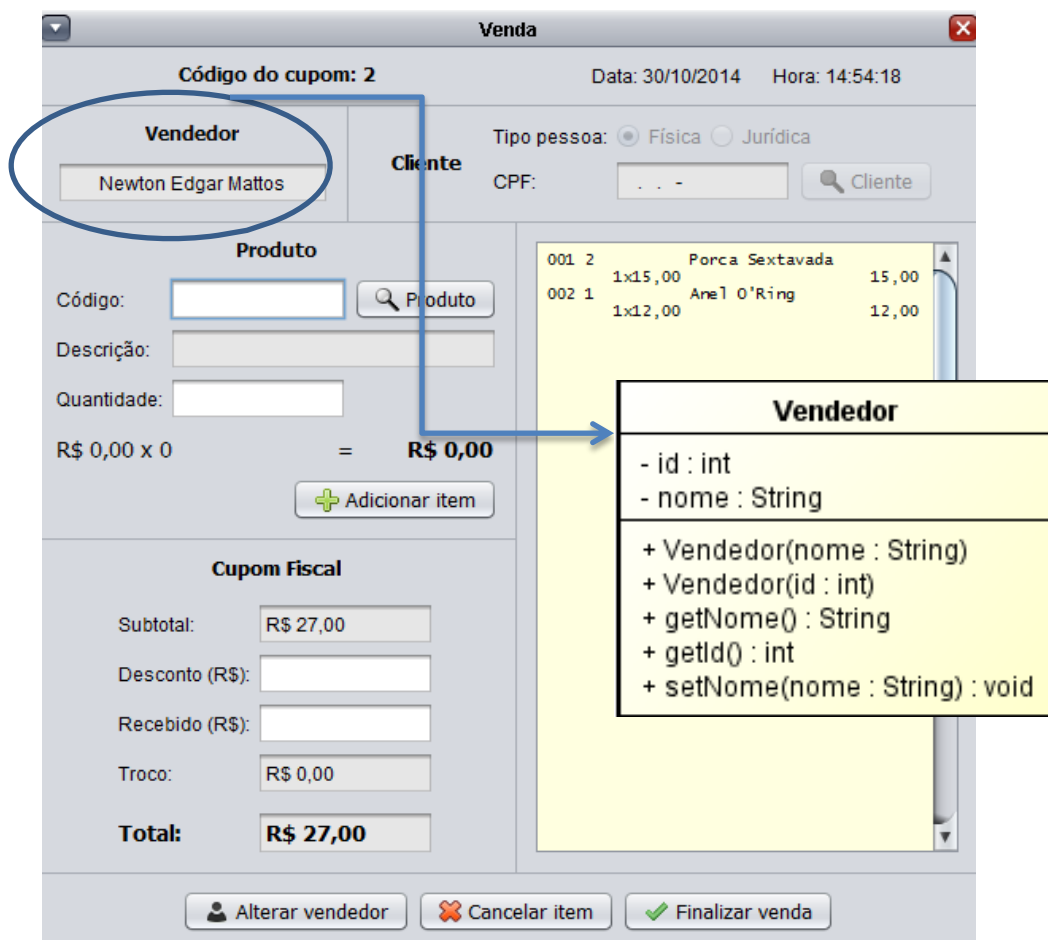


Figura 4 – Identificação e Mapeamento de Classes para o Framework

Na classe abstrata, chamada *Database*, estão declarados os métodos abstratos relacionados com as operações de banco de dados, como conexão, criação do banco e tabelas, cadastros e consultas. Para utilizar essa classe abstrata, basta criar uma classe e utilizar a diretiva *extends* para herdá-la e implementar os métodos abstratos que servirão como base para as interações da aplicação com o banco de dados a ser utilizado. Se o banco de dados da aplicação for *MySQL*, já existe a classe *Mysql* com a implementação adequada para esse tipo de banco. Se for outro banco de dados usado na aplicação, como por exemplo *SQL*, *Oracle*, *Firebird*, basta que o programador/utilizador do *framework* crie uma outra classe herdeira de *Database*.

