

DESENVOLVIMENTO DE APLICAÇÃO DE PONTO DE VENDA USANDO JAVAFX

DEVELOPMENT APPLICATION OF AN POINT OF SALE USING JAVAFX

Alex De Souza Markezini¹ Emerson Ceruti Aguinaldo² Osvaldo Cesar Pinheiro de Almeida³

RESUMO

A evolução tecnológica vem oferecendo novas abordagens no processo de desenvolvimento de aplicações. Uma dessas abordagens está na criação de aplicações RIA, com interfaces de usuários mais ricas. A tecnologia JavaFX oferece ferramentas para a criação de interfaces de usuário que tem como objetivo melhorar a experiência do usuário final. Usando essa nova tecnologia, é possível criar, ou remodelar, aplicações que ofereçam novos recursos de usabilidade ao usuário. Por isso, o objetivo desse projeto foi o desenvolvimento de uma aplicação de ponto de venda (PDV), utilizando-se a tecnologia JavaFX. Para isso, foram criadas interfaces gráficas usando o recurso FXML, do JavaFX, e a aplicação foi implementada, utilizando-se a linguagem Java e o sistema de gerenciamento de banco de dados MySQL. Ao final do processo de desenvolvimento, foi criada uma aplicação PDV com todos os recursos funcionais necessários para o processo de venda comercial e que, além disso, possui uma interface intuitiva e que pode ser facilmente customizada, sem que seja necessário alterar a codificação da aplicação.

Palavras-chave: FXML. JavaFX. PDV. RIA.

ABSTRACT

Technological development has been offering new approaches in application development processes. One of these approaches is the development of RIA applications with richer user interfaces. JavaFX technology provides tools for creating user interfaces aiming at improving user experiences. With this new technology it is possible to create or remodel applications which offer new usability features. Therefore, the aim of this project was to develop an application of a sale point (SP) using JavaFX technology. Thus graphical interfaces were created using the FXML (by JavaFX) and implementing was carried out using Java language and database management from MySQL. A SP application was carried out using all necessary functional features for commercial sale processes. Furthermore the process has an intuitive interface which can be easily customized without the necessity of changing application code.

Keywords: FXML. JavaFX. POS. RIA.

¹Graduando do curso de Informática para Negócio da Faculdade de Tecnologia de Botucatu. Av. José Ítalo Bacchi, s/n – Jd. Aeroporto – Botucatu/SP – CEP: 18606-851. E-mail: alexsmarquez@hotmail.com.

²Tecnólogo em Informática para Negócio. Valesys Consultoria e Sistemas. Rua Cel. Fonseca, 2041 – Botucatu/SP – CEP: 18603-495. Site: www.valesys.com.br. E-mail: emerson@valesys.com.br.

³Bacharel em Ciências da Computação, Mestre em Ciências da Computação e Matemática Computacional, Doutorando em Agronomia. Professor da Faculdade de Tecnologia de Botucatu. Av. José Ítalo Bacchi, s/n – Jd. Aeroporto – Botucatu/SP – CEP: 18606-851. E-mail: cesar@fatecbt.edu.br.

1 INTRODUÇÃO

A escolha adequada de uma linguagem de programação e de uma tecnologia está diretamente ligada ao processo de desenvolvimento de uma aplicação. A definição de qual tecnologia utilizar está vinculada a uma série de fatores como facilidade e rapidez na implementação, portabilidade, acessibilidade, usabilidade, entre outros. Um grande fator a ser considerado atualmente é a usabilidade da aplicação, pois a inserção de novas tecnologias possibilitou a criação de aplicativos de uso mais intuitivos, elevando o nível de exigência dos usuários que já experimentaram esses novos recursos.

Na linha de evolução tecnológica de aplicações com usabilidade mais simplificada estão as aplicações *Rich Internet Application* (RIA). Apesar do termo RIA caracterizar aplicações de internet com interfaces ricas, o termo é utilizado de maneira mais ampla, caracterizando de maneira marcante a ideia das aplicações com interfaces ricas. Nesse contexto, a tecnologia RIA tem como característica a convergência entre os paradigmas de interface desktop e *web*, buscando simplificar e uniformizar a interface e a experiência do usuário, independentemente da plataforma, do dispositivo ou do navegador (PREMKUMAR; MOHAN, 2010).

Uma das tecnologias que exploram a característica RIA é a tecnologia JavaFX. O JavaFX é uma plataforma para desenvolvimento de interfaces ricas do lado do cliente. A intenção dessa tecnologia é possibilitar a criação de aplicações RIA que disponibilizem aos usuários uma mesma interface, independente do dispositivo usado. Para isso, apresenta uma série de recursos para criar gráficos, disponibilizar recursos multimídia (como som, imagem ou vídeo), entre outros recursos de interface. Além disso, toda essa plataforma pode ser integrada a recursos já criados em Java, possibilitando o reuso de aplicações já implementadas e, ainda, permite a integração com todos os recursos de alta complexidade da plataforma Java, mantendo o poder de desenvolvimento das aplicações e adicionando o potencial de novos recursos visuais (PREMKUMAR; MOHAN, 2010).

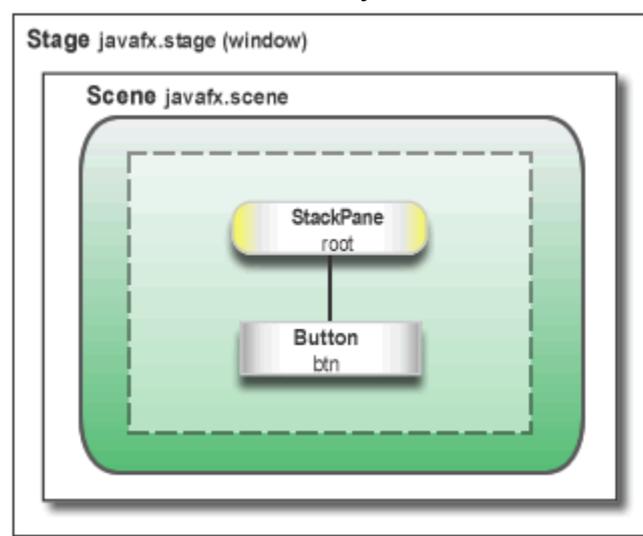
Agregar novas tecnologias ao desenvolvimento de aplicações pode trazer benefícios significativos. Ao desenvolver uma aplicação usando JavaFX, por exemplo, torna-se possível criar aplicações com interfaces ricas, melhorando o visual da aplicação, e pode-se utilizar a aplicação em diferentes plataformas. Além disso, é possível explorar outros benefícios como a integração com os diversos *frameworks* do Java, a fácil manutenção, adaptação e melhoria dos recursos visuais.

