

Universidade Estadual de Goiás
Unidade Universitária de Itaberaí

Danilo Ferreira Adorno
Leonardo Francisco Dos Santos

CRM - GESTÃO DE RELACIONAMENTO COM CLIENTE: SISTEMA
FORÇA DE VENDAS

ITABERAÍ
2017

Danilo Ferreira Adorno
Leonardo Francisco Dos Santos

**CRM - GESTÃO DE RELACIONAMENTO COM CLIENTE: SISTEMA
FORÇA DE VENDAS**

Trabalho Final de Curso apresentado à Universidade Estadual de Goiás, Unidade Universitária de Itaberaí, como requisito parcial para a conclusão do curso de graduação em Sistema de Informação, sob orientação do Professor Msc. Washington Duarte da Silva.

ITABERAÍ
2017

Dedicamos este trabalho a todos que contribuíram direta ou indiretamente em nossa formação acadêmica.

AGRADECIMENTOS

Agradecemos a todos que contribuíram no decorrer desta jornada, em especialmente:

A Deus, a quem devemos nossas vidas.

A nossa família que sempre nos apoiou nos estudos e nas escolhas tomadas.

“Há três caminhos para o fracasso: não ensinar o que se sabe, não praticar o que se ensina, e não perguntar o que se ignora.”

(São Beda)

RESUMO

Com a expansão da internet e dos dispositivos móveis as rotinas empresariais vêm automatizando-se cada vez mais, a partir desse contexto e visando auxílio nas empresas ligadas ao processo de vendas surgiu a ideia da aplicação, que consiste em um software mobile em Android para envio e recepção de informação dos agentes atuantes no processo de vendas a partir de comunicação por *web service*, relação base local da aplicação pela base do software da empresa, com o objetivo de garantir o máximo aproveitamento de tempo por parte do vendedor externo, maximização dos lucros por parte da empresa e a satisfação do atendimento por parte do cliente. Todos os aspectos da aplicação foram desenvolvidos de forma a proporcionar a melhor experiência aos usuários, foram utilizados os princípios da UML no desenvolvimento da arquitetura do projeto o que auxiliou na estruturação das classes, e todo o embasamento teórico obtido por meio de pesquisas bibliográficas.

Palavras-chave: Mobile, Android, Web service.

ABSTRACT

With the expansion of the internet and mobile devices business routines have been increasingly automated, from that context and seeking help in companies linked to the sales process came the idea of the application, which consists of mobile software in Android for sending and reception of information from the agents involved in the sales process from communication via web service, local base relation of the application by the company software base, with the purpose of guaranteeing maximum use of time by the external vendor, profit maximization by the company and customer satisfaction. All aspects of the application were developed in order to provide the best user experience, the UML principles were used in the development of the project architecture, which helped in the structuring of the classes and all the theoretical basis obtained through bibliographic research.

Keywords: Mobile, Android, Web service.

LISTA DE FIGURAS

Figura 1. Ciclo do CRM, Fonte: Newell, 2006, p. 56	18
Figura 2. Diagrama de caso de uso - sistema gestão prestadores de serviço.....	28
Figura 3. Exemplo Diagrama de Classes Sistema Agendamento de Prestador de Serviços	29
Figura 4. Exemplo Diagrama de Sequência - Sistemas de agendamento de consultas	31
Figura 5. Modelo Banco de dados Relacional - Sistema Bibliotecário.....	33
Figura 6. Exemplo MER - Sistema Imobiliário.....	35
Figura 7. Comunicação web service	37
Figura 8. Modelo do Diagrama de caso de Uso	43
Figura 9. Diagrama de Classes.....	49
Figura 10. Diagrama de Sequência.....	50
Figura 11. Tela Splash.....	51
Figura 12. Tela Principal	51
Figura 13. Tela Insight	52
Figura 14. Tela Configuração.....	52
Figura 15. Tela Detalhe Cliente	53
Figura 16. Tela Detalhe Cliente Títulos.....	53
Figura 17. Tela Novo Cliente	54
Figura 18. Tela Novo Pedido - Seleção Cliente.....	54
Figura 19. Tela Novo Pedido - Últimos Pedidos.....	55
Figura 20. Tela Novo Pedido - Seleção Produto	55
Figura 21. Tela Novo Pedido - Finalização Pedido	56
Figura 22. Tela Novo Pedido - Finalização Pedido - Adição Item.....	56
Figura 23. Tela Novo Pedido - Finalização Pedido – Confirmação.....	57
Figura 24. Tela Consulta Cliente.....	57
Figura 25. Tela Consulta Produtos.....	58
Figura 26. Classes Aplicação - Android Studio	59
Figura 27. Layout Aplicação - Android Studio	60

SUMÁRIO

INTRODUÇÃO.....	11
CAPÍTULO 1 - CUSTOMER RELATIONSHIP MANAGEMENT (CRM).....	14
1.1 - Softwares de apoio ao CRM.....	17
1.2 - Ciclo do CRM.....	18
1.3 - Sistemas Forças de Vendas	20
1.4 - CRM Operacional.....	21
CAPÍTULO 2 - RECURSOS TECNOLÓGICOS.....	23
2.1 - Sistema operacional Android.....	23
2.2 - Aplicações Android	25
2.2.1 - Android SDK	25
2.2.2 - Estrutura do projeto Android	25
2.2.3 - APK.....	26
2.2.4 - Mecanismos para publicação de APK	26
2.3 - Análise e projeto do sistema.....	27
2.3.1 - UML.....	27
2.3.2 - Diagrama de Caso de Uso.....	28
2.3.3 - Diagrama de Classe.....	29
2.3.4 - Diagrama de Sequência.....	30
2.4 - Banco de dados.....	31
2.4.1 - Bancos de dados relacionais	32
2.4.2 - Linguagem SQL.....	33
2.4.3 - Modelo Entidade-Relacionamento (MER)	34
2.4.4 - SQLite	35
2.5 - Linguagem de programação Java	36
2.6 - Web Service.....	37

CAPÍTULO 3 - ESTUDO DE CASO	39
3.1 - Identificação do problema	39
3.2 - Objetivos do trabalho	40
3.2.1 - Geral.....	40
3.2.2 - Específicos	40
3.3 - Justificativa.....	40
3.4 - Requisitos funcionais.....	41
3.5 - Requisitos não funcionais.....	42
3.6 - Diagramas de Caso de Uso.....	43
3.7 - Descrição dos casos de usos	43
3.8 - Modelo de Entidade e Relacionamento	48
3.9 - Diagrama de classes	49
3.10 – Diagrama de Sequência	50
3.11 - Prototipação	51
CONSIDERAÇÕES FINAIS	61
REFERÊNCIAS BIBLIOGRÁFICAS	63
ANEXOS	65

INTRODUÇÃO

O processo de automação para as rotinas comerciais no Brasil está ocorrendo em um ritmo cada vez mais célere, a popularização dos *smartphones* e computadores no mercado é um grande propulsor para esse cenário, de acordo com dados da Pesquisa Anual de Administração e Uso da Tecnologia da informação nas empresas, a projeção de densidade de dispositivos conectados à internet será de dois por habitante, alcançando 416 milhões de aparelhos em 2018, as empresas cada vez mais atentas a essas tendências, estão buscando absorver o melhor dessa tecnologia e transformando em ganhos ao seu negócio.

Se tratando do ramo de mercado B2B (*Business to business*), a figura do representante que simplesmente vende os produtos está cada vez mais extinta, as empresas estão enxergando que eles podem absorver com facilidade rotinas que antes eram desempenhadas internamente no âmbito empresarial. Com o auxílio da tecnologia as empresas cada vez mais transpõem essas rotinas comerciais para a palma das mãos dos representantes, ganhando tempo na execução de tarefas, objetivando o negócio, assim focando no que realmente é necessário.

Os sistemas de CRM (*Customer Relationship Management*) são as ferramentas que estão garantindo esse sucesso, cada vez mais decisivos nas elaborações de estratégias de crescimento de negócio para as empresas, a utilização dessas tecnologias está deixando de tornar-se uma tendência e cada vez mais se tornando um pré-requisito na busca de resultados, obviamente não são os únicos responsáveis por esse feito, pois sua base está interligada aos softwares de gestão empresariais e dos sistemas forças de vendas.

Assim um bom sistema de vendas é aquele que consegue unir o melhor do CRM aos sistemas forças de vendas o que transitoriamente tende a proporcionar para o vendedor externo uma imensa praticidade. Pense na seguinte situação, ao invés de ir até o cliente carregando catálogos, blocos de notas, e mais um monte de documentos, o vendedor começa a se apresentar simplesmente com um *smartphone* que, com o sistema de CRM de força de vendas adequado, será o bastante para suprir todas essas necessidades ora levantadas.

São todas as funcionalidades apresentadas diretamente no dispositivo móvel, com isso o vendedor tem maior fluidez para demonstrar produtos, consultar seus preços, montar orçamentos, avaliar descontos e ainda contatar a empresa caso necessário. O que torna o atendimento muito mais ágil.

Outrossim, basta possuir uma conexão com a internet, o vendedor já poderá encaminhar o pedido diretamente para o ERP (*Enterprise Resource Planning*) da empresa no momento da venda por meio da comunicação entre web service (*Serviços Web*), o que irá proporcionar à empresa uma maior agilidade também para a entrega, pois poderá se programar com maior tempo.

Um vendedor externo com um bom sistema CRM de força de vendas não vai ter o problema de não saber o preço do produto, ou de se complicar com as possibilidades de prazo da negociação, pois as informações relativas ao cliente estarão diretamente em suas mãos, assim o atendimento ganha não só em agilidade, mas também em satisfação do cliente.

Com um sistema CRM de força de vendas, a empresa pode gerenciar todas as transações feitas pelos seus representantes. Com essas transações integradas ao sistema de gestão na empresa, é possível comparar informações de quem está vendendo mais, qual cliente é um bom comprador, que tipo de produto é mais interessante para cada região de vendas, entre outros dados essenciais para uma boa gestão. Isso facilita ao gestor o corte de gastos desnecessários, o investimento nos mercados certos, um trabalho mais eficiente e, finalmente, o aumento da lucratividade do negócio, e ainda é possível transpor essas informações ao vendedor externo, o que o auxilia no ato da venda.

A importância dos sistemas CRM é cada vez mais vital na simplificação do processo de vendas, e a solução mobile para forças de vendas funciona como alicerce na busca por resultados, garantindo que as metas serão cumpridas e com máxima utilização do tempo para todos envolvidos nesse processo, é uma relação mútua em que todos ganham principalmente o cliente com a máxima eficácia em seu atendimento.

Diante disso nos próximos capítulos aborda-se todo esse enredo aplicando aos aspectos empresariais, tecnologias por trás, e como as soluções de CRM estão trazendo inovações nesse cenário, proporcionando cada vez mais um maior engajamento da relação entre as partes envolvidas no processo das vendas.

No capítulo 1 apresenta-se a definição do CRM, conceitos gerais, características, objetivos para empresa e também será definido os sistemas forças de vendas e como as tecnológicas de CRM vem inovando o contexto deles, no capítulo 2 aborda-se os recursos tecnológicos integrados, com definição da linguagem de programação Android. Também se aborda a definição da linguagem de modelagem UML que fora a linguagem de modelagem escolhida no contexto do trabalho, juntamente com os diagramas de caso de uso e de classe e sequência que a compõem.

Por fim no capítulo 5 trata-se o estudo de caso referente o assunto abordado, trazendo toda sua contextualidade e definição do problema que veio a proporcionar esse enredo possível.

CAPÍTULO 1 - CUSTOMER RELATIONSHIP MANAGEMENT (CRM)

CRM consiste no conjunto de ferramentas administrativas embasadas no marketing de relacionamento, este termo justamente é utilizado por sua proposta de agrupar ferramentas de comunicação e contato com o cliente, é uma arte que proporciona a empresa um melhor conhecimento de seus clientes valorizando-os essa teoria pode ser comprovada pela afirmação de Souki (2006, p. 10): “Fica cinco vezes mais barato manter os clientes que já conquistamos do que sair à procura de novos. ”

Para Brown (2001), “O CRM permite que uma empresa aborde todos os tipos de clientes que ela atendeu ou atende em momentos diferentes do seu ciclo de vida, que escolha o programa de marketing que melhor se enquadre com o ponto de vista de um cliente em relação à empresa e a sua vontade de comprar os serviços e produtos oferecidos”.

Em tese trata-se de uma estratégia de negócio, que softwares com foco no cliente utilizam, tudo se baseia em um conjunto de processos focados na gestão dos clientes, conforme necessidade empresarial, tendo como principal objetivo a fidelização de seus clientes e a plena satisfação dos mesmos, é também um sistema integrado de gestão com foco no cliente, que reúne vários processos e tarefas de uma forma organizada e integrada.

O CRM surge no universo empresarial como um conjunto de ferramentas com o objetivo de estreitar o relacionamento com os clientes, sendo muito mais do que um simples software, mas a base para a sustentação de estratégias de marketing de relacionamento desenvolvidas pelas empresas.

Assim sendo, sistemas CRM são desenvolvidos para a gestão de clientes de uma empresa e se subdividem nos seguintes tipos:

a) Operacional:

Refere-se a integração da tecnologia aos processos de gestão e operação, para orientar e auxiliar o relacionamento entre cliente e empresa e aplica-se a tecnologia para a melhoria da eficiência desse relacionamento, através de sistemas como automatização da força de vendas. Outro ponto característico é a utilização das centrais de atendimento para a coleta de dados, centros de atendimento a clientes (call centers), sites de comércio eletrônico e sistemas de pedido automatizados.

b) Colaborativo:

Refere-se a aplicação da tecnologia a automação de todos os pontos de contato da empresa com seus clientes, a principal função do CRM Colaborativo é permitir a automação e a integração entre todos os pontos de contato da empresa com o cliente, como telefone, chat, e-mail, autoatendimento e redes sociais. O objetivo é disseminar informações obtidas tanto pelo CRM Analítico quanto pelo CRM Operacional. Essa subdivisão é bastante utilizada nas companhias quando é preciso ter uma visão mais ampla de mercado, integrando uma quantidade maior de informações no banco de dados da empresa.

c) Analítico:

Refere-se a identificação e aferição de diferentes tipos de clientes da empresa, a partir disso é possível traçar estratégias para atendê-los de forma a satisfazer suas necessidades específicas. A função da CRM analítico é analisar os dados do cliente gerados através de ferramentas operacionais para fins de gerenciamento de desempenho empresarial, buscando prospectar, segmentar, identificar clientes, produtos, etc. É a fonte de toda a inteligência do processo. Sem essa estratégia todos os outros esforços são nulos. O CRM analítico é a parte do ambiente de CRM que fornece a análise do cliente por dados de seu comportamento para tomar decisões empresariais. Com isso pode-se conseguir uma redução significativa dos custos de atendimento aos clientes de menor potencial e também um aumento dos custos no atendimento dos clientes de maior potencial.

d) Social:

Refere-se ao próprio termo, que é a interação com o cliente por intermédio das diversas mídias existentes. Sua relevância no contexto deste trabalho está relacionada a infusão das tendências de marketing de relacionamento ao processo de vendas, possibilitando auxílio na tomada de decisão por parte do vendedor externo, pois a forma que enxergará o cliente será diferente partindo do princípio que houve recepção de informações valiosas expostas pelos mesmos.

Algumas empresas utilizam o CRM diretamente com a ferramenta de BI (Inteligência de negócios) para ter uma análise melhor das informações. Ao adotar a estratégia de CRM a empresa interage com todos os clientes, conhece os seus hábitos, criando eventos, promoções e campanhas direcionadas, e torna o atendimento mais personalizado e consegue a fidelização do cliente.

As organizações devem saber gerenciar ao máximo as diferenças existentes entre a variedade de clientes que sua empresa trabalha, para que possam entender cada um deles, é um processo que aproveita o conhecimento obtido do cliente através das diferentes interações para melhorar a satisfação geral do cliente, agradando da melhor forma possível, como se fosse o único, ou o último cliente da face da terra.

Para isso, serão apresentados os principais conceitos do CRM e marketing de relacionamento dentro do universo empresarial, bem como, demonstrar como a utilização de ferramentas de CRM pode ajudar na fidelização de clientes e avaliar o papel do profissional de TI no desenvolvimento de ferramentas de CRM.