Quatro das classes contidas no *framework* representam elementos necessários para uma aplicação de emissão de cupom fiscal: *Cliente*, *Produto*, *Vendedor* e *Venda*. As classes *Cliente*, *Produto* e *Vendedor* foram desenvolvidas para facilitar processos como a gestão dessas informações e possuem todos os atributos básicos exigidos em um cupom fiscal. A classe *Venda* se beneficia dessas outras três classes; entre os atributos da classe *Venda*, está o cliente, o vendedor e uma lista de produtos pertencentes à venda. As classes *Cliente*, *Produto*, *Vendedor* e *Venda* se conectam com os métodos da classe *Mysql* (ou com outra implementação da classe abstrata *Database*). No método de

cadastro de produtos, recebe-se por parâmetro um objeto da classe Produto; no método de consultar clientes, o retorno do método é uma *ArrayList* de objetos da classe Cliente.

Outra classe que merece destaque no *framework* é a classe CupomFiscal, que trabalha junto com a interface Impressora. Elas têm como função acessar e utilizar as funções da DLL (*Bemafi32* ou *Bemafi64*) da impressora fiscal *Bematech*, com auxílio da biblioteca JNA (Java Native Access). A interface Impressora declara as funções presentes nessa DLL. Para possibilitar o acesso a essas funções na DLL, ela herda a classe *StdCallLibrary* da biblioteca JNA. A classe CupomFiscal cria um objeto da interface Impressora com a função *Native.loadLibrary*, passando por parâmetros o nome da DLL em String e a classe da interface que irá acessar as funções da DLL, nesse caso, Impressora. Com isso, é possível acessar todas as funções da DLL que foram declaradas na interface e interagir com a impressora fiscal, utilizando somente o objeto da interface Impressora. A classe CupomFiscal já implementa alguns métodos de interações básicas com a impressora fiscal, como abrir um cupom, adicionar itens (recebendo por parâmetro um objeto da classe).

As funções disponíveis para a impressora fiscal da Bematech podem ser encontradas no arquivo de ajuda da DLL *Bemafi32*, como pode ser observado na Figura 5. Na coluna da esquerda estão as funções disponíveis para a impressora fiscal, e na coluna da direita é exibida a documentação de cada função, incluindo informações como parâmetros utilizados, retornos da função e exemplos de uso em diversas linguagens.