Nesse contexto, o objetivo deste trabalho foi o desenvolvimento de uma aplicação RIA, a ser utilizada como um ponto de venda (PDV), usando a tecnologia de desenvolvimento JavaFX e explorando suas características técnicas para implementar um ponto de venda, que possua uma interface rica, com interoperabilidade entre plataformas e que reduza o trabalho de modificação e manutenção do sistema.

2 TECNOLOGIA JAVAFX

O JavaFX permite criar interfaces gráficas a partir de códigos escritos diretamente em Java. Para isso, é necessário utilizar um gerenciador de aplicação, acessado normalmente por meio da classe “javafx.application.Application”. Dessa maneira, o gerenciador será uma classe principal do sistema JavaFX que deve estender a classe “Application” e sobrecarregar o método “start()”. Esse método é responsável por fornecer um elemento “stage”, que define um container de alto nível hierárquico, em que serão inseridas as cenas (“scene”). Essas cenas são os containers onde serão inseridos os elementos de representação gráfica da aplicação (CHAPPELL et al., 2013). A Figura 1 ilustra um esquema de uma aplicação JavaFX para inserir um elemento gráfico que representa um botão (com os dizeres “Say ‘Hello World’”). A Figura 2 ilustra a aplicação do esquema da Figura 1.

Figura 1 – Ilustração da composição de uma aplicação JavaFX para apresentar um botão com os dizeres: “Say ‘Hello World’”



Fonte: CHAPPELL et al., 2013.

Figura 2 – Aplicação JavaFX representado pelo esquema da Figura 1



Fonte: CHAPPEL et al., 2013.

Apesar da possibilidade de se criar uma interface gráfica usando JavaFX diretamente com a codificação estática na linguagem Java, existem outras possibilidades de se criar essa interface usando outras linguagens como Groovy, Scala ou Visage, ou até mesmo usando uma linguagem de marcação de texto, o FXML. O FXML é basicamente um XML (W3C, 2013) contendo as especificações dos elementos gráficos que deverão ser criados na cena da aplicação. A especificação da interface gráfica usando FXML permite manter a aplicação e a interface com um acoplamento fraco e, também, permite que sejam realizadas alterações na interface gráfica sem que seja necessário alterar a codificação da aplicação. Dessa maneira, o processo de carregamento da interface gráfica ocorre de maneira dinâmica, com a aplicação fazendo a carga das interfaces (representadas por arquivos FXML), à medida que for necessário (WEAVER et al., 2012).

O FXML possibilita a criação de todo tipo de interface de usuário. Adicionalmente, oferece grande facilidade na criação de interfaces com cenas gráficas complexas e/ou numerosas, formulários de entrada de dados e animações. Ainda, pode-se destacar alguns benefícios do uso do FXML para a criação de interfaces de usuário (FEDORTSOVA, 2013):

- Facilidade no desenvolvimento, manutenção e teste das interfaces gráficas;
- O FXML não é compilado, portanto, não é necessário recompilar após modificá-lo;
- O uso de mecanismos de localização para modificar o idioma corrente é mais simples;

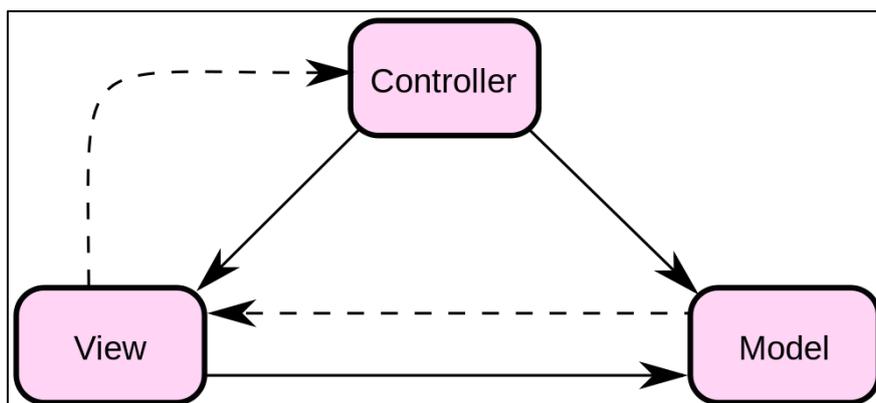
- O FXML pode ser usado com outras linguagens como Groovy, Scala e Clojure;
- O FXML pode ser usado como serviço, ou tarefa, ou objeto de domínio e permite o uso de linguagens de *script*, como o JavaScript.

Um maneira ágil de criar interfaces gráficas utilizando FXML é por meio da ferramenta JavaFX Scene Builder. Essa é uma ferramenta de designer que gera o código fonte da interface gráfica em FXML. Além disso, permite fazer a ligação entre a interface gráfica e a aplicação. O uso da ferramenta segue o padrão de arrastar e soltar, permitindo que o desenvolvedor tenha uma rápida percepção do resultado da criação da interface visual (CASTILLO, 2013).

A criação de aplicações JavaFX usando o recurso do FXML permite explorar de maneira bastante eficiente o modelo de desenvolvimento conhecido como “Model-View-Controller” (MVC) (traduzido como Modelo-Visão-Controleador). Esse modelo envolve a interação de três estruturas (ilustrada na Figura 3) (TEMPLE et al., 2004):

- **Model** – Representa o comportamento e modelo de dados do domínio da aplicação;
- **View** – Representa a interface de interação com o usuário;
- **Controller** – Representa o controle de fluxo e lógica da aplicação.