Assim, o CRM possibilita a empresa o armazenamento de informações dos clientes de sua base, sendo que essas informações são transpassadas ao vendedor externo por meio dos sistemas forças de vendas, o que possibilita ao vendedor confrontar esses dados com o perfil de compra de seus clientes. A empresa pode até segmentar seus clientes para melhores direcionamentos de seus produtos, o que torna a solução de CRM uma das tecnologias corporativas mais importantes e inovadoras disponíveis para empresas, pois reúne essas informações para gerenciar os clientes em um único local, isso mostra que os gerentes de organizações estão cada vez mais focados nas necessidades de seus clientes e estão buscando cada vez mais transpassa-las a todo nicho empresarial.

“Se uma organização estiver procurando afinar todos os pontos de contato com a marca, integrando pessoas, processos e tecnologia do ponto de vista do cliente, resultando em valor de longo prazo para a marca, para a lealdade do cliente e rentabilidade, então pode-se ter certeza de que ela está entendendo o que significa CRM.” (LOBO, 2002).

Se tratando de estratégia, Brown, (2001) afirma:

“O CRM permite que uma empresa aborde todos os tipos de clientes que ela atendeu ou atende em momentos diferentes de seu ciclo de vida, que escolha o programa de marketing que melhor se enquadre com o ponto de vista de um cliente em relação à empresa e sua vontade de comprar os serviços e produtos oferecidos.”

Em suma esse processo de valorização dos clientes possibilita a troca de conhecimento entre empresa e cliente, criando vínculos e fidelizando os mesmos, assim cabe a cada organização a manipulação do sistema conforme seus anseios para que as empresas possam conseguir uma vantagem no mercado cada vez mais competitivo:

“A real vantagem competitiva de uma empresa consiste em ter informação sobre seus clientes que a concorrência não tem e ser capaz de transformar rapidamente esse conhecimento em ação. Ganha quem consegue estabelecer relacionamentos dominantes em seu mercado”. (PEPPERS, 2003).

1.1 - Softwares de apoio ao CRM

Greenberg (2001) defende que para a adoção de estratégias de CRM, as organizações dependem da tecnologia de informação. Assim as tecnologias de CRM correspondem a um dos componentes do CRM, seja desde banco de dados a outras soluções. A contextualização do banco de dados entra nesse enredo como a parte que provem os dados para análise dos softwares.

As tecnologias de informação, *softwares*, de CRM podem ser desenvolvidos de modo a proporcionar armazenamento de dados, apoio a automação de vendas, análise de dados entre outros, o que proporciona a empresa um ganho em transformação de dados em informações.

Segundo Mañas (2002, p. 2) um sistema de informação permite mostrar em que ponto tomar decisões, para permitir maior agilidade, flexibilidade e competência diante da concorrência. O sistema de informação transforma dados em informação. Este sistema de informação é o conjunto de hardware e software na captação de dados. E planejamento e gestão organizacional contribui para a tomada de decisão.

Os sistemas de informação surgem como suporte ao processo de tomada de decisão, pois com o uso sistema de informação no apoio ao CRM, unifica-se no sistema de informação um grande volume de informações referente aos clientes da empresa, e assim proporcionado em vendas, maior rentabilidade, produtividade para enfim atingir a fidelização do cliente. Os Sistemas de informações são de grande relevância ao CRM na gestão e organização das empresas, e não se pode esquecer que CRM é também uma filosofia de trabalho.

Muitas organizações operam com vários comerciais externos e têm a necessidade de integrar as suas ofertas numa só aplicação, que é apoiada pela sua arquitetura de tecnologia de informação. O uso de aplicativos integrados de CRM para obter, processar e distribuir as informações do cliente através de múltiplos canais deverá afetar positivamente a capacidade de uma organização para gerir os contatos com os clientes e, assim, aumentar a satisfação do cliente (Mithas et al., 2005).

1.2 - Ciclo do CRM

CRM é uma estratégia de negócio voltada ao entendimento e à antecipação das necessidades dos clientes atuais e potenciais de uma empresa. Do ponto de vista tecnológico, CRM envolve capturar os dados do cliente ao longo de toda a empresa, consolidar todos os dados capturados interna e externamente em um banco de dados central, analisar os dados consolidados, distribuir os resultados dessa análise aos vários pontos de contato com o cliente e usar essa informação ao interagir com o cliente através de qualquer ponto de contato com a empresa. (PEPPERS; ROGERS GROUP, 2004, p.59).

Seu ciclo de subdivide da seguinte forma:



Figura 1. Ciclo do CRM, Fonte: Newell, 2006, p. 56

Marketing:

Aplicativos que otimizam o processo de marketing em uma empresa, com o objetivo de alocar recursos para mercados alvos com os maiores valores potenciais, ferramentas de *data mining* e de *dababase marketing*.

Vendas:

Conjunto de softwares que administra e otimiza o ciclo de vendas da empresa, aumentando sua produtividade, principalmente, por meio da otimização da realização dos contratos.

Pedidos:

Aplicativos que otimizam o processo de captação de vendas, como sistemas forças de vendas.

Suporte:

Aplicativos desenvolvidos à parte de outros pacotes CRM, como, por exemplo *help desks* automatizados e sistemas de *call center*.

Principais Características de um CRM:

- ✓ Reconhecer e cuidar do cliente em tempo real;
- ✓ Aumentar os resultados em vendas;
- ✓ Maximizar os canais de comunicação com os clientes;
- ✓ Racionalizar os processos, atividades e ações comerciais;
- ✓ Sistematizar e organizar os processos de vendas;
- ✓ Reduzir custos operacionais/comerciais;
- ✓ Gestão de banco de dados de clientes e fornecedores;
- ✓ Monitorar a equipe de vendas;
- ✓ Transforma dados dispersos em informações úteis e centralizadas;
- ✓ Formalizar os processos de comunicação com os clientes;
- ✓ Administrar equipes de atendimento com mais competência e organização;
- ✓ Atender ao público com engenhosidade e habilidade;
- ✓ Automação de marketing, força de vendas e serviços;
- ✓ Desenvolvimento do funil de vendas;
- ✓ Controlar oportunidades;
- ✓ Criar relatórios, formulários e métricas voltadas a estatística e dados analíticos;
- ✓ Estruturação da equipe de vendas;
- ✓ Desenvolvimento de ambiente colaborativo.

1.3 - Sistemas Forças de Vendas

Sistemas forças de vendas são definidos como softwares de apoio aos vendedores externos da empresa sua importância é altamente considerável no âmbito empresarial e sua origem está relacionada a difusão das soluções móveis e da internet no cenário das empresas, para os vendedores externos os sistemas forças de vendas são sua ligação com a empresa e para o cliente em alguns casos a empresa é o vendedor, Kotler (2000, p 638) dá mais ênfase na afirmação “os vendedores servem de elo pessoal da empresa com os clientes e o profissional de vendas é a empresa para muitos clientes dela”, por isso sua necessidade é crucial ao negócio, pois está relacionado a toda a cadeia do processo de vendas.

No ciclo de vendas relaciona-se como a ferramenta que possibilita ao vendedor da empresa a comunicação com a empresa, podendo também ser definido como softwares de apoio a operações comerciais, eles visam facilitar as rotinas de vendedores externos e de seus gestores, pensando em um mercado cada vez mais globalizado os sistemas forças de vendas proporcionam inúmeros benefícios as empresas, como: economia de tempo, melhor atendimento ao cliente, aumento da receita, gestão de oportunidades, gerenciamento de equipe de vendas e tomada de decisões rápidas de forma ágil e segura com o auxílio da portabilidade.

Uma equipe de vendedores precisa estar preparada para o processo de venda, conhecendo exatamente os seus clientes, produtos, rentabilidade e outras informações inerentes a operação que será realizada, afinal de contas o objetivo principal da arte de vender é aumentar a lucratividade da empresa consequentemente o vendedor aumenta sua comissão.

Esse processo de automação da força de vendas oferece a equipe externa (vendedores) uma ferramenta possível para que eles possam tomar as decisões certas pertinentes aos seus objetivos a fim de alcançar as metas traçadas pela organização.

Ao nível da operação, os sistemas forças de vendas são as extremidades do *iceberg* do processo de venda, como são em sua maioria transmissores/receptores de informações necessitam dos softwares de ERP (*Enterprise Resource Planning*) para o enlace completo, pois as informações consumidas e enviadas serão realizadas por meio deles, outrossim o fato primordial para que haja a comunicação está relacionada figura dos *web services* que possibilitam essa ponte.

Assim conclui-se que sistemas forças de vendas são softwares para auxílio nas rotinas diárias dos vendedores, apoiando seu esforço proporcionando eficiência pois acaba livrando-o de papeladas, e ainda gerando ganhos a empresa, pois torna a venda mais objetiva

e ajuda no processo de fidelização dos clientes, isto desde que sejam aplicados juntamente com o contexto que abordaremos a seguir.

1.4 - CRM Operacional

Este é um segmento que se assemelha ao *Enterprise Resource Planning* (ERP), é desenvolvido com o foco principal na automação e na melhoria de processos de negócios que são baseados em clientes reais, a importância do CRM operacional está relacionada diretamente com o Marketing da empresa, operações de venda e serviços. Os CRM operacionais tratam dos aspectos operacionais da relação com os clientes, com ele a empresa cuida da automação de todos os seus processos de venda, marketing e atendimento ao cliente. Assim possui foco principal na automação e na melhoria de processos de negócios que são baseados em clientes.

O CRM operacional prevê a integração de todos os sistemas para proporcionar o melhor conjunto de informações para o atendimento do cliente, assim é possível inserir nesse contexto o CRM analítico, pois ele é o componente do CRM que identifica e acompanha diferentes tipos de clientes dentro da carteira da empresa, com essas informações intercaladas e em mãos é possível elaborar estratégia para alcançar as diferentes necessidades dos clientes proporcionando um ganho de satisfação para os mesmos.

O CRM operacional auxilia o gerenciamento de pedidos, produção e feedback direcionados a empresa, neste contexto é possível aplicá-lo na automação de vendas, garantindo a empresa um ganho de tempo e melhor utilização de informação de seus clientes, a partir disso também é possível notar sua relação com os sistemas forças de vendas pois os mesmos são partes fundamentais na captação dessa base de dados, pois é por meio deles que as informações do relacionamento entre cliente-empresa são criadas, toda captação de pedido é transformada em uma informação que poderá ser utilizada nesse processo, informações de itens vendidos podem ser transformadas em sugestões para próximas compras e até aplicadas a clientes com as mesmas características, e assim sucessivamente com outros dados inerentes ao processo de venda.

Diante disso pode-se definir o CRM operacional como a aplicação da tecnologia de informação para melhoria da eficiência do relacionamento entre cliente-empresa, pois o CRM operacional são também ferramentas e processos adotados para as atividades de atendimento ao cliente, registro de suas reclamações e encaminhamento destas

as áreas responsáveis da empresa. O CRM operacional também pode ser compreendido como uma base de dados com informações dos clientes da empresa, que são utilizadas de maneira a proporcionar um amplo conhecimento sobre o cliente e seu relacionamento com a empresa.

CAPÍTULO 2 - RECURSOS TECNOLÓGICOS

Baseado no contexto proposto pelo trabalho, a ligação das tecnologias dos sistemas forças de vendas e do CRM tendem a ter ligação sobre os aspectos que serão abordados a seguir, neste cenário as ênfases que serão dadas referem-se a aspectos necessários para sua implantação, desde sistema operacionais para execução da aplicação até sua contextualização desde seu surgimento e definições sobre o mecanismo de banco de dados, que é um dos pilares para que haja operabilidade desses sistemas interligados.

2.1 - Sistema operacional Android

Em outubro de 2003, Andy Rubin, Rich Miner, Nick Sears e Chris White fundaram a Android Inc. uma empresa que se concentrava no "desenvolvimento de software para telefones celulares". A empresa passou quase dois anos trabalhando "na sombra", até que o Google começou a recrutar algumas startups com a intenção de replicar o sucesso da Web no futuro das telecomunicações sem fio.

No dia 5 de novembro de 2007 foi fundado o OHA (Open Handset Alliance), uma aliança comercial dos 35 componentes iniciais liderados pela Google, que incluiu fabricantes de celulares, operadores de telecomunicações, fabricantes de chips e desenvolvedores de software. Assim logo fora anunciado o que conhecemos hoje como o Android, uma plataforma de código aberto para dispositivos móveis baseado em Linux, sem configurações fixas de hardware e software.

Foi a Google que publicou a maior parte do código-fonte do sistema operacional, graças ao software Apache, que é uma base que suporta projetos de software de código aberto.

Segundo GOOGLE INC (2012):

“O Android foi idealizado desde o início para ser um sistema com código fonte open source, facilitando a adequação a diversos dispositivos diferentes e personalização do conteúdo e, devido a essa facilidade, é comum encontrarmos aparelhos que possuem personalizações da operadora de telefonia ou do fabricante como forma de concorrência entre os mesmos. Pelo fato de ser baseado no Linux, ele também possui um repositório de aplicativos, onde todos os aplicativos públicos a serem instalados nos sistema podem ser visualizados e adquiridos.”

Como características principais do Android [LEE, 2011] afirma:

“A principal vantagem do Android é que ele oferece uma abordagem unificada para desenvolvimento de aplicativos. ”internamente suporta as seguintes Funcionalidades:

- ✓ Armazenamento –usa o SQLite uma base de dados relacional leve, para armazenamento de dados;
- ✓ Conectividade – suporta WIFI, Bluetooth, GSM e outras;
- ✓ Navegador Web – baseado no WebKit de código aberto, juntamente com JavaScript V8 do chrome;
- ✓ Suporte a meios – inclui MP3, MP4, MIDI, JPEG, ETC;
- ✓ Suporte a hardware – sensor de acelerômetro, câmera bússola digital, sensor de proximidade;
- ✓ Multitoques – suporte a telas multitoques;
- ✓ Multitarefa – suporte aplicativo multitarefas;
- ✓ Suporte flash – Android 2.3 suporta o flash 10.1;
- ✓ Tethering – suporte ao compartimento de conexões de Internet como hotspot com e sem fios. ”

O Android está atualmente na versão 8.0 Oreo, de acordo com GOOGLE INC (2012) de tempos em tempos são disponibilizadas novas versões do Android, normalmente contendo novos recursos, melhoria de desempenho e correção de bugs e problemas.

A premissa para designar os nomes das versões vem a partir da definição de um nome de doce, sempre em ordem alfabética, atualmente já fora lançada as versões:

- ✓ Android 1.5: Cupcake (30 de abril de 2009);
- ✓ Android 1.6: Donut (15 de setembro de 2009);
- ✓ Android 2.1: Eclair (26 de outubro de 2009);
- ✓ Android 2.2: Froyo (20 de maio de 2010);
- ✓ Android 2.3: Gingerbread (06 de dezembro de 2010);
- ✓ Android 3.0: Honeycomb (22 de fevereiro de 2011);
- ✓ Android 4.0: Ice Cream Sandwich (19 de outubro de 2011);
- ✓ Android 4.1: Jelly Bean (9 de julho de 2012);
- ✓ Android 4.4.2: KitKat (31 de outubro de 2013);
- ✓ Android 5.0: Lollipop (12 de novembro de 2014);
- ✓ Android 6.0: Marshmallow (05 de outubro de 2015);
- ✓ Android 7.0: Nougat (22 de agosto de 2016);
- ✓ Android 8.0: Oreo (21 de agosto de 2017);

2.2 - Aplicações Android

2.2.1 - Android SDK

Segundo LEE (2011, p.9) O Android SDK (Kit de desenvolvimento de software) fornece as bibliotecas da API (*Application Programming Interface*) e ferramentas de desenvolvimento necessárias para construir, testar e depurar aplicativos para Android.

Assim o Android SDK permite aos desenvolvedores criarem aplicativos para a plataforma Android de forma nativa, com isso os aplicativos são escritos usando a linguagem de programação Java e executados em máquinas virtuais personalizadas e projetadas para rodar dentro dos dispositivos Android.

Sempre que há lançamento de uma nova versão do Android, a Google também lança uma versão do SDK correspondente, tem se como base de programação utilizar-se do ambiente de desenvolvimento integrado (IDE), em particular a IDE Android Studio, partindo do princípio que é a ferramenta oficial da Google para desenvolvimento para a plataforma.