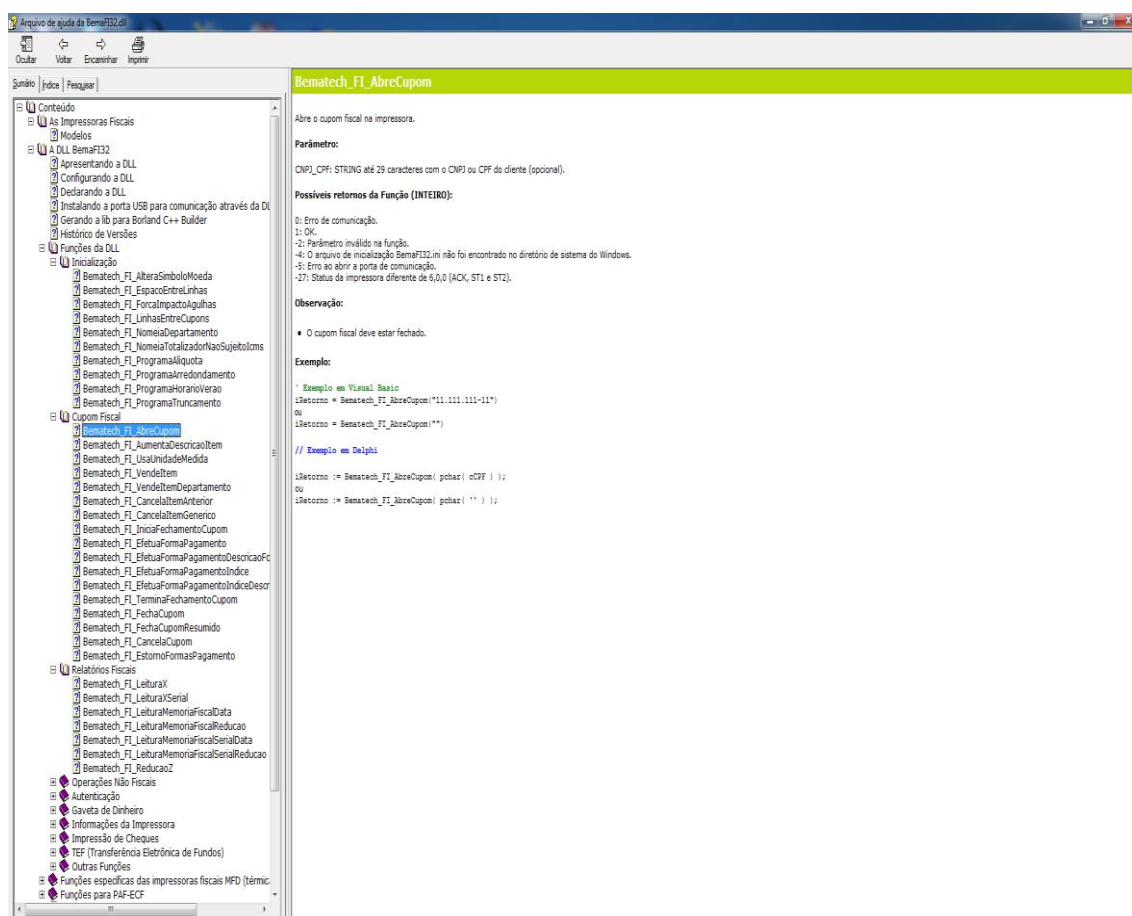


Figura 5 – Tela do arquivo de ajuda da DLL *Bemafi32*.

3.2. Implementação

Na fase de implementação do projeto, foi escolhida a linguagem Java para codificação do *framework*. Essa escolha está alicerçada pelo fato dessa linguagem ser orientada a objetos e se enquadrar às necessidades do projeto (também orientado a objetos). Além disso, ela oferece portabilidade do código gerado, através de *bytecodes* que são interpretados pela máquina virtual Java, ou *Java Virtual Machine* (JVM). Isso permite que uma mesma aplicação de ECF gerada possa ser utilizada por uma variedade de outras plataformas computacionais. Em linhas gerais, o *framework* desenvolvido em Java permite a sua utilização em qualquer sistema operacional que possua uma JVM compatível [Rogers, 1997].

Diante disso, a implementação de todas as classes segue o procedimento de transformar em código todas as classes projetadas e apresentadas na Figura 3. Um exemplo dessa atividade pode ser observada na Figura 5, em que ao lado esquerdo tem-se a classe *Vendedor* projetada e ao lado direito tem-se a mesma classe codificada. Ou seja: a implementação do projeto.