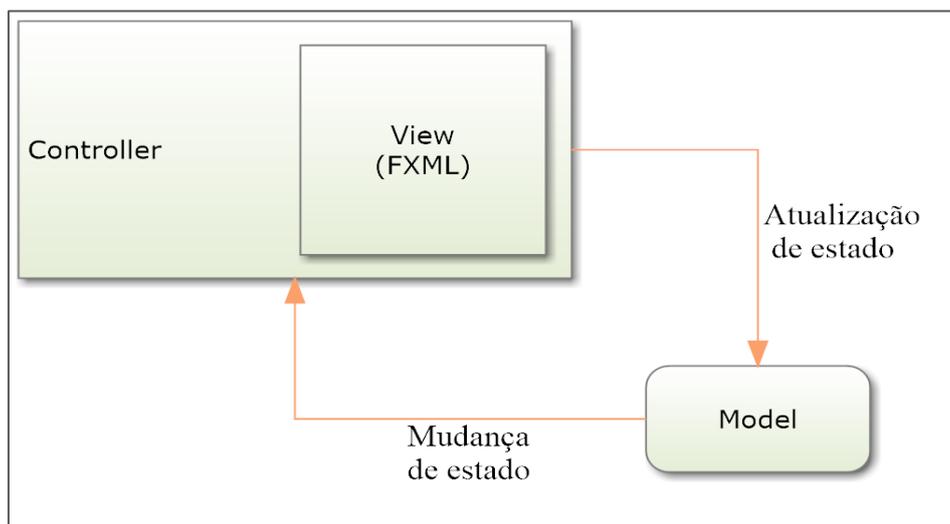
Figura 3 – Ilustração da interação do modelo MVC – “Model-View-Controller”.



Observando-se uma aplicação JavaFX sob a ótica do modelo MVC, o FXML representa a descrição da interface do usuário, ou seja, o “View”. O “Controller” é definido como uma classe Java, normalmente como uma classe de inicialização, e definida no FXML como controlador. E o “Model”, como os dados da aplicação, são

definidos como classes Java, que farão a interação entre o controlador e a interface de usuário, onde serão exibidos esses dados (FEDORTSOVA, 2013). A Figura 4 ilustra a interação de uma aplicação JavaFX sob a perspectiva do modelo MVC.

Figura 4 – Interação de uma aplicação JavaFX sob a perspectiva do modelo MVC



Ao se criar uma interface usando o FXML, é necessário vincular o controlador dessa interface. Para isso, é utilizado o atributo “fx:controller” que definirá qual será o controlador gráfico da aplicação para FXML criado. A definição do controlador pelo FXML limita a capacidade de explorar os recursos do controlador. Uma alternativa criada é o uso da tag “fx:root” no FXML, que torna possível criar um controlador customizado para o FXML. Usando a classe “FXMLLoader”, que possui os métodos “setRoot()” e “setController()”, é possível carregar um FXML, definir a raiz do objeto e o controlador da interface gráfica. Esse processo de construção oferece maior flexibilidade no desenvolvimento de aplicações, possibilitando reuso dos controladores (FEDORTSOVA, 2013).

3 MATERIAIS E MÉTODOS

Para o desenvolvimento da aplicação de ponto de venda (PDV), foi necessário inicialmente especificar as principais funcionalidades do sistema. O propósito básico foi permitir que a aplicação operasse uma venda de produtos a um determinado cliente. Por essa razão, fez-se necessário ter acesso às informações básicas sobre os produtos (como valor, desconto, incidência de impostos, entre outras), os clientes, os usuários e as

formas de pagamento aceitáveis. Essas informações básicas foram acessadas de uma base já existente; portanto, não foi concebido o processo de cadastro dessas informações nesse projeto. Além disso, foi especificado o processo de interação com impressora fiscal, essencial em uma aplicação PDV. Com os processos especificados foi desenvolvido o modelo da base de dados para armazenar as informações pertinentes a esses processos. A partir do modelo, foi criada a base de dados utilizando-se o sistema gerenciador de banco de dados (SGBD) MySQL (MYSQL, 2013).

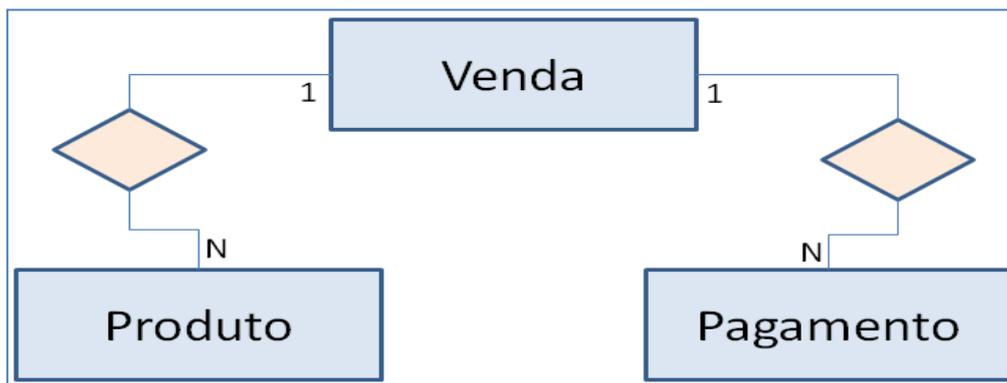
A aplicação foi desenvolvida usando o ambiente de desenvolvimento integrado NetBeans. Foram utilizados a linguagem orientada a objetos Java, os recursos da plataforma JavaFX, para a implementação da interface de usuário, e o *framework* Hibernate, para a conexão com o banco de dados. Durante o desenvolvimento, foi aplicado o modelo MVC, pelo qual a interface gráfica foi desenvolvida usando a ferramenta JavaFX Scene Builder. Os arquivosFXML criados foram vinculados à aplicação por meio de controladores customizados. Foram criadas todas as interfaces necessárias para a execução do processo de venda no PDV.