A IDE Android Studio fornece uma interface gráfica que possibilita aos desenvolvedores executar tarefas de desenvolvimento em ritmo mais célere.

Em suma o Android SDK é a ferramenta que fornece um ambiente de desenvolvimento rápido e confiável para aqueles que querem desenvolver aplicativos Android.

2.2.2 - Estrutura do projeto Android

A IDE Android Studio como estrutura do projeto possui as seguintes propriedades de componentes:

Gradle Scripts – Arquivos que serão utilizados para geração do nosso Aplicativo.

App – Pasta do nosso Aplicativo: MANIFEST (Configurações do seu aplicativo).

Java – Pasta com dois Pacotes: *Application* (onde iremos criar nossos arquivos java); e *Application Test* (onde faremos testes da nossa Aplicação).

Res – de Resources: drawable (imagens do nosso Aplicativo); layout (onde ficarão as nossas telas); menu (Tela de MENU); mipmap (ícones para dispositivos variados); values (dentro dessa pasta tem a pasta *dimens.xml* que guarda informações de DIMENSÕES para telas menores e maiores (w820dp) de dispositivos.; Temos a pasta *strings* onde conseguimos centralizar a criação de textos para que possam ser utilizados em mais de uma tela do aplicativo; *styles*, permite estilizar, customizar a interface dos aplicativos.

Visualizações:

Android – Visualização Inicial;

Package; Todas as pastas dentro do nosso app. Como ficará organizado nosso Projeto utilizando as Pastas;

Project – Exibe a pasta App e outras pastas (build, mostra as versões do nosso App; lib utilizado para guardar bibliotecas.

2.2.3 - APK

Android Package Manger (APK) refere-se a uma coleção de arquivos diferentes (como código fonte, imagens e ícones, áudio, vídeo, etc.) compilados e empacotados em um arquivo para a finalidade de distribuição e instalação.

Os aplicativos móveis foram inicialmente desenvolvidos como um substituto para os softwares de escala completa a serem executados nos dispositivos e tablets móveis, assim um APK é uma compilação de um código em java que será interpretado pelo SO Android.

2.2.4 - Mecanismos para publicação de APK

Segundo GOOGLE INC:

“A publicação é o processo geral que disponibiliza aplicativos Android para os usuários. Quando se publica um aplicativo Android, executam-se duas tarefas principais:

Preparação do aplicativo para lançamento.

Durante a etapa de preparação, compila-se uma versão de lançamento do aplicativo que pode ser baixada e instalada pelos usuários em dispositivos com o Android.

Liberação do aplicativo para os usuários.

Durante a etapa de lançamento, ocorre a divulgação, venda e distribuição da versão de lançamento do aplicativo para os usuários. ”

Assim a partir dessa preparação do aplicativo para lançamento temos o processo de publicação do mesmo, que é realizado a partir de sua própria loja denominada Google Play como padrão, mas também é possível lançar o aplicativo em seu site ou envia-lo diretamente ao usuário.

2.3 - Análise e projeto do sistema

2.3.1 - UML

Definida como Linguagem Unificada de Modelagem, a UML caracteriza-se por ser uma linguagem gráfica para elaboração da estrutura de projetos de software, seus objetos são: visualização, especificação e construção, na etapa de visualização são elaborados desenhos a fim de apresentar o sistema, na etapa de especificação é escrito em detalhes partes do sistema no intuito de detalhá-lo ao máximo possível, por fim na etapa de construção é mostrado o modo ao qual o sistema foi construído.

Deve-se atentar que sua abrangência inclui diversos níveis de sistemas, porém não é caracterizada como uma metodologia de desenvolvimento, ao âmbito de desenvolvimento de sistemas ela permitirá aos desenvolvedores a visualização dos produtos de seus trabalhos em diagramas personalizados.

A UML foca nos elementos conceituais e físicos que representa o sistema, mas deve-se notar que ela não é apenas uma linguagem visual. Ao nível de projeto proporciona vantagens ao direcionar o desenvolvimento de forma mais célere e efetiva, e ainda traz um melhor entendimento do sistema que está sendo construído.

Juntamente com essas definições temos também a documentação de artefatos de sistemas complexos de software, que é onde explica-se cada passo do sistema a nível de execução. Baseado nessas proposições afirma-se que a UML é uma linguagem muito expressiva, abrangendo todas as visões necessárias do desenvolvimento e implementação desses sistemas, uma linguagem que vai mostrar tudo de forma bem detalhada, de passo a passo.

Seguindo essa linha de raciocínio, o autor Booch (2005) destaca que a UML poderá ser empregada para visualização, especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software, isso mostra que vão ser gráficos mostrando de modo exemplificados um sistema de software e seu comportamento.

O autor Eduardo Bezerra (2007, pg.15) destaca: “A UML é uma linguagem visual para modelar sistemas orientadas a objetos”. Isso fortalece a ideia de que a UML é um modelo de análise, projeto e programação de sistemas de software, baseado na composição entre diversas unidades de software chamadas de objetos.

Diante das afirmativas e agregando um maior embasamento Gilleanes T.A Guedes (2011, p.19) afirma: “É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação”. Isso tudo enriquece ainda mais o que já fora relatado, a UML não é uma linguagem de programação e sim uma linguagem de modelagem, uma notação, cujo objetivo é auxiliar e também ajudar a definir as características do sistema.

2.3.2 - Diagrama de Caso de Uso

Diagramas de Casos de Uso mostram as funcionalidades do sistema do ponto de vista do usuário, são elaborados com o intuito de visualizar, especificar e documentar o comportamento de um elemento. “Um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema” (Eduardo Bezerra, 2007), A Figura 2 ilustra um diagrama de caso de uso de um sistema de prestadores de serviço:

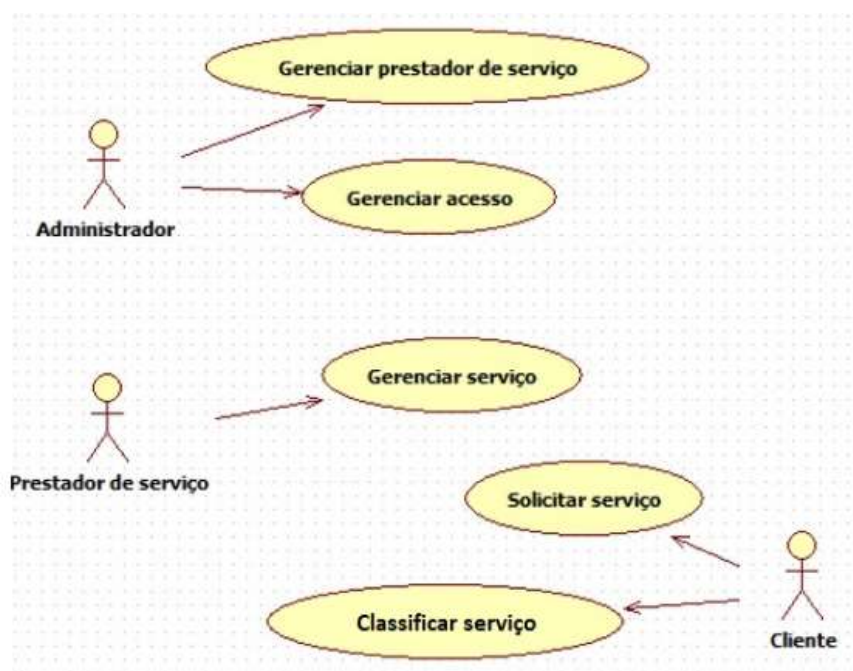


Figura 2. Diagrama de caso de uso - sistema gestão prestadores de serviço

Gilleanes T.A Guedes (2011, p.30) complementa: “Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar”.

2.3.3 - Diagrama de Classe

O diagrama de classe é o diagrama que contém o nome da classe, seus atributos e métodos, os atributos são suas características e os seus métodos são suas ações que podem se relacionar e trocar informações.

O autor Gilleanes T. A. Guedes (2011, p.31) enfatiza: “Como o próprio nome diz, define a estrutura das classes utilizadas pelo sistema, determinado os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si.”; ou seja, contém o nome da classe, seus atributos que vão ser sua composição e seus métodos que são suas ações de sua classe e vão poder se relacionar e trocar informações entre si.

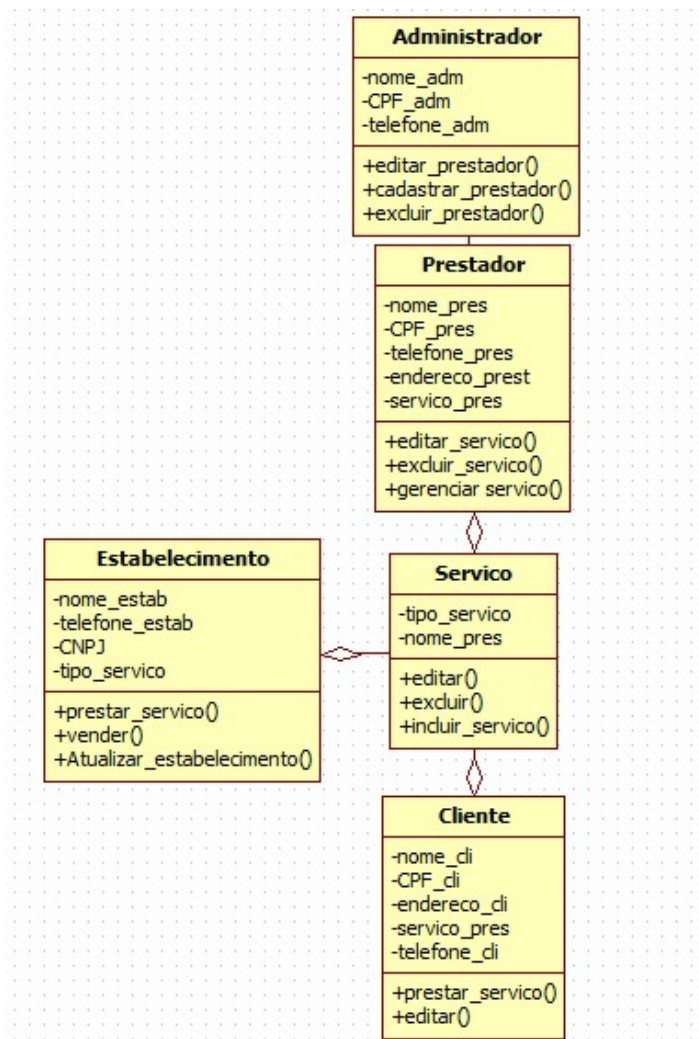


Figura 3. Exemplo Diagrama de Classes Sistema Agendamento de Prestador de Serviços

2.3.4 - Diagrama de Sequência

O diagrama de sequência é um diagrama de interações entre objetos na ordem temporal, ou seja, vai mostrar o tempo que uma ação vai acontecer no sistema, seguindo a linha de raciocínio de Eduardo Bezerra (2007, p.193): “O objetivo do diagrama de sequência é apresentar as interações entre objetos na ordem temporal em que elas acontecem. ”, isso mostra as ações do sistema na ordem em que elas vão acontecer e o tempo que elas irão durar.

Gilleanes T.A Guedes (2008, p.33) relata: “O diagrama é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. ” Isso significa que um diagrama de sequência foca na sequência de ações e mostra também o tempo de cada ação.

Seus componentes são:

Linha de Vida: As linhas de vida compõem a dimensão vertical. Uma linha de vida é composta de duas partes, a cabeça e a cauda. A cabeça é representada por um retângulo com dois compartimentos, no compartimento superior a identificação do objeto é exibida e no compartimento inferior (cuja utilização é opcional), aparecem valores para os atributos definidos na classe do objeto. A cauda corresponde a uma linha vertical tracejada.

Mensagens: É a interação entre objetos e representada por mensagens, para outros objetos e para o mesmo.

Ator: São entidades externas que interagem com o sistema e que solicitam serviços. Normalmente, o ator primário é o responsável por enviar a mensagem inicial que inicia a interação entre os objetos e o tempo.

Para montá-los seguimos a linha:

Apresenta-se a colaboração dinâmica entre vários objetos de um sistema;

Através deste diagrama é possível perceber a sequência de mensagens enviadas entre objetos;

Mostra o que ocorre em pontos específicos da execução do sistema;

Apresenta as interações entre atores e sistema.

Objeto: Representam as instâncias das classes representadas no processo.

Por fim para conclusão seguimos o raciocínio que o autor Booch contextualiza: “Um diagrama de sequência dá ênfase à ordenação temporal das mensagens. ” Isso significa que vai mostrar a ordem e o tempo das mensagens do sistema.

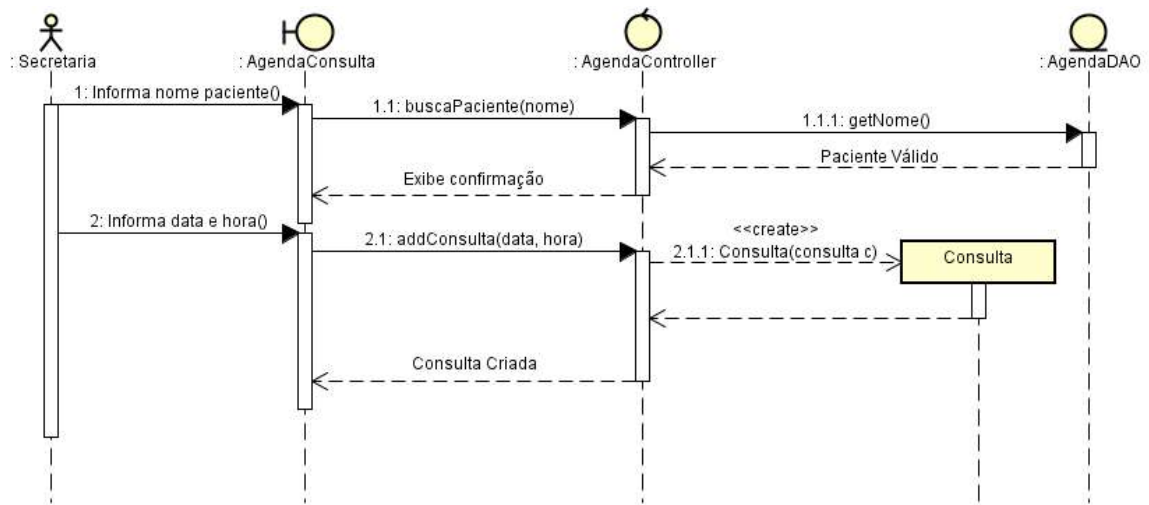


Figura 4. Exemplo Diagrama de Sequência - Sistemas de agendamento de consultas

2.4 - Banco de dados

Definido como coleções dados interligados entre si e organizados para fornecer informações, normalmente os bancos de dados estão estruturados em arquivos, tabelas e registros, e são operados por sistemas gerenciados de banco de dados (SGBD), possuem como principal aplicação o controle de operações empresariais.

Para ELMASRI e NAVATHE (2011, p. 3):

“Um banco de dados representa algum aspecto do mundo real, às vezes chamado de minimundo ou de universo de discurso (UoD – Universe of Discourse). As mudanças no minimundo são refletidas no Banco de Dados.

Um banco de dados é uma coleção logicamente coerente de dados com algum significado inerente. Uma variedade aleatória de dados não pode ser corretamente chamada de banco de dados.”

“Um banco de dados é projetado, construído e populado com dados para uma finalidade específica. Ele possui um grupo definido de usuários e algumas aplicações previamente concebidas nas quais esses usuários estão interessados.”

De acordo com Elmasri (2011, p.3), um banco de dados normalmente é caracterizado por algumas propriedades implícitas:

Só é considerado banco de dados, aquele que representa algum aspecto do mundo real, às vezes chamado de minimundo ou de universo de discurso (*UoD - Universe of Discourse*). As mudanças no minimundo são refletidas no banco de dados.

Apenas coleções logicamente coerentes de dados com algum significado inerente são consideradas um banco de dados. Uma variedade aleatória de dados não pode ser corretamente chamada de banco de dados.

Um banco de dados é projetado, construído e populado com dados para uma finalidade específica. Ele possui um grupo definido de usuários e algumas aplicações previamente desenvolvidas nas quais esses usuários estão interessados.