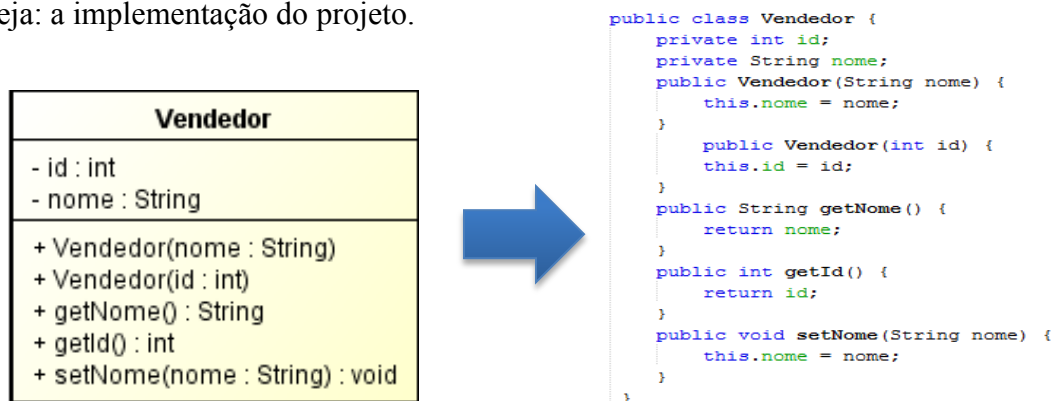


Figura 5 – Implementação das Classes do *Framework*

Adicionalmente, o *framework* possui outras classes complementares que visam ampliar a sua utilização e facilitar, ainda mais, a reutilização de projetos. Essas classes complementares, tais como: *ClienteCadastro*, *ClienteConsulta*, *ProdutoCadastro*, *ProdutoConsulta*, *VendedorCadastro* e *VendedorConsulta*. São essas classes com interfaces gráficas que, normalmente, poderiam ser desenvolvidas por terceiros em suas respectivas aplicações de ECF mas que foram implementadas para permitir que os utilizadores do *framework* já possam partir de um projeto modelo, caso julgarem necessário. A Figura 6 ilustra uma dessas classes, com a interface gráfica para cadastrar um novo produto.

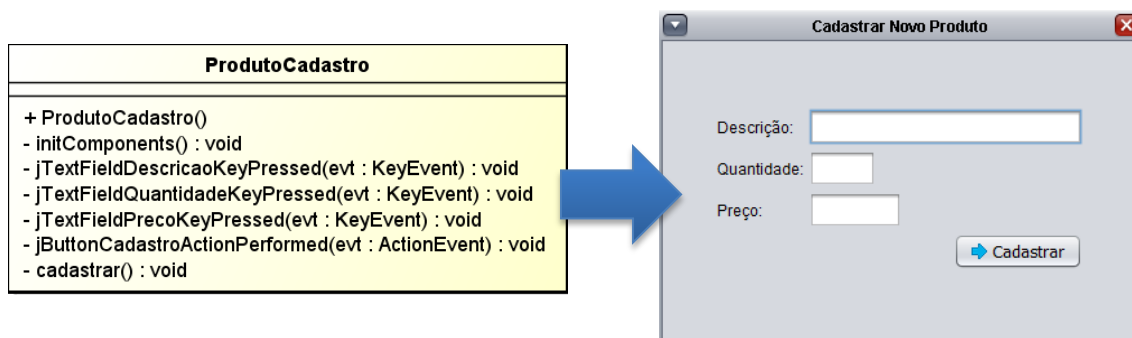


Figura 6 – Exemplo de uma Classe Complementar Implementada

4. Validação

Esta seção apresenta o teste utilizado para validação do *framework* PAF-ECF proposto no âmbito deste trabalho. A ideia para o cenário de testes consistiu em desenvolver uma aplicação ECF a partir da utilização do *framework* desenvolvido. Inicialmente, foi criado um novo projeto Java (*TesteFramework*) utilizando o NetBeans. Em seguida, todo o *framework* foi adicionado ao projeto, ou seja, todas as classes existentes no *framework* foram adicionadas através da importação da biblioteca *NewtonECF.jar*. A partir desse momento foi criada uma classe com o mesmo nome do projeto contendo o método *main*. Nessa classe foi instanciada a classe *TelaPrincipal* do *Framework*, conforme pode ser observado na Figura 7.