Como uma aplicação PDV precisa de uma impressora fiscal, foram implementadas em Java as funcionalidades de integração com a impressora Bematech, modelo MP-2100 TH FI (BEMATECH, 2013). Dessa maneira, conforme são executados os processos de venda na aplicação, as informações correspondentes são automaticamente enviadas à impressora fiscal. Por se tratar de um desenvolvimento inicial, foi utilizado um emulador dessa impressora fiscal para testar a aplicação.

4 RESULTADOS E DISCUSSÃO

Como resultado do processo de desenvolvimento foi gerada uma aplicação de ponto de venda de produtos. O modelo de dados gerado é baseado em três tabelas: **Venda**, que representa as informações gerais de cada venda realizada; **Produto**, que representa as informações detalhadas dos produtos vendidos para cada venda; **Pagamento**, que representa as informações detalhadas sobre as formas de pagamento de cada venda. A Figura 5 ilustra o modelo da base de dados definida para a aplicação do PDV.

Figura 5 – Modelo da base de dados específica do processo de venda do PDV



Ao iniciar-se a aplicação, é exibida ao usuário a tela inicial do PDV (ilustrada pela Figura 6), onde é requisitada ao usuário uma autenticação. Deve ser informado o nome e a senha de um usuário habilitado a utilizar o sistema; se o nome e a senha não forem informados, será emitida uma mensagem de usuário ou senha incorreta. Caso o usuário e a senha sejam válidos, será exibida a tela principal da aplicação (ilustrada pela Figura 7). Na tela principal, são apresentadas as informações sobre o usuário conectado, os acessos às principais funcionalidades (menu principal) e os acessos rápidos (ícones de acesso rápido).

Figura 6 – Tela inicial de autenticação de usuário do PDV



O menu principal possui o acesso às funcionalidades da aplicação, sendo que no menu “PDV” são disponibilizadas as ações de configuração do PDV e impressora fiscal. As inicializações envolvem o cadastro inicial de impostos e outras informações na impressora fiscal. Assim, antes de utilizar a impressora fiscal pela primeira vez, é necessário cadastrar todas as alíquotas de impostos de ICMS e ISS que poderão ser utilizadas durante o processo de venda de produtos. Também é necessário cadastrar as

formas de pagamento que podem ser utilizadas. Após todas as informações serem cadastradas, a impressora pode ser utilizada em uma venda.

Figura 7 – Tela principal do PDV com os acessos às principais funcionalidades da aplicação

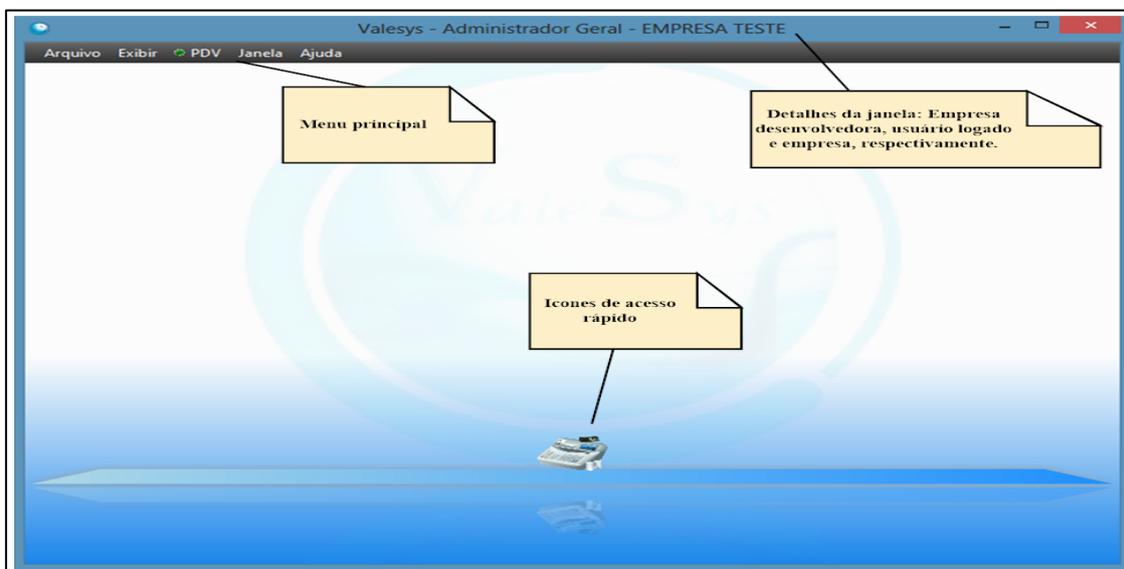
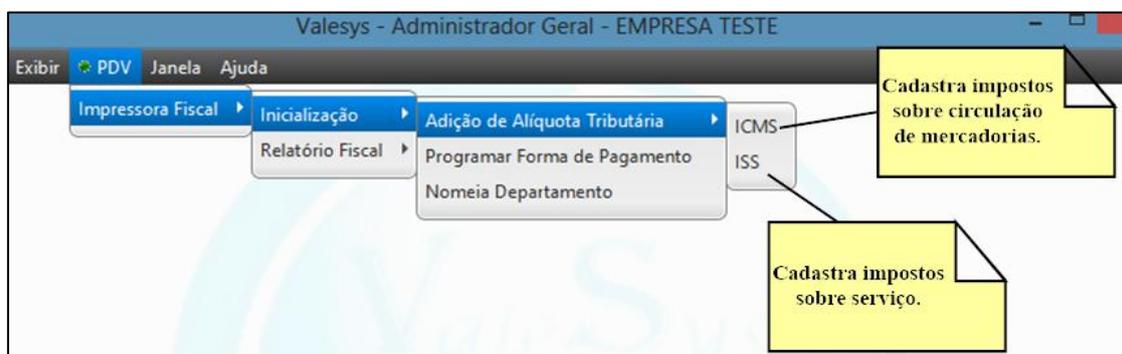
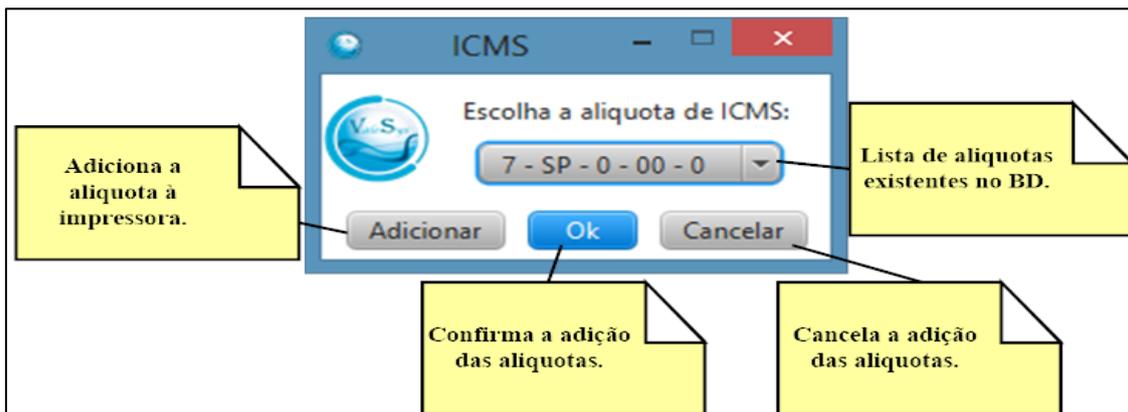