Portanto, entende-se que um banco de dados é composto por dados que podem ser provenientes de uma fonte qualquer, também possuindo algum grau de interação com acontecimentos no mundo real e tendo um público bastante interessado em seu conteúdo.

Os usuários finais de um banco de dados podem manipulá-lo, realizando alterações em seus dados, fazendo assim, com que as informações mudem. Desta forma, para que o banco de dados seja preciso e confiável o tempo todo, ele precisa ser um reflexo fiel do minimundo que representa. Portanto, as alterações feitas pelos usuários finais precisam ser atualizadas no banco de dados o quanto antes.

2.4.1 - Bancos de dados relacionais

Banco de dados relacionais são definidos como conjunto de dados que armazenam dados como um conjunto de tabelas, em tese cada uma com informações diferentes. Assim os dados estão relacionados a fim de garantir que seja possível acessar informações de tabelas distintas simultaneamente, diante disso define-se esse modelo no conceito de “relação”.

Basicamente refere-se a tabelas conectadas garantindo o relacionamento com a outra por chave, chaves são consistidas em primária e estrangeira sendo que:

Chave primária: é utilizada para identificação de uma tabela por meio de uma identificação única.

Chave estrangeira: é utilizada para as tabelas de referência cruzada, assim uma chave estrangeira pode representar uma chave primária em outra tabela.

No exemplo abaixo temos uma demonstração de uma estrutura de um banco de dados relacional, com as ligações entre tabelas realizadas por intermédio de chave estrangeira:

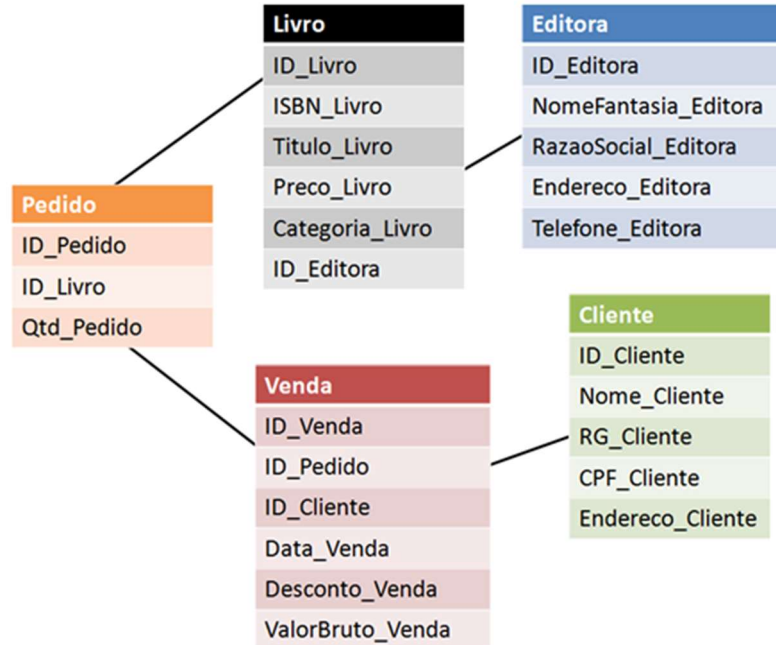


Figura 5. Modelo Banco de dados Relacional - Sistema Bibliotecário

2.4.2 - Linguagem SQL

SQL (*Structured Query Language*) é uma linguagem de programação padronizada usada para gerenciar bancos de dados relacionais e executar várias operações nos dados neles. Inicialmente criado na década de 1970, o SQL é usado regularmente por administradores de banco de dados, bem como por desenvolvedores que escrevem scripts de integração de dados e analistas de dados que procuram configurar e executar consultas analíticas.

Os usos do SQL incluem a modificação da tabela de banco de dados e estruturas de índice; adicionando, atualizando e excluindo linhas de dados; e recuperar subconjuntos de informações de um banco de dados para processamento de transações e aplicativos de análise. Consultas e outras operações SQL assumem a forma de comandos escritos como instruções - as instruções SQL comumente usadas incluem selecionar, adicionar, inserir, atualizar, excluir, criar, alterar e truncar.

Para Battisti (2006):

“Ao usar o SQL, não é necessário saber a respeito do software de banco de dados ou do hardware envolvido em uma operação. Tudo o que você precisa conhecer são os comandos/instruções SQL padrão para solicitar informações, que obrigatoriamente é o mesmo em todos os sistemas que utilizam o SQL. ”

Abaixo o comando em SQL para criação da tabela Cliente, exemplificada na Figura 5. Modelo Banco de dados Relacional - Sistema Bibliotecário:

```
CREATE TABLE `Cliente` (  
  `ID_Cliente` int(11) NOT NULL AUTO_INCREMENT,  
  `Nome_Cliente` varchar(45) NOT NULL,  
  `RG_Cliente` varchar(45) NOT NULL,  
  `CPF_Cliente` varchar(45) NOT NULL,  
  `Endereco_Cliente` varchar(45) NOT NULL,  
  PRIMARY KEY (`ID_Cliente`)  
)
```

2.4.3 - Modelo Entidade-Relacionamento (MER)

Pode ser definido como um modelo de alto-nível criado com o objetivo de representar a semântica associada aos dados do minimundo, assim o MER é amplamente utilizado na fase de projeto conceitual, digamos onde o esquema conceitual do banco de dados da aplicação é concebido.

Com conceitos intuitivos que permitem aos projetistas de banco de dados capturar conceitos associados aos dados da aplicação, sem que haja interferência da tecnologia específica de implementação do banco de dados, baseando nesse conceito ao utilizar o MER o modelo conceitual que será criado é denominado Diagrama Entidade-Relacionamento (DER), que é definido como o resultado do processo de modelagem executado pelo projetista de dados que aplica o MER.

Tem-se por base que o objeto mais elementar que o MER representa é a entidade, entidade é caracterizada como algo do mundo real que possui uma existência

independente, alguns exemplos são: objetos, pessoas. Por fim cada entidade possui propriedades particulares que são conhecidos como atributos.

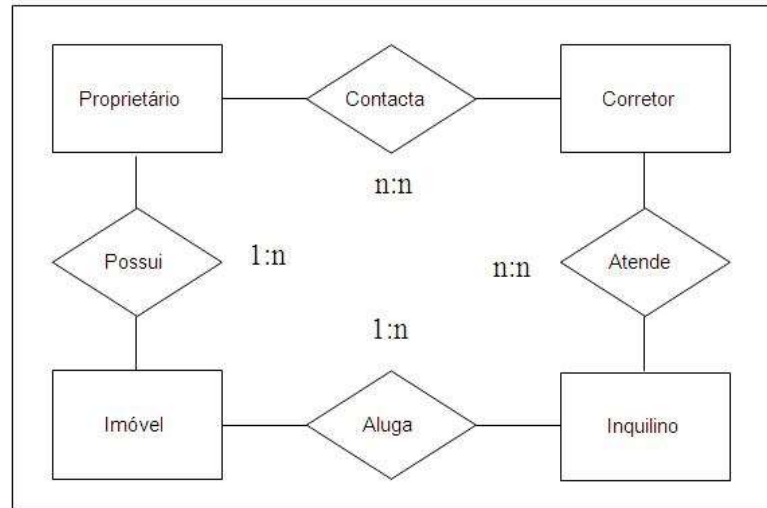


Figura 6. Exemplo MER - Sistema Imobiliário

2.4.4 - SQLite

SQLite é um sistema gerenciador de banco de dados, em menor formato, permite com que desenvolvedores possam armazenar os dados de suas aplicações em tabelas e manipular esses dados através de comandos SQL conhecidos de outros bancos de dados, como consiste em um banco de dados portátil é utilizado em aplicações mobile por sua facilidade e integração aos softwares mobiles.

Como características [LEE, 2011] define:

- ✓ “Software gratuito, multiplataforma, desenvolvido em C padrão (ANSI);
- ✓ Todo o banco de dados é guardado localmente (junto com a aplicação), em um único arquivo que possui a extensão db;
- ✓ A base de dados pode ter tamanho superior a 2 terabytes;
- ✓ Não necessita de instalação, configuração ou de administração;
- ✓ Suporta o uso de transações (COMMIT / ROLLBACK);
- ✓ Não oferece integridade referencial (chaves estrangeiras);
- ✓ Principais aplicações: programas locais, sites web, substituto de dados de dados em aulas ou demonstrações, substitui arquivos texto ou arquivos proprietários;
- ✓ Não deve ser usado nos seguintes casos: aplicações de alta concorrência e sistemas ou aplicações web de porte muito grande.”

2.5 - Linguagem de programação Java

Java consiste em uma linguagem de programação orientada a objetos, compilada para um *bytecode* e interpretada por máquina virtual (*Java Virtual Machine*), sua história de criação está definida abaixo:

Em dezembro de 1990, Patrick Naughton, engenheiro da Sun Microsystems, recrutou vários colegas, incluindo James Gosling e Mike Sheridan, para trabalhar em um novo projeto conhecido como "*The Green Project*". Com a ajuda de outros engenheiros, eles começaram a trabalhar em um pequeno escritório na Sand Hill Road, em Menlo Park, Califórnia. Eles estavam tentando desenvolver uma nova tecnologia para programar a próxima geração de dispositivos inteligentes.

No início, considerou-se C++ como uma linguagem para usar, mas tanto Gosling quanto Bill Joy achavam isso inadequado. Gosling primeiro tentou estender e modificar o C++ resultando na linguagem C++ ++ - (++ - porque C++ foi adicionado e removido), mas abandonou-o para criar uma nova linguagem que chamou de Oak. (Em razão do carvalho que ele via através da janela de seu escritório).

Oak tinha semelhanças com C, C++ e Objective C e não estava vinculado a um tipo de CPU específico. Mais tarde, o nome de Oak seria mudado para Java, devido à propriedade intelectual, porque já existia uma linguagem com o nome de Oak. Assume-se que eles definiram o nome enquanto tomavam café (Java é o nome de um tipo de café, originário da Ásia), embora outros afirmam que o nome deriva das siglas de James Gosling, Arthur Van Hoff e Andy Bechtolsheim.

Em 1992, o time já havia desenvolvido um sistema em um protótipo chamado Star7, um dispositivo semelhante a um PDA, cujo nome veio da combinação de teclas telefônicas do escritório do Projeto Verde que permitia aos usuários responderem ao telefone de qualquer lugar.

Em 23 de maio de 1995, na conferência SunWorld, John Gage da Sun Microsystems e Marc Andreessen co-fundador e vice-presidente da Netscape, anunciaram a versão alfa do Java, que na época estava funcionando apenas no Solaris e o fato que o Java deveria ser incorporado no Netscape Navigator, o navegador de Internet mais usado naquela época.

Em janeiro de 1995, a Sun formou a empresa Java Soft para desenvolver produtos baseados em Java, trabalhando com terceiros para criar aplicativos, ferramentas, sistemas de plataforma e serviços para aumentar as capacidades de idiomas.

Assim que a Netscape Communications decidiu suportar applets Java no Netscape Navigator 2.0. O Java se popularizou no âmbito tecnológico. Como parte de sua estratégia de crescimento global e para promover a nova tecnologia, a Java Soft concedeu licenças a outras empresas para que elas pudessem acessar o código fonte enquanto melhorassem seus navegadores. Também lhes permitiu criar ferramentas de desenvolvimento para a programação Java e capacitá-los a atualizar as máquinas virtuais Java (JVMs) para vários sistemas operacionais. As licenças incluíam empresas de prestígio como IBM, Microsoft, Symantec, Silicon Graphics, Oracle, Toshiba e Novell.

Atualmente você pode encontrar a tecnologia Java em todas as partes, suas características (orientação a objetos, portabilidade, recursos de rede, segurança) permitiram tamanha pulverização, além da licença gratuita.

2.6 - Web Service

Web services são tecnologias que permitem a comunicação entre aplicações de maneira independente do sistema operacional e de linguagem de programação, assim o uso dos Web Services são requisitos primordiais para a integração entre serviços distintos e padronização no retorno de cada requisição de serviços, a Figura 1 ilustra um exemplo de como é realizada a comunicação a partir dos mesmos:

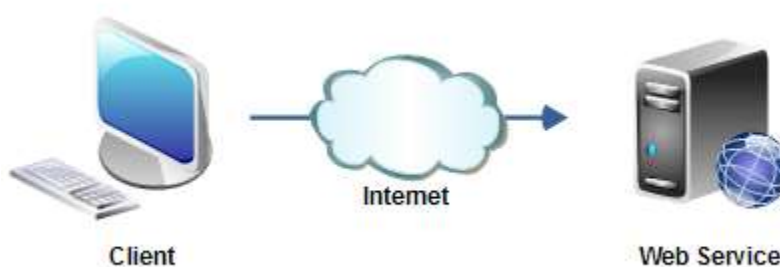


Figura 7. Comunicação web service

(CERAMI, 2002) Define Web service como um serviço que possibilita a comunicação de duas aplicações através de uma interface específica. Isso o caracteriza como um serviço disponível na internet para fornecer informações através da comunicação nos formatos dos protocolos disponíveis. Os Web services consistem em soluções utilizadas na integração de sistemas e na comunicação de aplicações diferentes.

Como formato de comunicação o web service dispõem de:

REST (*Representational State Transfer*)

Consiste em um protocolo de comunicação disponibilizado pelos web services baseados na hipermídia HTTP. Sua grande vantagem em relação aos outros formatos de comunicação é que ele não impõe restrição quanto ao formato da mensagem, sua restrição está assegurada apenas ao comportamento dos componentes envolvidos, assim sua maior vantagem é a flexibilidade, pois o desenvolvedor pode optar pelo formato mais adequado para troca de mensagens conforme suas necessidades, alguns dos formatos disponíveis para troca de dados são:

- JSON (*JavaScript Object Notation*);
- XML (*eXtensible Markup Language*)
- Texto puro.

SOAP (*Simple Object Access Protocol*)

Ao contrário do protocolo de transferência REST, o SOAP é baseado exclusivamente no formato XML para troca de dados, quando aplicado a um webservice geralmente usa-se o WSDL (*Web Services Description Language*) para descrever a estrutura das mensagens SOAP, sua desvantagem está ligada a demora de consumo de dados, visto que todas mensagens passarão por serialização e desserialização, o que tende a deixar o processo mais pesado se comparado ao REST.

Se tratando do âmbito empresarial os *web services* podem trazer inúmeras vantagens às empresas por meio de seus processos ligados, que garantem a eficiência na comunicação e principalmente torna a troca de mensagens mais dinâmica e segura visto que abole a intervenção humana.

CAPÍTULO 3 - ESTUDO DE CASO

3.1 - Identificação do problema

Baseando no cenário atual onde as empresas estão cada vez mais atentas às novidades externas, a estruturação de seu departamento de vendas possibilita ganhos em relação aos concorrentes, com um departamento bem estruturado a tendência é que atinjam todos os nichos de mercado ao qual atuam, pensando nessa maior fidelização e buscando aumentar o maior número de prospects as empresas estão julgando ser necessário cada vez mais investir em tecnologia, e uma das tecnologias que mais estão despontando no mercado são os sistemas CRM de gestão de clientes, para equipe de vendas.

Hoje em dia já fora rompida diversas barreiras, se analisarmos em relação a um período anterior os métodos de vendas e busca de fidelização de clientes eram ultrapassados, o que não proporcionava uma visão geral da real saúde do negócio, ou seja gerando prejuízo às empresas, os sistemas de gestão de relacionamento com os clientes estão ajudando a melhorar esse contexto.

Esses sistemas estão cada vez mais presentes no mercado e buscam facilitar as rotinas diárias dos vendedores externos, ganhando tempo em execução, porém nesse mesmo cenário ainda há empresas que não fazem o uso da tecnologia e permanecem com a baixa qualidade no atendimento e com vendedores externos cada vez mais sobrecarregados.

Um sistema CRM de força de vendas bem estruturado proporciona à empresa uma potencialização de toda a equipe de vendas, os gastos para aquisição de plataformas como essas são irrisórios em razão de todo o benefício proporcionado, empresas que ainda utilizam métodos arcaicos em rotinas de vendas tendem a não conseguir os mesmos resultados de empresas que utilizam a tecnologia a seu favor, e ainda sofrem com a perda de produtividade de seus vendedores externos, demora de recebimento de pedidos e falta de informação para mensuração de indicadores de venda.