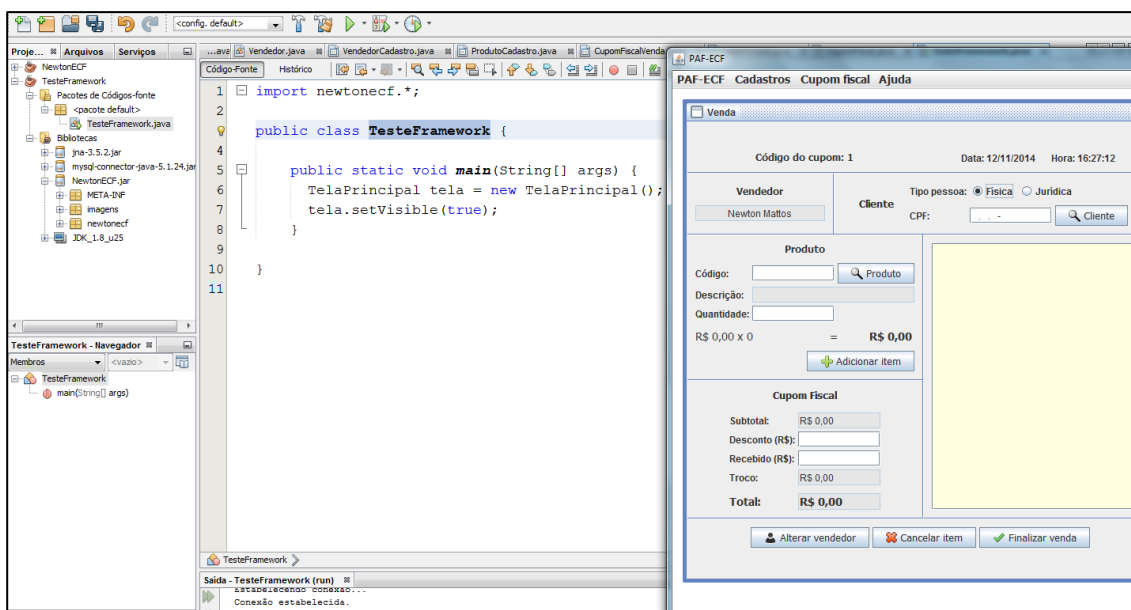


Figura 7 – Aplicação Teste utilizando o *Framework* PAF-ECF

Ao executar a aplicação, foi possível validar a objetividade do *framework*, pois o programa emissor de cupom fiscal, desenvolvido com a utilização deste *framework*, funcionou perfeitamente, emitindo cupom, de acordo com a legislação, conforme a Figura 8.

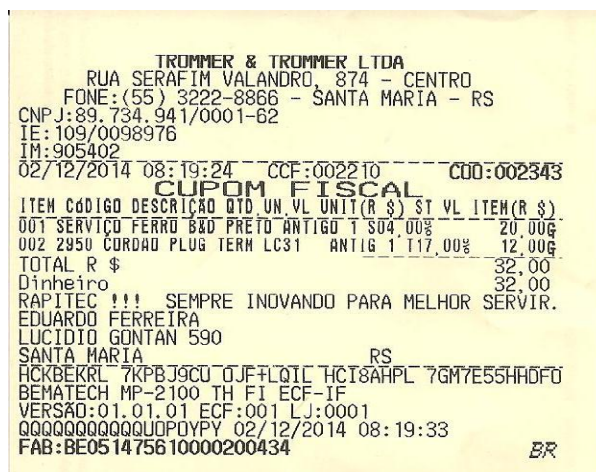


Figura 8 – Cupom Fiscal impresso pelo ECF desenvolvido através do *framework*

5. Conclusões

Este trabalho apresentou a proposta de um *framework* para auxiliar no desenvolvimento de aplicações que necessitam emitir cupom fiscal. Dentro dessa abordagem, o trabalho realizou uma pesquisa bibliográfica sobre o tema contemplando ECF, PAF e *frameworks*. A principal motivação para a realização deste trabalho esteve relacionada com a capacidade da proposta oferecer, para terceiros desenvolvedores de *software*, uma solução que abstraísse toda a complexidade atrelada ao uso de emissores de cupom fiscal, seja do ponto de vista legal ou técnico.