Figura 8 – Ilustração do menu de configuração inicial do PDV e impressora fiscal



A Figura 9 ilustra a tela de cadastro das alíquotas de imposto de ICMS, que é similar a tela de cadastro de ISS. Nessa tela, são destacadas todas as alíquotas de ICMS encontradas nos produtos da base e que ainda não estão cadastradas na impressora fiscal. O usuário deve escolher cada uma das alíquotas desejada e adicionar na impressora fiscal. Assim que todas estiverem cadastradas, o usuário irá confirmar no botão “OK”. Todas as alíquotas devem ser cadastradas, para que não ocorrerá nenhum erro na impressora no momento da venda. De maneira similar, é disponibilizada a tela de cadastro de formas de pagamento.

Figura 9 – Tela de cadastro das alíquotas do imposto de ICMS.



A partir do menu “PDV”, também podem ser acessados os relatórios fiscais da impressora. A Figura 10 ilustra os menus de acesso aos relatórios fiscais. Podem ser gerados os relatórios de “Leitura X”, “Leitura X Serial” e “Redução Z”. Esses relatórios são gerados pela impressora fiscal, porém é preciso solicitar a sua geração através da aplicação. Ao solicitar a impressão de um desses relatórios a impressora irá imprimir as informações sobre vendas armazenadas. O relatório de “Redução Z”, por exemplo, só pode ser gerado uma vez ao dia, fazendo com que todas as informações de venda do dia sejam sumarizadas em um documento fiscal. A Figura 11 ilustra, de maneira parcial, as informações que são impressas, quando solicitada a geração da “Redução Z”.