A partir dessa situação, com o propósito de atingir essas empresas alvos neste trabalho é apresentado um sistema CRM de forças de vendas, uma solução robusta que possibilita o envio dos pedidos e o cadastro de novos clientes diretamente pelo representante comercial, a captação de dados do cliente, o estudo do perfil de compra, agilizando os processos internos da empresa, e garantindo a máxima eficácia do atendimento.

3.2 - Objetivos do trabalho

3.2.1 - Geral

Desenvolver um sistema mobile para envio e recepção de informações entre representante e empresa, captação e tratamento de informações relativas ao perfil de compra do cliente, sendo destinado a garantir melhor usabilidade para a equipe de vendas das empresas, proporcionando benefícios de melhor gestão e maior controle de rotinas diárias as mesmas.

3.2.2 - Específicos

- Levantar os requisitos funcionais e não funcionais do sistema;
- Definir as regras de negócio do sistema;
- Documentar os requisitos utilizando a linguagem de modelagem UML;
- Utilizar banco de dados SQLite;
- Implementar utilizando a linguagem de programação Java (Android);
- Utilizar ferramenta para elaboração dos diagramas de Caso de Uso e Classes.
- Implantar tendências de Marketing de relacionamento.

3.3 - Justificativa

O uso dos sistemas CRM de gestão de cliente: forças de vendas possibilitam às empresas ganhos na execução de simétricas diárias quanto em mensais, seu alto nível de abrangência e sua facilidade de uso no dia-a-dia, são pontos fundamentais nos modelos de negócios atuais, compreende-se que essas tecnologias são a base para que a organização possa ter êxito no mercado, atraindo e fidelizando o maior número de clientes, pois garantem a máxima eficácia em seu atendimento, destaca-se alguns ganhos ao âmbito empresarial: uma maior produtividade, reduções de custo, informações mais precisas, agilidade na execução de rotinas, análise de crédito de clientes, confiabilidade por parte do cliente, pedidos entregues sem atrasos, dentre outras; diante dessas tendências o uso desses sistemas torna-se peça chave na busca de melhores resultados para as empresas.

3.4 - Requisitos funcionais

a) Envio de pedidos

O sistema deve permitir aos representantes comerciais o envio de pedidos a empresa, sendo por descargas individuais ou grupais, informações como (prioridade de entrega, agendamento de entrega, data de faturamento) deverão ser elucidadas e transpassadas, sendo esses pedidos recebidos pelo ERP da empresa de forma a garantir um fluxo de comunicação mais preciso possível.

b) Validação de estoque

O sistema deve retornar dados relativos a quantidade em estoque dos itens no momento do envio dos pedidos, aceitando ou rejeitando de acordo com a disponibilidade.

c) Consulta de estoque

O sistema deve permitir aos representantes comerciais a consulta de estoque dos produtos comercializados pela empresa, a fim de garantir precisa comunicação entre o representante comercial e o cliente, coibindo possíveis transtornos em não entrega dos itens negociados.

d) Consulta de produtos

O sistema deve permitir aos representantes comerciais a consulta de informações dos produtos disponíveis para comercialização, informações como (imagens do produto) devem estar sempre acessíveis, para garantir um melhor fluxo de comunicação no momento da venda.

e) Consulta de histórico de relacionamento com o cliente

O sistema deve permitir aos representantes comerciais a consulta das faturas emitidas aos clientes aos quais atenderam, assim possibilitando um acompanhamento da historicidade de seus pedidos e verificação de possíveis erros relacionados a preço ou quantidade, também a consulta e acompanhamento dos títulos emitidos, permitindo que realizem a cobrança dos que se encontram vencidos.

f) Acompanhamento do perfil de consumo

O sistema deve permitir ao representante comercial visualizar dados relativos ao perfil de consumo do cliente, assim possibilitando ao mesmo decisões mais assertivas quanto a sugestões de compra.

3.5 - Requisitos não funcionais

a) Usabilidade

O sistema deve ser intuitivo e de fácil compreensão, botões deverão ser de fáceis visualizações possibilitando uma interação mais amigável entre o usuário e o sistema.

b) Linguagem de programação

O sistema deve ser desenvolvido em linguagem de programação java, dando amplitude a especificação para android, devido sua maior popularidade no cenário atual do mercado.

c) Mobilidade

O sistema deve ser mobile, o que garantirá ao representante comercial uma eficácia na comunicação com a facilidade de informações na palma de sua mão.

3.6 - Diagramas de Caso de Uso

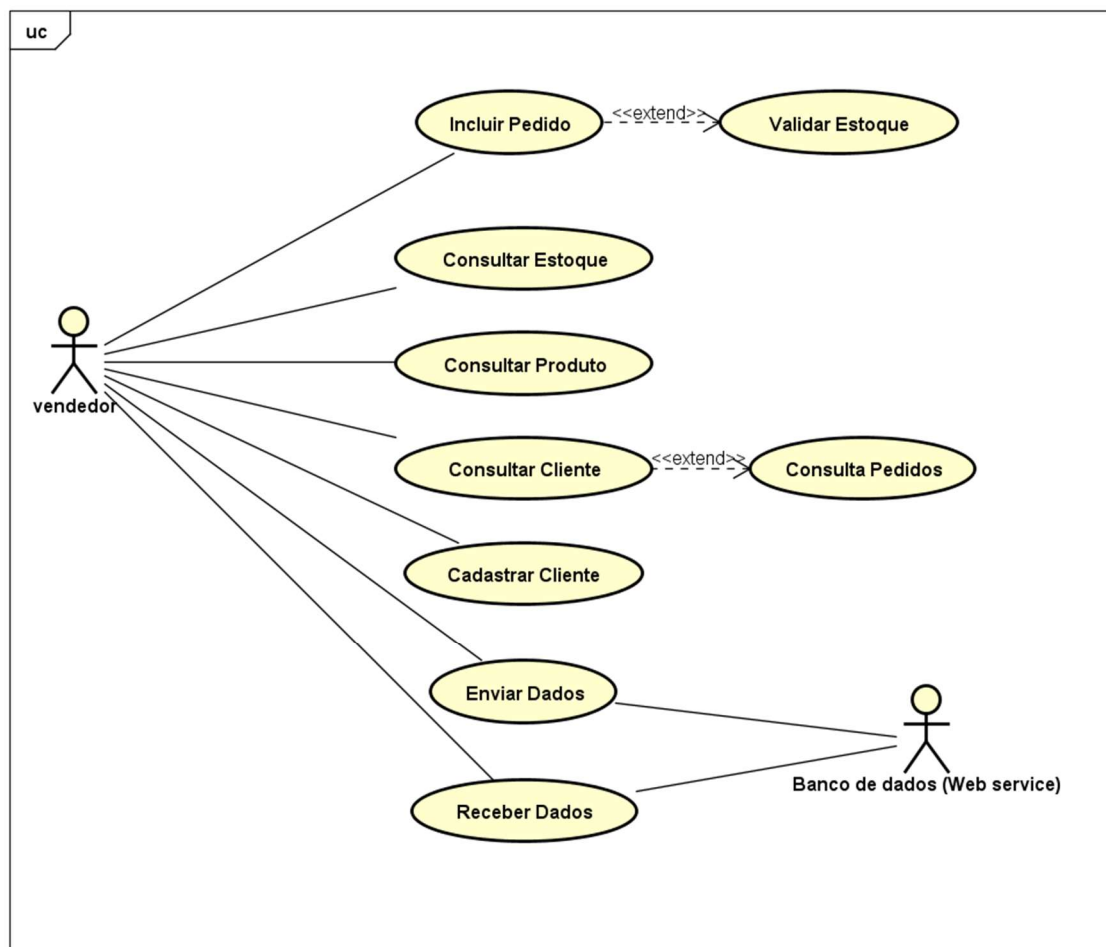


Figura 8. Modelo do Diagrama de caso de Uso

3.7 - Descrição dos casos de usos

[UC 01]: Incluir pedido

Descrição: Neste caso de uso o usuário (vendedor) deverá imputar os dados relativos ao pedido proposto e realizar a transmissão ao ERP, sendo recebida a validação do estoque do pedido o que caracteriza que há mercadoria disponível para atender à venda solicitada.

Atores envolvidos: Usuário do sistema (vendedores).

Pré-condição: O usuário deverá estar logado no sistema.

Cenário normal:

1. O sistema apresenta a tela de clientes;
2. O usuário seleciona o cliente;
3. O usuário seleciona o produto.
4. O usuário transmite o pedido;
5. O sistema valida o estoque do pedido.
6. O sistema devolve o retorno da validação ao usuário.

[UC 02]: Consultar estoque

Descrição: Neste caso de uso o usuário (vendedor) deverá selecionar o produto ao qual deseja verificar o saldo disponível em estoque, e consultar o estoque do mesmo junto ao ERP.

Atores envolvidos: Usuário do sistema (vendedores).

Pré-condição: O usuário deverá estar logado no sistema.

Cenário normal:

1. O sistema apresenta a tela de produtos;
2. O usuário seleciona o produto;
3. O usuário consulta o estoque do produto.
4. O sistema devolve o saldo disponível do produto consultado.

[UC 03]: Consultar produto

Descrição: Neste caso de uso o usuário (vendedor) deverá selecionar o produto ao qual deseja buscar os dados cadastrais, e consultar as informações do mesmo sendo transferido dados disponíveis do ERP referentes ao produto solicitado.

Atores envolvidos: Usuário do sistema (vendedores).

Pré-condição: O usuário deverá estar logado no sistema.

Cenário normal:

1. O sistema apresenta a tela de produtos;
2. O usuário seleciona o produto;
3. O usuário consulta as informações do produto.
4. O sistema devolve todos os dados inerentes ao produto consultado.

[UC 04]: Consultar cliente

Descrição: Neste caso de uso o usuário (vendedor) deverá selecionar o cliente ao qual deseja buscar informações.

Atores envolvidos: Usuário do sistema (vendedores).

Pré-condição: O usuário deverá estar logado no sistema.

Cenário normal:

1. O sistema apresenta a tela de clientes;
2. O usuário seleciona o cliente;
3. O usuário consulta as informações do cliente.
4. O sistema devolve os dados relativos ao cliente consultado.

[UC 05]: Consulta pedidos

Descrição: Neste caso de uso o usuário (vendedor) realiza a consulta dos últimos pedidos efetuados pelo cliente.

Atores envolvidos: Usuário do sistema (vendedores).

Pré-condição: O usuário deverá estar logado no sistema.

[UC 06]: Cadastrar cliente

Descrição: Neste caso de uso o usuário (vendedor) deverá imputar dados relativos ao cliente a ser cadastrado.

Atores envolvidos: Usuário do sistema (vendedores).

Pré-condição: O usuário deverá estar logado no sistema.

Cenário normal:

1. O sistema apresenta a tela de cadastro de cliente;
2. O usuário informa os dados inerentes ao novo cliente a ser cadastrado;
3. O usuário transmite o cliente cadastrado.
4. O sistema devolve a confirmação do êxito no cadastro.

[UC 07]: Enviar Dados

Descrição: Neste caso de uso o usuário (vendedor) enviará dados dos pedidos realizados ao banco de dados da empresa por meio de conexão via web service.

Atores envolvidos: Usuário do sistema (vendedores), web service.

Pré-condição: O usuário deverá estar logado no sistema, e deverá ter conexão de internet disponível.

Cenário normal:

1. O usuário seleciona a opção de envio de dados.
2. Dados são enviados ao web service;
3. Dados são persistidos no banco de dados.
4. O sistema devolve a validação dos dados imputados.

[UC 08]: Receber Dados

Descrição: Neste caso de uso o usuário (vendedor) receberá da empresa novas informações inerentes a sua carteira de clientes.

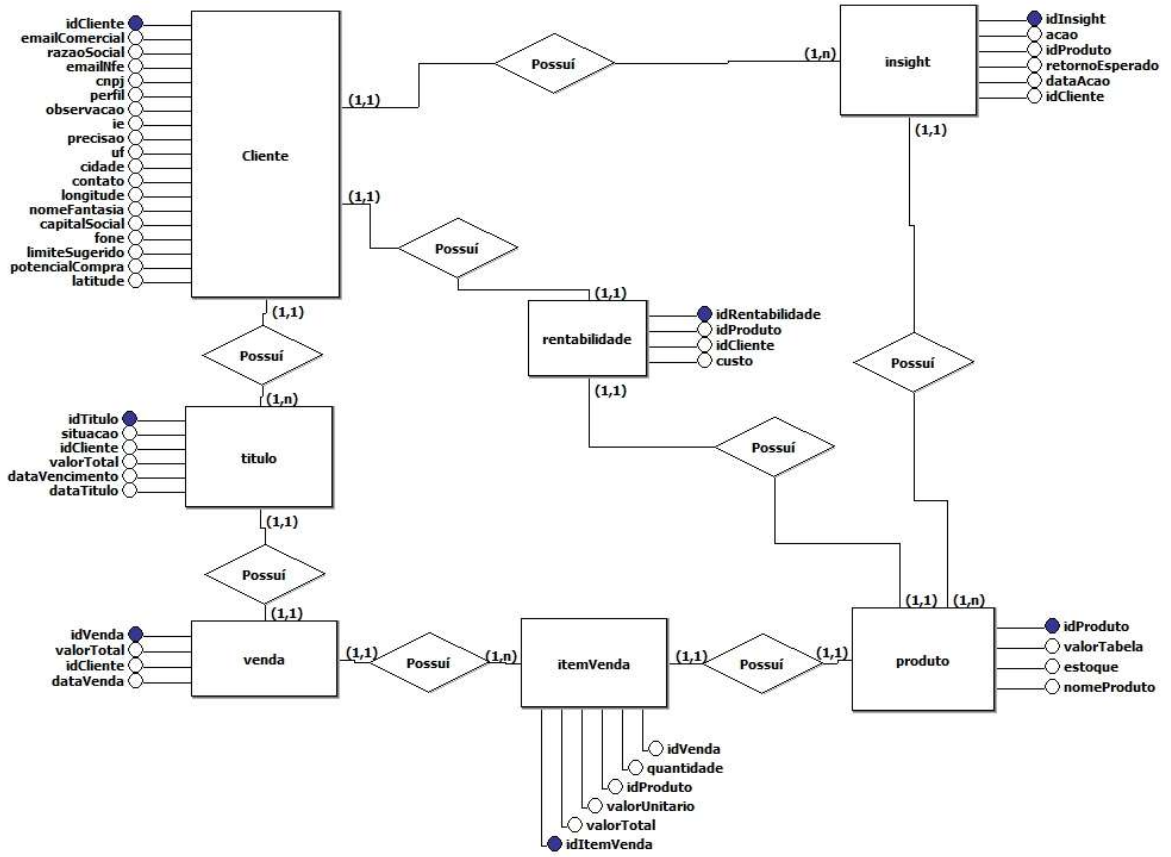
Atores envolvidos: Usuário do sistema (vendedores), web service.

Pré-condição: O usuário deverá estar logado no sistema, e deverá ter conexão de internet disponível.

Cenário normal:

1. O usuário seleciona a opção de recebimento de dados.
2. Dados são recebidos do web service;
3. Dados são persistidos no banco de dados local da aplicação.
4. O sistema devolve a validação dos dados imputados.