Além disso, disponibilizar um conjunto de classes para essa finalidade (ECF-PAF) através do *framework* buscou-se oferecer uma redução no tempo de desenvolvimento de novas aplicações para aqueles profissionais que estão iniciando nessa área. Outro fator relevante é que o *framework* oferece, ainda, classes de interfaces gráficas que vão ao encontro de um conceito fundamental dos *frameworks*, que é, justamente, a reutilização de projeto. Nesse caso, se o programador/utilizador do *framework* julgar relevante iniciar um novo projeto tendo como base um projeto modelo, este poderá o fazer tranquilamente. Caso contrário, poderá criar uma aplicação ECF tendo como base as demais classes da estrutura.

Adicionalmente, caso novas funcionalidades sejam identificadas ou, ainda, novas tecnologias de banco de dados ou afins sejam detectadas pelos programadores, o *framework* poderá evoluir com as novas regras de negócios, podendo ser adicionadas novas classes ao *framework* existente, garantindo a modularidade, a flexibilidade e a expansibilidade conforme o caso.

Por fim, a construção de uma aplicação exemplo serviu como forma de validar a real utilização do *framework* proposto, bem como a sua simplicidade de uso e, principalmente, a sua potencialidade.

Como trabalho futuro é importante o desenvolvimento das classes para cada impressora fiscal existente no mercado, como por exemplo, *Yanco*, *Sweda*, *Daruma*, *Elgin* e *Epson*. Com isto basta selecionar o tipo de impressora a ser usada pela aplicação para ativar a classe específica.

Outro trabalho importante também é o desenvolvimento de classes para a conexão com os diversos bancos de dados existentes. Por exemplo, se o banco de dados for *Oracle* deverá haver uma classe *Oracle* para fazer a conexão com o banco, criar o banco e as tabelas permitindo acesso rápido e seguro.

Referências

- Bona, Cristina. “Avaliação de processos de software: um estudo de caso em XP e Iconix”. Dissertação (Pós-Graduação em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 2012.
- Erdogmus, H.; Morisio, M.; Torchiano, M. (2005) “*On the effectiveness of the test-first approach to programming.*” IEEE Trans. Softw. Eng., IEEE Press, Piscataway, NJ,

USA, v. 31; Acesso em Setembro de 2014, Disponível em: <http://dx.doi.org/10.1109/TSE.2005.37>

- Gamma, E. et al. Padrões de projeto: soluções reutilizáveis de software orientado a objetos, Porto Alegre: Bookman, 2000. 364 p. ISBN: 85-7307-610-0.
- Gonçalves, Paulo Gilberto. “Manual Fiscal do Usuário de ECF - Emissor de Cupom Fiscal”, Belo Horizonte, 2014.
- Gonçalves, Paulo Gilberto. “Manual do Desenvolvedor de Programa Aplicativo Fiscal Emissor de Cupom Fiscal – PAF ECF”, Belo Horizonte, 2014.
- Gonçalves, Paulo Gilberto, Olímpio, Francisco. “Manual Fiscal do Usuário de ECF” , Secretaria de Estado de Fazenda, Belo Horizonte, 2006.
- Miller L, Sy D, (2009) “*Agile user experience SIG*”, Conference on Human Factors in Computing Systems, Boston.
- Mouta, Albert Eije Barreto. “Manual de Implantação do PAF-ECF”, Brasília, 2014.
- Oliveira, Éric C. M. “Persistência em Java com API JPA”, São Paulo, Casa do Código, 2009.
- Silva, George, Silva, Gilbert, Guimarães, Gabriel, Medeiros, Rodrigo, Rossini, Tiago. “Utilizando ICONIX no Desenvolvimento de Aplicações Delphi”, João Pessoa, 2007.
- Roger, G. F. Framework-based Software Development in C++, New Jersey: Prentice- Hall, 1997. 382 p. ISBN: 0-13-533365-2.
- Shalloway, A.; Trott, J. R. Explicando Padrões de Projeto: uma nova perspectiva em projeto orientado a objeto. Porto Alegre: Bookman, 2004. 328 p. ISBN: 85-363-0403-0.