Figura 10 – Tela com os menus de acesso aos relatórios fiscais da impressora

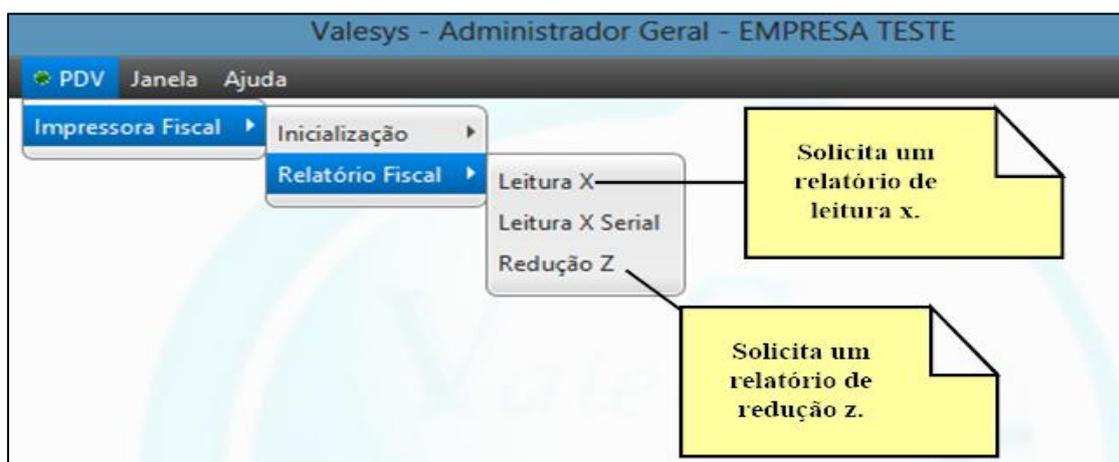


Figura 11 – Ilustração de uma emissão de relatório de “Redução Z” da impressora fiscal

```

Emul2100 1.01
Emulador da Impressora Fiscal Termica Bematech.

BEMATECH IND E COM DE EQUIP ELETRONICOS S/A
Estrada de Santa Candida, 203 Santa Candida
82.630-400 Curitiba - Parana
CNPJ:82.373.077/0001-71
IE:10181465-30 UF:PR
IM:ISENTO

15/10/2013 22:12:05 000:000004

REDUÇÃO Z
SEM MOVIMENTO FISCAL
-----CONTADORES-----

● ● ●

-----TOTALIZADORES-----
TOTALIZADOR GERAL: 0,00
VENDA BRUTA DIÁRIA: ***.***.***.*** **
CANCELAMENTO ICMS: 0,00
DESCONTO ICMS: 0,00
Total de ISSQN: 0,00
CANCELAMENTO ISSQN: 0,00
DESCONTO DE ISSQN: 0,00
VENDA LÍQUIDA: 0,00
ACRÉSCIMO ICMS: 0,00

● ● ●

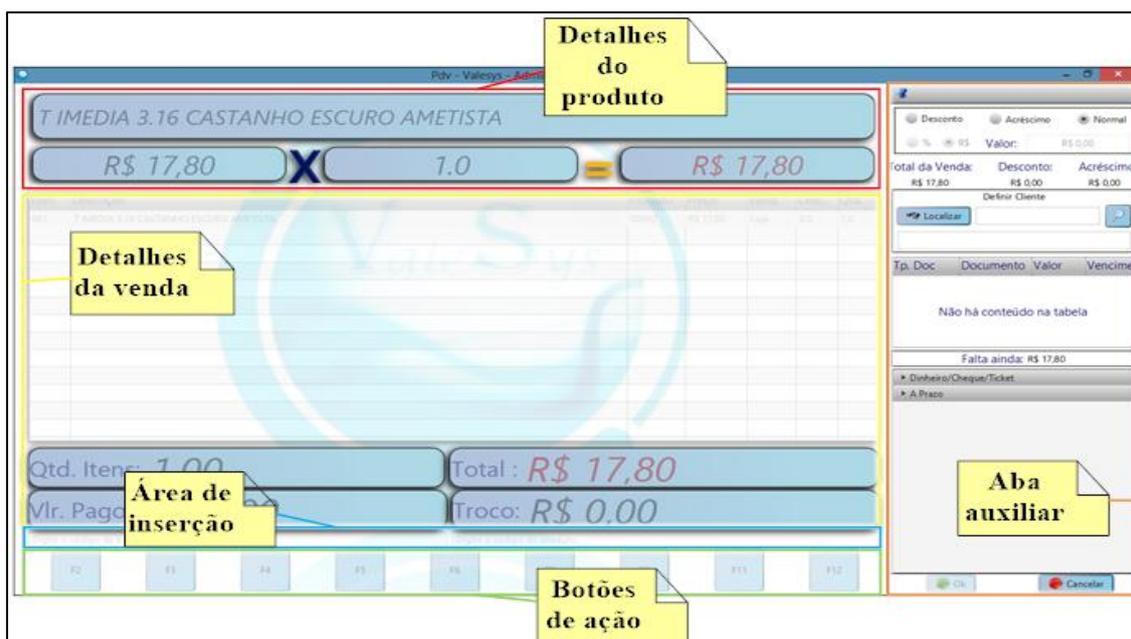
---MEIOS DE PAGAMENTO---
NP Meio Pagamento Valor Acumulado (R$)
01 Dinheiro 0,00
-----
Comprovante Não Emitido: 0000
Tempo Emitindo Doc. Fiscal: 00:00:00
Tempo Operacional: 2843 dias 07:07:04
Qtd. Reduções Restantes: 4649
Número série MFD: 8 MFD
-----
BEMATECH MP-2100 TH F1 ECF-IF
VERSÃO:01.00.01 ECF:001 LJ:0001
000000000000000000 15/10/2013 22:12:05
FAB:EMULADOR BR
-----

```

O menu “Janelas” disponibiliza o acesso às ações da aplicação. Essas ações também podem ser acessadas pelos ícones de acesso rápido. É a partir de um item desse menu que é acessada a tela de venda do PDV. Essa tela, ilustrada pela Figura 12, pode ser dividida em 5 seções importantes:

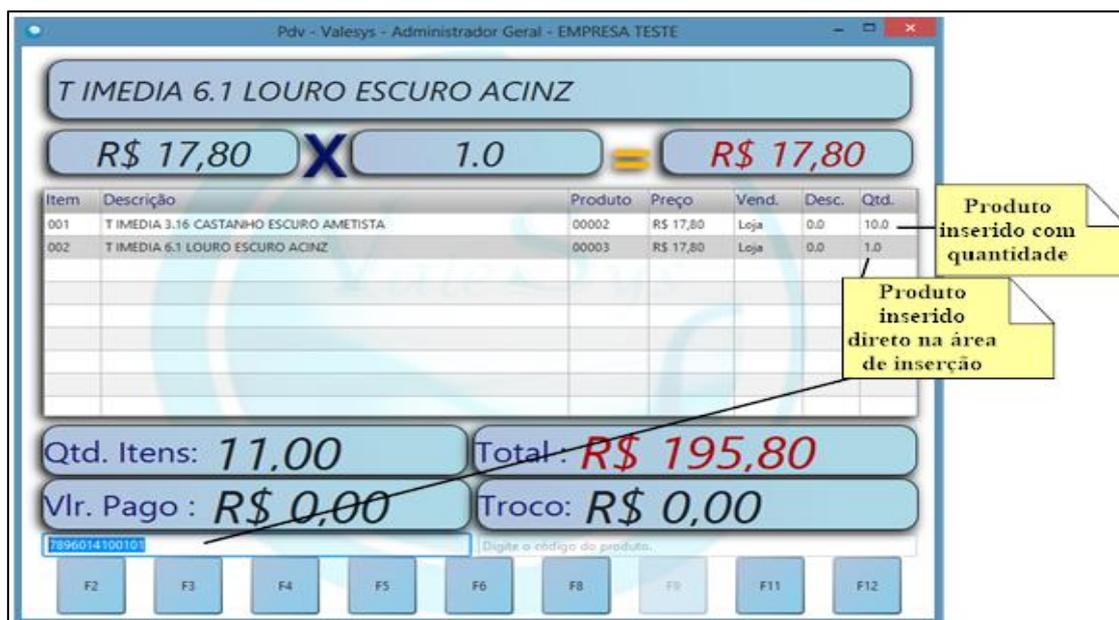
1. Seção de detalhes do produto que está sendo vendido;
2. Seção de detalhes de venda (onde serão exibidos todos os produtos já cadastrados na venda corrente);
3. Seção de inserção de dados para escolha do produto que será vendido;
4. Seção de botões de ações, que disponibiliza o acesso às ações que podem ser tomadas na venda;
5. Seção auxiliar, onde são disponibilizadas informações auxiliares como forma de pagamento da venda que está sendo realizada.