3.8 - Modelo de Entidade e Relacionamento



3.9 - Diagrama de classes

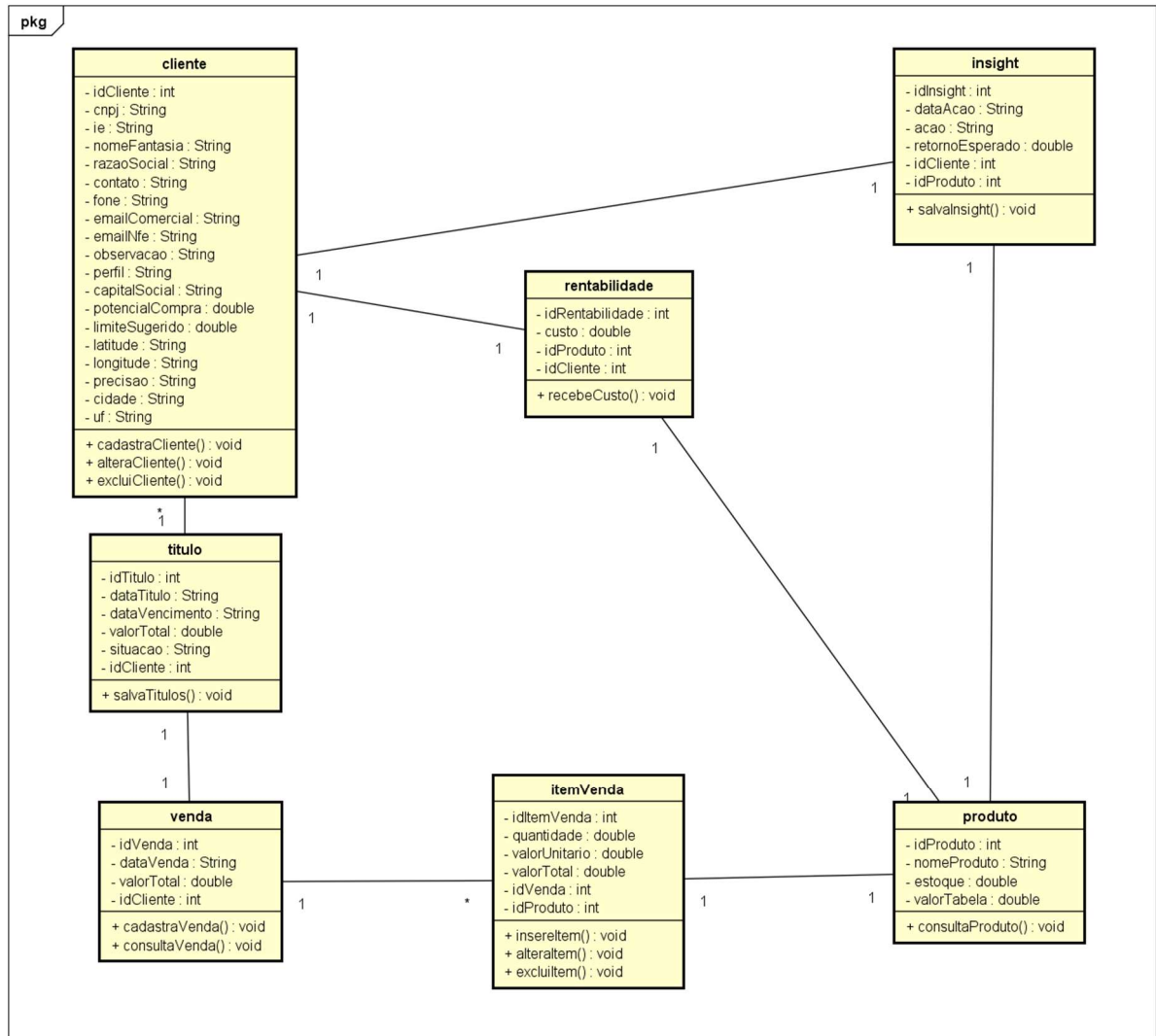


Figura 9. Diagrama de Classes

3.10 – Diagrama de Sequência

Na figura 10 abaixo, está representado o diagrama de sequência para a rotina de consulta de estoque, todos os demais diagramas de classe sequeem o mesmo modelo, a exposição deste fez se necessário apenas para fins de exemplificação da aplicação dos diagramas e estereótipos da UML para o trabalho proposto.

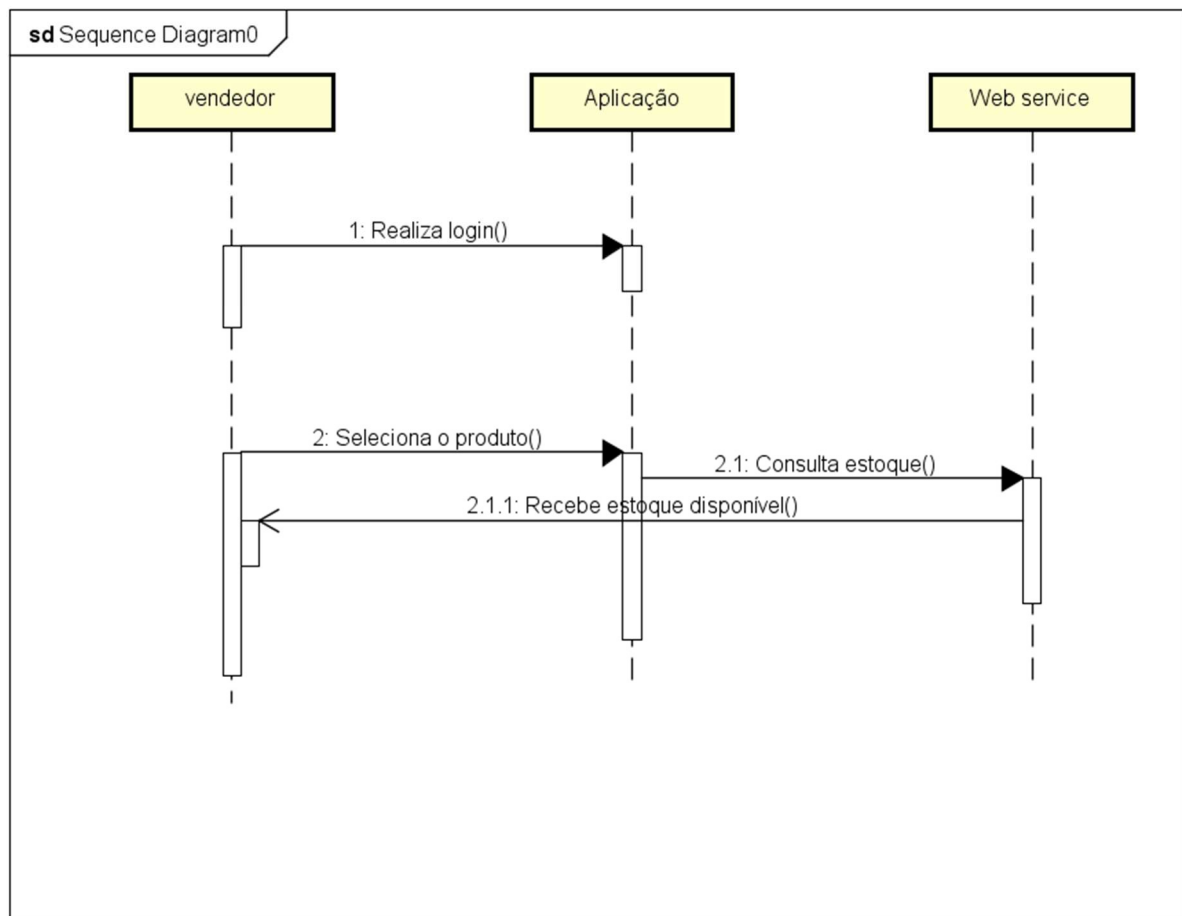


Figura 10. Diagrama de Sequência

3.11 - Prototipação

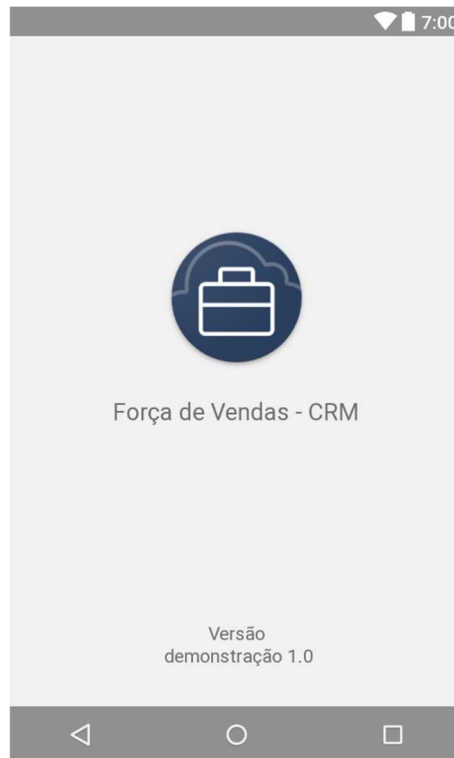


Figura 11. Tela Splash

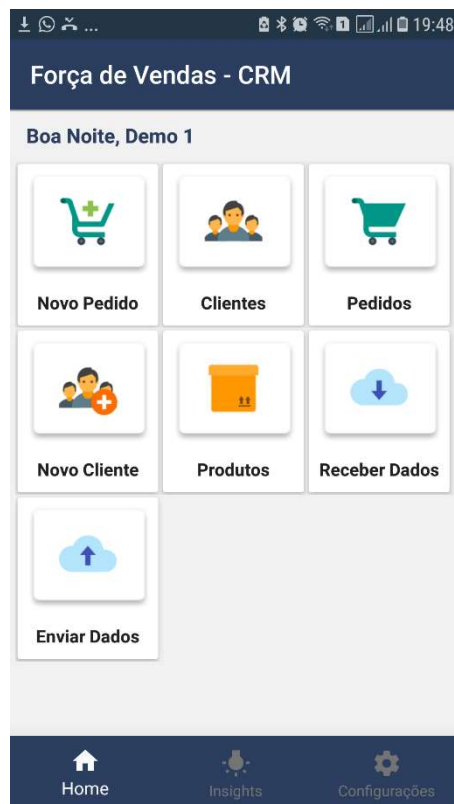


Figura 12. Tela Principal



Figura 13. Tela Insight

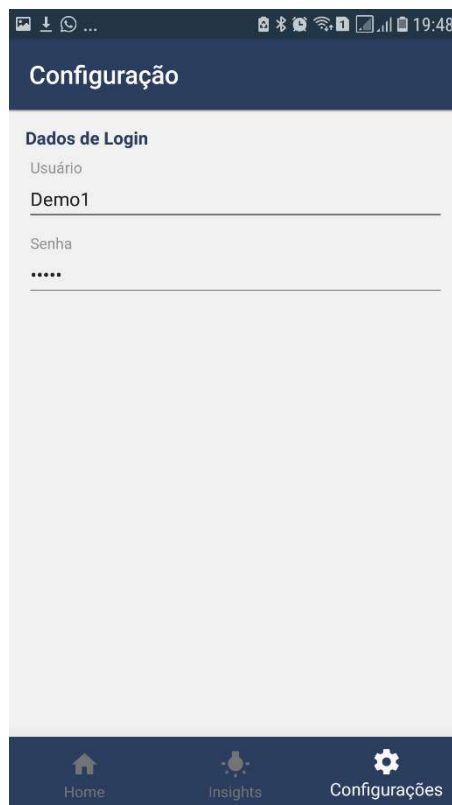


Figura 14. Tela Configuração



Figura 15. Tela Detalhe Cliente

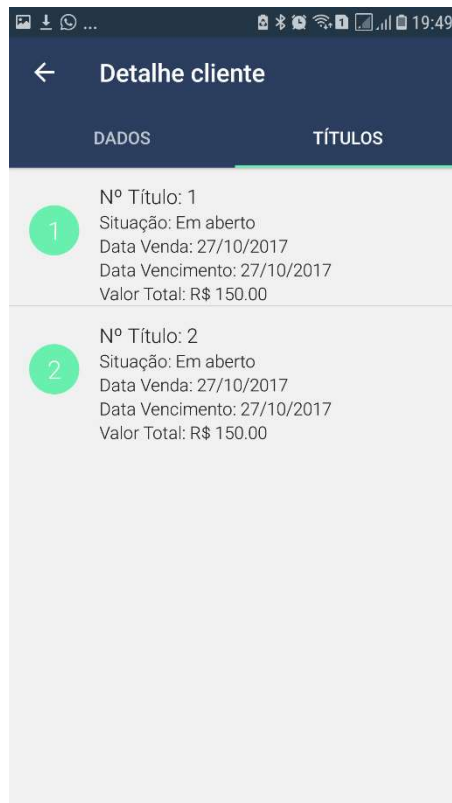
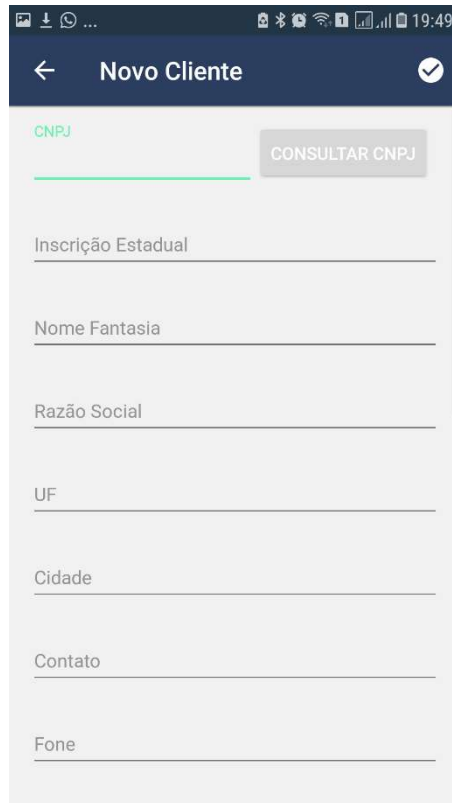


Figura 16. Tela Detalhe Cliente Títulos



Novo Cliente

CNPJ

Inscrição Estadual

Nome Fantasia

Razão Social

UF

Cidade

Contato

Fone

Figura 17. Tela Novo Cliente



Selecione o cliente

- Código: 1
Razão Social: GLOBAL ALIMENTOS
Nome Fantasia: GLOBAL ALIMENTOS
Cidade: Goiânia-GO
- Código: 2
Razão Social: MAIS SABOR S/A
Nome Fantasia: MAIS SABOR
Cidade: Brasília-DF
- Código: 3
Razão Social: LINEA ALIMENTOS S/A
Nome Fantasia: LINEA ALIMENTOS
Cidade: Goiânia-GO
- Código: 4
Razão Social: PLENO ALIMENTOS S/A
Nome Fantasia: PLENO ALIMENTOS
Cidade: Goiânia-GO
- Código: 5
Razão Social: INOVA FOODS S/A
Nome Fantasia: INOVA FOODS
Cidade: Goiânia-GO

Figura 18. Tela Novo Pedido - Seleção Cliente

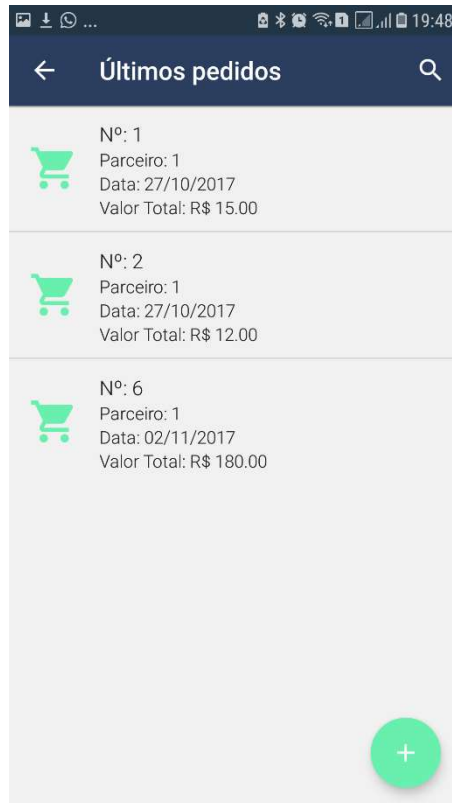


Figura 19. Tela Novo Pedido - Últimos Pedidos



Figura 20. Tela Novo Pedido - Seleção Produto

The screenshot shows the 'Novo pedido' screen with the following details:

- Rentabilidade Item:** Ótima
- Estoque:** 10.00 KG
- Valor Tabela:** R\$ 12.00
- Item 1:** PIZZA DE FRANGO
- Unidade:** CP
- Valor Venda:** 12.0
- Quantidade:** 1 (with minus and plus buttons)
- Valor Total Item:** R\$ 12.00
- Quantidade Total:** 1 CP
- Summary:** Valor Total Pedido R\$ 12.00 and a FINALIZAR button.