Figura 12 – Tela de venda do PDV, com detalhamento de suas 5 principais seções



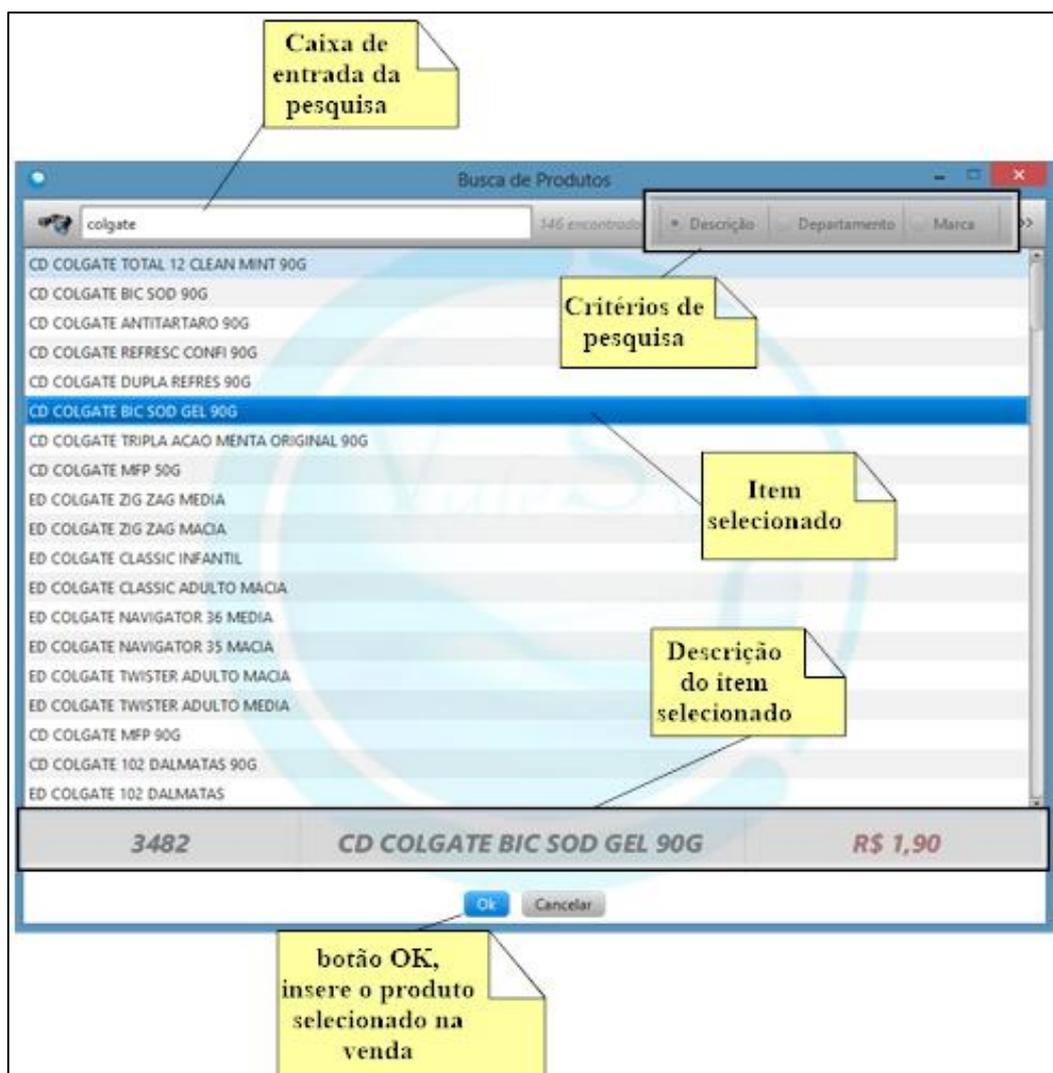
Para iniciar o processo de venda, basta o usuário cadastrar um primeiro produto para venda. Para isso, o usuário pode acessar a opção do botão “F2”, que lhe permitirá inserir a quantidade de produtos a serem vendidos. Outra opção é inserindo o código de barras do produto na área de inserção, para a venda de um único item. A Figura 13 ilustra uma venda onde foram inseridos dois produtos, usando primeiro a função “F2” e depois diretamente com o código de barra.

Figura 13 – Tela de venda com o cadastro de dois produtos com 10 e 1 unidade, respectivamente



O PDV possui também uma opção de busca pelo produto a ser vendido, a partir da função “F12”. Acessando essa funcionalidade, o usuário do PDV pode fazer uma busca pelo produto da base usando diversos critérios como descrição, departamento, marca ou código do produto. Ao fazer essa busca, é exibida uma lista de produtos que se encaixam nos critérios de busca. Ao navegar por essa lista de produtos são exibidos os detalhes de código e preço do produto previamente selecionado. Quando o usuário escolher o produto desejado, basta clicar no botão “OK”, o que irá inserir o produto selecionado diretamente na lista dos produtos vendidos. A opção “F12” também pode ser usada apenas para consulta de preços. Para isso, basta o usuário selecionar a opção “Cancelar” após a consulta. A Figura 14 ilustra a tela de busca dos produtos cadastrados na base principal.

Figura 14 – Tela de busca dos produtos cadastrados na base principal



Antes de iniciar uma venda, o usuário do PDV pode utilizar a função “F9” para cadastrar o CPF do cliente. Isso serve para incluir no cupom fiscal o CPF do cliente para uso da nota fiscal paulista (para o estado de São Paulo). Essa opção só pode ser realizada antes do início da venda, já que essa informação deve ser fornecida à impressora fiscal no momento da abertura do cupom. Após a abertura da venda, a opção “F9” fica inativa até a sua finalização. A Figura 15 ilustra a tela de entrada de informação do CPF do cliente. A Figura 16 ilustra a informação apresentada no cupom fiscal gerado pela impressora.