Figura 21. Tela Novo Pedido - Finalização Pedido

The screenshot shows the 'Produtos' screen with a list of items:

- Item 1:** Código: 1, Nome: LASANHA DE FRANGO, Estoque: 20.00, Valor Tabela: R\$ 15.00
- Item 2:** Código: 2, Nome: PIZZA DE FRANGO, Estoque: 10.00, Valor Tabela: R\$ 12.00
- Item 3:** Código: 3, Nome: NUGGETS RECHEADOS, Estoque: 5.00, Valor Tabela: R\$ 8.00
- Item 4:** Código: 4, Nome: FRANGO RESFRIADO, Estoque: 200.00, Valor Tabela: R\$ 5.00
- Item 5:** Código: 5, Nome: PEITO DE FRANGO, Estoque: 100.00, Valor Tabela: R\$ 5.00

A dark grey toast notification at the bottom of the list reads: "Item adicionado: 1".

Figura 22. Tela Novo Pedido - Finalização Pedido - Adição Item

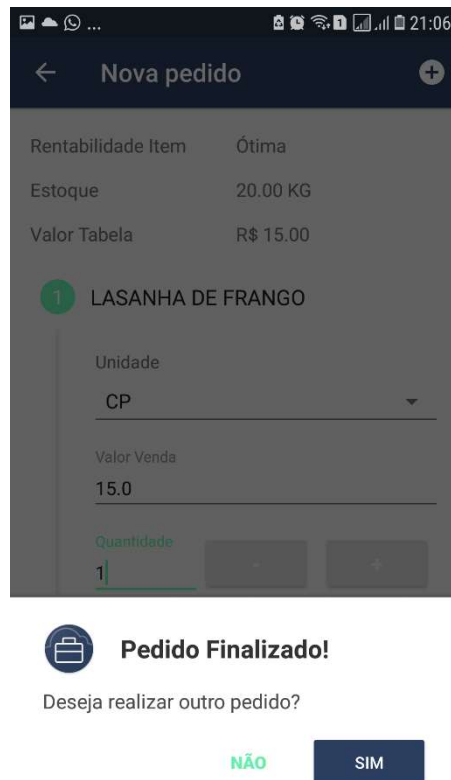


Figura 23. Tela Novo Pedido - Finalização Pedido – Confirmação



Figura 24. Tela Consulta Cliente



Código	Nome	Estoque	Valor Tabela
1	LASANHA DE FRANGO	20.00	R\$ 15.00
2	PIZZA DE FRANGO	10.00	R\$ 12.00
3	NUGGETS RECHEADOS	5.00	R\$ 8.00
4	FRANGO RESFRIADO	200.00	R\$ 5.00
5	PEITO DE FRANGO	100.00	R\$ 6.00

Figura 25. Tela Consulta Produtos

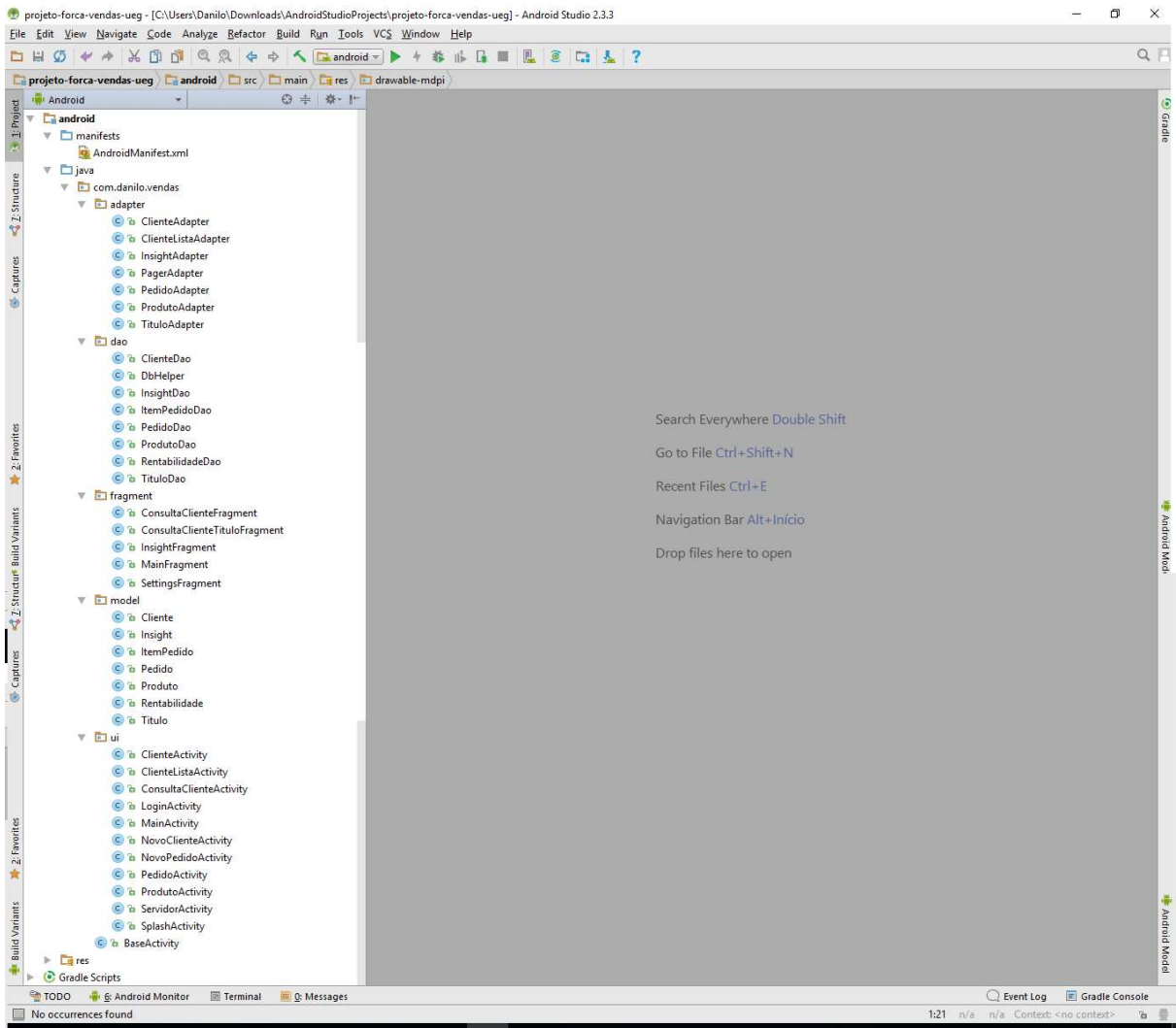


Figura 26. Classes Aplicação - Android Studio

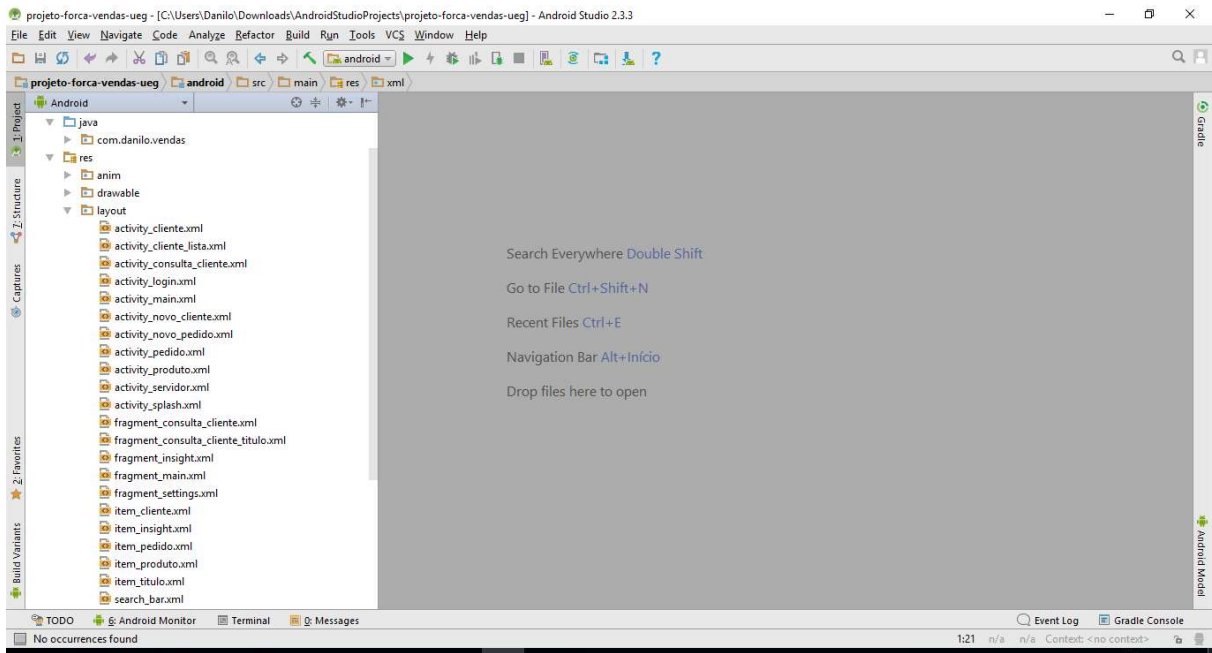


Figura 27. Layout Aplicação - Android Studio

CONSIDERAÇÕES FINAIS

O objetivo desse trabalho foi discorrer sobre os sistemas forças de vendas e sua aplicação no processo de vendas, trazendo uma abordagem sobre como as tecnologias de CRM vem trazendo inovações aos contextos dos mesmos, também fora abordado como as soluções que agem por trás dos sistemas forças de vendas funcionam, abordando a definição do ERP, *web service*.

No decorrer de todo desenvolvimento do projeto, pode-se notar que o mercado baseado no proposito referido de seu contexto está em ritmo intenso de expansão, sendo uma ótima oportunidade de investimento por parte de desenvolvimento de solução mobile. O projeto teve o intuito de abordar a contextualização da venda e as vantagens que são proporcionadas pela sua melhor exploração utilizando as técnicas das soluções CRM, que garantem a empresa o aproveitamento de oportunidades e ganhos em vendas.

Toda metodologia de pesquisa baseou-se em pesquisas bibliográficas para um melhor embasamento teórico, ainda que alguns assuntos não são elucidados tão abertamente em livros e artigos, como foi o caso encontrado dos sistemas de força de venda.

O primeiro passo do trabalho foi identificar, através de estudos o contexto do tema e como ele vem sendo aplicado ao cenário explorado, a aplicação foi desenvolvida tentando assemelhar aos padrões já existentes das mesmas soluções disponíveis no mercado, até para proporcionar uma melhor adaptação aos usuários.

Paralelamente fora abordado e enriquecido o conteúdo por meio de conhecimento de soluções semelhantes no mercado, o fator de contextualização de tecnologias como ERP e web services trouxe conteúdo mais sucinto a todo enredo abordado e proporcionou um amplo engajamento sobre o assunto.

As questões envolvidas em relação ao pouco tempo de desenvolvimento e pesquisa disponíveis proporcionou um desafio para abordagem do vasto conteúdo explorado, mas do que isso, toda abordagem e pesquisa sobre a base teórica possibilitou vislumbrar cenários altamente factíveis.

A conclusão final é que o software possibilitará uma abordagem diferente no processo de vendas, mas vale frisar que não é pioneiro nesse quesito, apesar de que não será um desafio inexplorado seu continuamento, visto da rica oportunidade, ele ainda não está finalizado, a ideia de sua adaptação para os métodos de envio e recepção de informação variarão de acordo com os padrões de software da empresa a ser implantada, todas suas

soluções e telas foram desenvolvidas visando a melhor usabilidade, agora cabe investir no acúmulo de mais opções para um diferencial em relação aos concorrentes do mesmo segmento que atuam no mercado atual.

REFERÊNCIAS BIBLIOGRÁFICAS

27ª Pesquisa Anual de Administração e Uso da Tecnologia da informação nas empresas, realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP).

BATTISTI, Júlio César Fabris. **Noções da linguagem SQL para consultas**. 2006. Disponível em: <http://www.linhadecodigo.com.br/artigo/165/noco-es-da-linguagem-sql-para-consultas.aspx>. Acesso em 13 dez. 2017.

BROWN, Stanley A, **CRM - Customer Relationship Management: Uma Ferramenta estratégica para o mundo E-business**. Makron Books (2001).

CERAMI, Ethan. **Web services Essentials: distributed applications with XML-RPC, SOAP, UDDI & WSDL**. O'Reilly Media, Incorporated, 2002.

DATE, C. J. **INTRODUÇÃO A SISTEMAS DE BANCOS DE DADOS**. 8. ed. Rio de Janeiro: Elsevier, 2003.

Eduardo Bezerra. **Princípios de análise e Projeto de Sistemas com UML**. Rio de Janeiro, 2007- 9º Reimpressão

ELMASRI, Ramez / NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. São Paulo: Pearson Addison Wesley, 2011.

Gilleanes T. A. Guedes. **UML 2, uma abordagem prática**. 2º ed São Paulo: Novatec Editora, 2011.

GOOGLE INC. **What is Android?** Disponível em: <http://developer.android.com/guide/basics/what-is-android.html>. Acesso em: 12 dez. 2017

GOOGLE INC. **Publicar o aplicativo**. Disponível em: <https://developer.android.com/studio/publish/index.html?hl=pt-br>. Acesso em: 12 dez. 2017

Greenberg, P. (2001), **"CRM na Velocidade da Luz"**, Rio de Janeiro: Campus.

Hicks, Donald A. (1995). **"The ERP maze"**, IIE Solutions, Agosto/95, p.13-16.

KOTLER, Philip – **Administração de Marketing** – 10ª Edição, 7ª reimpressão – Tradução Bazán Tecnologia e Linguística; revisão técnica Arão Sapiro. São Paulo: Prentice Hall, 2000.

KRISHNAN, M. S., e Fornell, C. (2005), **"Why do customer relationship management applications affect customer satisfaction?"**, Journal of Marketing, Vol. 69 (October), pp. 201–209.

LEE, Wei-Meng. **Introdução ao Desenvolvimento de Aplicativos para Android**. 1. E. Rio de Janeiro: Editora Ciência Moderna Ltda, 2011.

LOBO, Alexandre. **Marketing de relacionamento**. São Paulo: Seminários SSJ, 2002.

MAÑAS, A.V. **Administração de Sistemas de Informação**. São Paulo: Érica, 2002.

NEWELL, F. Lealdade.com – **CRM: o gerenciamento das relações com o consumidor na era do marketing pela internet**. São Paulo: Makron Books, 2006.

PEPPERS, Don; ROGERS, Martha. **CRM ganha força no país**. HSM Management, v. 7, n. 38, p.49-59, maio/jun. 2003.

SOUKI, Omar. **As 7 chaves da fidelização de clientes**. São Paulo: Harbra, 2006.