Figura 15 – Tela de entrada do CPF do cliente da venda

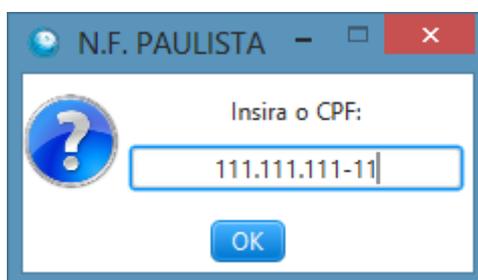


Figura 16 – Cupom fiscal com o CPF do cliente em destaque



Após todos os produtos serem inseridos, o usuário pode finalizar a venda. Para isso, basta usar uma das opções “F3” ou “F5”. Pela opção “F3” deve ser informada a forma de pagamento escolhida como à vista, em dinheiro ou cartão, ou a prazo, em cheque ou cartão. As formas de pagamento dependem do que estiver cadastro no sistema previamente. Já pela opção “F5” a venda é fechada com pagamento à vista. Essa opção realiza o processo de fechamento da venda de maneira menos burocrática e, por isso, mais rápida.

Todo o processo de venda no PDV está vinculado ao uso de uma impressora fiscal. Para que seja possível usar qualquer tipo de impressora fiscal no PDV desenvolvido, foi necessário criar um modelo genérico de conexão do PDV com a impressora. Assim, foi criada uma interface de conexão onde são especificadas todas as funcionalidades de interação entre a aplicação e o PDV. Posteriormente a isso, foi implementada uma classe de conexão à impressora que estava sendo usada durante os testes. Essa classe faz a ligação com as bibliotecas disponibilizadas pela fabricante. Para utilizar outra impressora, basta fazer a implementação da classe de conexão específica, ajustando-se às especificidades da biblioteca do fabricante. A Figura 17 ilustra a interação entre a interface e as bibliotecas das impressoras. A Figura 18 ilustra a impressora do emulador utilizado durante os testes de desenvolvimento do PDV.

Figura 17 – Interação entre a interface do sistema e as bibliotecas de impressora fiscal

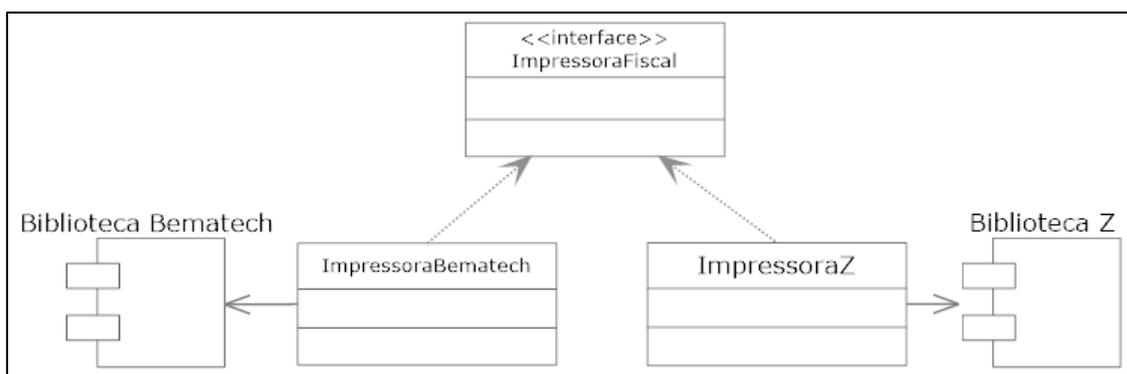


Figura 18 – Emulador da impressora fiscal Bematech, modelo MP-2100 TH FI



5 CONCLUSÕES

Ao concluir-se o projeto, foi originada uma aplicação RIA que utiliza os recursos do JavaFX com a criação de interface de usuário usando FXML. A aplicação de um ponto de venda (PDV) criada pode ser instalada em ambientes comerciais para realizar a operação de venda de produtos. Essa ferramenta oferece os principais recursos usados nessa operação comercial, agilizando-se o processo e a interação entre o comerciante e seus clientes.

O uso da nova tecnologia JavaFX para a geração de aplicações RIA oferece uma ótima ferramenta de desenvolvimento, pois facilita a criação de interfaces intuitivas e que podem ser criadas e alteradas de maneira desvinculada à codificação. Isso possibilita a customização da interface de usuário para melhor adequar às suas experiências e, assim, facilitar o aprendizado e uso das novas aplicações. Ainda, o uso da plataforma Java possibilita que a aplicação seja utilizada em qualquer plataforma desktop como Windows ou Linux.

REFERÊNCIAS

BEMATECH. **Emulador Impressora Fiscal**: versão 1.01. BEMATECH. 2013. Disponível em: <http://www.bematech.com.br/suporte/dwn_fisc_windows.asp>. Acesso em: jul. 2013.

CASTILLO, C.. **JavaFX: Getting Started with JavaFX Scene Builder**, Release 1.1. ORACLE. 2013. Disponível em: <http://docs.oracle.com/javafx/scenbuilder/1/get_started/jsbpub-get_started.pdf>. Acesso em: set. 2013.

CHAPPELL, G.; POTTS, J.; HILDEBRANDT, N.. **JavaFX: Getting Started with JavaFX**, Release 2.2.40. ORACLE. 2013. Disponível em: <http://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.pdf>. Acesso em: set. 2013.

FEDORTSOVA, I.. **JavaFX: Mastering FXML**, Release 2.2. ORACLE. 2013. Disponível em: <http://docs.oracle.com/javafx/2/fxml_get_started/jfxpub-fxml_get_started.pdf>. Acesso em: set. 2013.

MYSQL. **MySQL Server**: versão 5.5. ORACLE. 2013. Disponível em: <<http://www.mysql.com/downloads/mysql>>. Acesso em: ago. 2013.

PREMKUMAR, L; MOHAN P.. **Beginning JavaFX**. New York, Apress. 2010.

TEMPLE, A.; MELLO, R. F.; CALEGARI, D. F.; SCHIEZARO, M.. **Programação Web com Jsp, Servlets e J2EE**. 2014. ISBN: 85-905209-1-9.

WEAVER, J. L.; GAO, W.; CHIN, S.; IVERSON, D.; VOS, J.. **Pro JavaFX 2: A Definitive Guide to Rich Clients with Java Technology**. New York, Apress. 2012.

W3C. **Extensible Markup Language (XML)**. 2013. Disponível em: <<http://www.w3.org/XML>>. Acesso em: set. 2013.