ANEXOS

Activity (Classe) Model:

```
1 package com.danilo.vendas.model;
2 /**
3  * Classe responsável pelo transporte dos dados entre a interface(tela) e Banco de Dados
4  */
5 public class Pedido {
6     private Integer idVenda;
7     private String dataVenda;
8     private Double valorTotal;
9     private Integer idCliente;
10    public Pedido() {
11    }
12    public Pedido(Integer idVenda, String dataVenda, Double valorTotal, Integer idCliente)
13    {
14        this.idVenda = idVenda;
15        this.dataVenda = dataVenda;
16        this.valorTotal = valorTotal;
17        this.idCliente = idCliente;
18    }
19    public Integer getIdVenda() {
20        return idVenda;
21    }
22    public void setIdVenda(Integer idVenda) {
23        this.idVenda = idVenda;
24    }
25    public String getDataVenda() {
26        return dataVenda;
27    }
28    public void setDataVenda(String dataVenda) {
29        this.dataVenda = dataVenda;
30    }
31    public Double getValorTotal() {
32        return valorTotal;
33    }
34    public void setValorTotal(Double valorTotal) {
35        this.valorTotal = valorTotal;
36    }
37    public Integer getIdCliente() {
38        return idCliente;
39    }
40    public void setIdCliente(Integer idCliente) {
41        this.idCliente = idCliente;
42    }
43 }
```

Activity (Classe) DAO:

```

1 package com.danilo.vendas.dao;
2 import android.content.ContentValues;
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import com.danilo.vendas.model.Pedido;
7 import java.util.ArrayList;
8 import java.util.List;
9 /**
10 * Classe responsável pela manipulação de dados
11 */
12 public class PedidoDao {
13     private static String nomeTabela = "venda";
14     SQLiteDatabase db;
15     DBHelper dbHelper;
16     ContentValues values;
17     public PedidoDao(Context context) {
18         dbHelper = new DBHelper(context);
19     }
20     public boolean insert(Pedido vo) {
21         db = dbHelper.getWritableDatabase();
22         values = new ContentValues();
23         values.put("idVenda", vo.getIdVenda());
24         values.put("dataVenda", vo.getDataVenda());
25         values.put("valorTotal", vo.getValorTotal());
26         values.put("idCliente", vo.getIdCliente());
27         return db.insert(nomeTabela, null, values) > 0;
28     }
29     public boolean delete(Pedido vo) {
30         db = dbHelper.getWritableDatabase();
31         if (db.delete(nomeTabela, "idVenda=?", new String[]{vo.getIdVenda().toString()}) >
32 0) {
33             return true;
34         }
35         return false;
36     }
37     public boolean update(Pedido vo) {
38         db = dbHelper.getWritableDatabase();
39         values = new ContentValues();
40         values.put("idVenda", vo.getIdVenda());
41         values.put("dataVenda", vo.getDataVenda());
42         values.put("valorTotal", vo.getValorTotal());
43         values.put("idCliente", vo.getIdCliente());
44         if (db.update(nomeTabela, values, "idVenda=?", new
45 String[]{vo.getIdVenda().toString()}) > 0) {
46             return true;
47         }
48         return false;

```

```

47 }
48 public List<Pedido> getAll(String sql) {
49     db = dbHelper.getWritableDatabase();
50     Cursor rs = db.rawQuery(sql, null);
51     List<Pedido> lista = new ArrayList();
52     while (rs.moveToNext()) {
53         Pedido vo = new Pedido();
54         vo.setIdVenda(rs.getInt(rs.getColumnIndex("idVenda")));
55         vo.setDataVenda(rs.getString(rs.getColumnIndex("dataVenda")));
56         vo.setValorTotal(rs.getDouble(rs.getColumnIndex("valorTotal")));
57         vo.setIdCliente(rs.getInt(rs.getColumnIndex("idCliente")));
58         lista.add(vo);
59     }
60     rs.close();
61     db.close();
62     return lista;
63 }

```

Activity (Classe) BDHelper:

```

1 package com.danilo.vendas.dao;
2 import android.content.Context;
3 import android.database.sqlite.SQLiteDatabase;
4 import android.database.sqlite.SQLiteOpenHelper;
5 /**
6  * Classe responsável pela criação do banco de dados
7  */
8 public class DbHelper extends SQLiteOpenHelper {
9     private static final String dbName = "dbvendas.db";
10    private static final int versao = 1;
11    private static final String cliente = "CREATE TABLE cliente (idCliente INTEGER
PRIMARY KEY, cnpj STRING, ie STRING, nomeFantasia STRING, razaoSocial STRING,
contato STRING, fone STRING, emailComercial STRING, emailNfe STRING, observacao
STRING, perfil STRING, capitalSocial DOUBLE, potencialCompra DOUBLE,
limiteSugerido DOUBLE, latitude STRING, longitude STRING, precisao STRING, cidade
STRING, uf STRING)";
12    private static final String produto = "CREATE TABLE produto (idProduto INTEGER
PRIMARY KEY, nomeProduto STRING, estoque DOUBLE, valorTabela DOUBLE)";
13    private static final String venda = "CREATE TABLE venda (idVenda INTEGER
PRIMARY KEY, dataVenda STRING, valorTotal DOUBLE, idCliente INTEGER
REFERENCES cliente (idCliente))";
14    private static final String itemVenda = "CREATE TABLE itemVenda (idItemVenda
INTEGER PRIMARY KEY, quantidade DOUBLE, valorUnitario DOUBLE, valorTotal
DOUBLE, idVenda INTEGER REFERENCES venda (idVenda), idProduto INTEGER
REFERENCES produto (idProduto))";
15    private static final String titulo = "CREATE TABLE titulo (idTitulo INTEGER
PRIMARY KEY, dataTitulo STRING, dataVencimento STRING, valorTotal DOUBLE,
situacao STRING, idCliente INTEGER REFERENCES cliente (idCliente))";

```

```

16     private static final String rentabilidade = "CREATE TABLE rentabilidade
(idRentabilidade INTEGER PRIMARY KEY, custo DOUBLE, idProduto INTEGER
REFERENCES produto (idProduto), idCliente INTEGER REFERENCES cliente
(idCliente))";
17     private static final String insight = "CREATE TABLE insight (idInsight INTEGER
PRIMARY KEY, dataAcao STRING, acao STRING, retornoEsperado DOUBLE, idCliente
INTEGER REFERENCES cliente (idCliente), idProduto INTEGER REFERENCES produto
(idProduto))";
18     private static final String insertCliente = "INSERT INTO cliente VALUES (1,
'01.111.111-1111/60', 'ISENTO', 'GLOBAL ALIMENTOS', 'GLOBAL ALIMENTOS',
'JOÃO', '62-999998888', 'joão@global.com.br', 'joão@global.com.br', 'Vai mudar em breve',
'Industrias', 5000.000, 6000.000, 4000.000, '-16.000', '-49.000', '5 mts', 'Goiânia', 'GO'), (2,
'02.222.222-2222/60', 'ISENTO', 'MAIS SABOR', 'MAIS SABOR S/A', 'JOSÉ', '62-
777778888', 'jose@maissabor.com.br', 'jose@maissabor.com.br', 'Loja nova', 'FoodService',
4000.000, 5000.000, 3000.000, '-16.000', '-49.000', '5 mts', 'Brasília', 'DF'), (3, '03.3333.333-
3333/60', 'ISENTO', 'LINEA ALIMENTOS', 'LINEA ALIMENTOS S/A', 'JOÃO', '62-
999998888', 'joão@linea.com.br', 'joão@linea.com.br', 'Sob nova direção', 'Industrias',
5000.000, 6000.000, 4000.000, '-16.000', '-49.000', '5 mts', 'Goiânia', 'GO'), (4, '04.444.444-
4444/60', 'ISENTO', 'PLENO ALIMENTOS', 'PLENO ALIMENTOS S/A', 'JOÃO', '62-
999998888', 'joão@pleno.com.br', 'joão@pleno.com.br', 'Vai mudar em breve', 'Industrias',
5000.000, 6000.000, 4000.000, '-16.000', '-49.000', '5 mts', 'Goiânia', 'GO'), (5, '05.555.555-
5555/60', 'ISENTO', 'INOVA FOODS', 'INOVA FOODS S/A', 'JOÃO', '62-999998888',
'joão@inovafoods.com.br', 'joão@inovafoods.com.br', 'Vai mudar em breve', 'FoodService',
1000.000, 1000.000, 1000.000, '-16.000', '-49.000', '5 mts', 'Goiânia', 'GO');
19     private static final String insertProduto = "INSERT INTO produto VALUES (1,
'LASANHA DE FRANGO', 20.000, 15.00), (2, 'PIZZA DE FRANGO', 10.000, 12.00), (3,
'NUGGETS RECHEADOS', 5.000, 8.00), (4, 'FRANGO RESFRIADO', 200.000, 5.00), (5,
'PEITO DE FRANGO', 100.000, 6.00)";
20     private static final String insertVenda = "INSERT INTO venda VALUES (1,
'27/10/2017', 15.00, 1), (2, '27/10/2017', 12.00, 1), (3, '27/10/2017', 15.00, 2), (4, '27/10/2017',
5.00, 2), (5, '27/10/2017', 5.00, 3)";
21     private static final String insertItemVenda = "INSERT INTO itemVenda VALUES (1,
10, 15.000, 150.000, 1, 1), (2, 10, 15.000, 150.000, 1, 1), (3, 10, 15.000, 150.000, 2, 2), (4, 10,
15.000, 150.000, 2, 2), (5, 10, 15.000, 150.000, 3, 3)";
22     private static final String insertTitulo = "INSERT INTO titulo VALUES (1, '27/10/2017',
'28/10/2017', 150.000, 'Em aberto', 1), (2, '27/10/2017', '28/10/2017', 150.000, 'Em aberto', 1),
(3, '27/10/2017', '28/10/2017', 150.000, 'Em aberto', 2), (4, '27/10/2017', '28/10/2017',
150.000, 'Baixado', 2), (5, '27/10/2017', '28/10/2017', 150.000, 'Baixado', 3)";
23     private static final String insertRentabilidade = "INSERT INTO rentabilidade VALUES
(1, 15.00, 1, 1), (2, 11.00, 1, 2), (3, 10.000, 1, 3), (4, 4.50, 1, 4)";
24     private static final String insertInsight = "INSERT INTO insight VALUES (1,
'27/10/2017', 'OFERECER NUGGETS', 15.00, 1, 3), (2, '27/10/2017', 'AUMENTAR
VOLUME DE VENDA DE NUGGETS', 15.00, 1, 3), (3, '27/10/2017', 'DIRECIONAR
FOCO DE VENDA PARA FRANGO RESFRIADO', 15.00, 1, 3)";
25     public DBHelper(Context ctx) {
26         super(ctx, dbName, null, versao);
27     }
28     @Override
29     public void onCreate(SQLiteDatabase db) {
30         db.execSQL(cliente);

```

```

31 db.execSQL(produto);
32 db.execSQL(venda);
33 db.execSQL(itemVenda);
34 db.execSQL(titulo);
35 db.execSQL(rentabilidade);
36 db.execSQL(insight);
37 db.execSQL(insertCliente);
38 db.execSQL(insertProduto);
39 db.execSQL(insertVenda);
40 db.execSQL(insertItemVenda);
41 db.execSQL(insertTitulo);
42 db.execSQL(insertRentabilidade);
43 db.execSQL(insertInsight);
44 }
45 @Override
46 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
47 db.execSQL("DROP TABLE IF EXISTS cliente");
48 db.execSQL("DROP TABLE IF EXISTS produto");
49 db.execSQL("DROP TABLE IF EXISTS venda");
50 db.execSQL("DROP TABLE IF EXISTS itemVenda");
51 db.execSQL("DROP TABLE IF EXISTS titulo");
52 db.execSQL("DROP TABLE IF EXISTS rentabilidade");
53 db.execSQL("DROP TABLE IF EXISTS insight");
54 onCreate(db);
55 }

```

Activity (Classe) Adapter:

```

1 package com.danilo.vendas.adapter;
2 import android.content.Context;
3 import android.support.v4.content.ContextCompat;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.BaseAdapter;
8 import android.widget.ImageView;
9 import android.widget.TextView;
10 import com.danilo.vendas.R;
11 import com.danilo.vendas.model.Pedido;
12 import java.util.List;
13 /**
14  * Classe responsável pela composição das Listas
15  */
16 public class PedidoAdapter extends BaseAdapter {
17     private Context ctx;
18     private List<Pedido> itens;
19     private LayoutInflater inflater;
20     private String filtro;
21     private List<Pedido> lista;

```

```

22 public PedidoAdapter(Context context, List<Pedido> itens) {
23     this.ctx = context;
24     this.itens = itens;
25     this.lista = itens;
26     this.filtro = filtro;
27     this.layoutInflater = LayoutInflater.from(context);
28 }
29 @Override
30 public int getCount() {
31     return this.lista.size();
32 }
33 @Override
34 public Object getItem(int position) {
35     return this.lista.get(position);
36 }
37 @Override
38 public long getItemId(int position) {
39     return (long) position;
40 }
41 @SuppressWarnings("WrongConstant")
42 @Override
43 public View getView(int position, View convertView, ViewGroup parent) {
44     Pedido vo = (Pedido) getItem(position);
45     ViewHolder holder;
46     if (convertView == null) {
47         convertView = ((LayoutInflater)
this.ctx.getSystemService("layout_inflater")).inflate(R.layout.item_pedido, null);
48         holder = new ViewHolder(convertView);
49         convertView.setTag(holder);
50     } else {
51         holder = (ViewHolder) convertView.getTag();
52     }
53     holder.tvNumeroPedido.setText(vo.getIdVenda().toString());
54     holder.tvParceiro.setText(vo.getIdCliente().toString());
55     holder.tvDataPedido.setText(vo.getDataVenda().toString());
56     holder.tvValorTotal.setText("R$ " + vo.getValorTotal().toString() + "0");
57     if (vo.getIdCliente().toString().substring(0, 1).toUpperCase().equals("S")) {
58         holder.imageEmail.setColorFilter(ContextCompat.getColor(ctx,
R.color.colorPrimary));
59     } else {
60         holder.imageEmail.setColorFilter(ContextCompat.getColor(ctx,
R.color.colorAccent));
61     }
62     return convertView;
63 }
64 static class ViewHolder {
65     //Declaração dos atributos
66     private ImageView imageEmail;
67     private TextView tvNumeroPedido;
68     private TextView tvParceiro;

```

```

69     private TextView tvDataPedido;
70     private TextView tvValorTotal;
71     public ViewHolder(View view) {
72         imageEmail = (ImageView) view.findViewById(R.id.imageEmail);
73         tvNumeroPedido = (TextView) view.findViewById(R.id.tvNumeroPedido);
74         tvParceiro = (TextView) view.findViewById(R.id.tvParceiro);
75         tvDataPedido = (TextView) view.findViewById(R.id.tvDataPedido);
76         tvValorTotal = (TextView) view.findViewById(R.id.tvValorTotal);
77     }
78 }

```

Activity (Classe) Main:

```

1 package com.danilo.vendas.ui;
2 import android.os.Bundle;
3 import android.support.annotation.IdRes;
4 import android.support.v4.app.Fragment;
5 import com.danilo.vendas.BaseActivity;
6 import com.danilo.vendas.R;
7 import com.danilo.vendas.fragment.InsightFragment;
8 import com.danilo.vendas.fragment.MainFragment;
9 import com.danilo.vendas.fragment.SettingsFragment;
10 import com.roughike.bottombar.BottomBar;
11 import com.roughike.bottombar.OnTabSelectListener;
12 /**
13  * Classe responsável pela manipulação de dados
14  */
15 public class MainActivity extends BaseActivity {
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20         //Lançando o fragment na act
21         if (savedInstanceState == null) {
22             Fragment f = new MainFragment();
23             getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
24 f).commit();
25         }
26         BottomBar bottomBar = (BottomBar) findViewById(R.id.bottomBar);
27         bottomBar.setOnTabSelectListener(new OnTabSelectListener() {
28             @Override
29             public void onTabSelected(@IdRes int tabId) {
30                 Fragment f;
31                 if (tabId == R.id.tab_home) {
32                     f = new MainFragment();
33                     getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
34 f).commit();
35                 }
36                 //Toolbar
37                 setUpToolbarWithTitle(getString(R.string.app_name));

```

```

35         getSupportActionBar().setDisplayHomeAsUpEnabled(false);
36         getSupportActionBar().setDisplayHomeAsUpEnabled(false);
37     }
38     if (tabId == R.id.tab_insight) {
39         f = new InsightFragment();
40
41         getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
42             f).commit();
43         //Toolbar
44         setUpToolBarWithTitle(getString(R.string.ui_insight));
45         getSupportActionBar().setDisplayHomeAsUpEnabled(false);
46         getSupportActionBar().setDisplayHomeAsUpEnabled(false);
47     }
48     if (tabId == R.id.tab_settings) {
49         f = new SettingsFragment();
50
51         getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
52             f).commit();
53         //Toolbar
54         setUpToolBarWithTitle(getString(R.string.ui_settings));
55         getSupportActionBar().setDisplayHomeAsUpEnabled(false);
56         getSupportActionBar().setDisplayHomeAsUpEnabled(false);
57     }
58     }
59     });
60     //Toolbar
61     setUpToolBarWithTitle(getString(R.string.app_name));
62     getSupportActionBar().setDisplayHomeAsUpEnabled(false);
63     getSupportActionBar().setDisplayHomeAsUpEnabled(false);
64 }
65 @Override
66 public void onBackPressed() {
67     super.onBackPressed();
68     finish();
69 }

```

Activity Fragment (Classe) Main:

```

1 package com.danilo.vendas.fragment;
2 import android.content.Intent;
3 import android.content.SharedPreferences;
4 import android.os.Bundle;
5 import android.support.v4.app.Fragment;
6 import android.support.v7.widget.CardView;
7 import android.view.LayoutInflater;
8 import android.view.View;
9 import android.view.ViewGroup;
10 import android.widget.TextView;
11 import android.widget.Toast;
12 import com.danilo.vendas.BaseActivity;

```



```

13 import com.danilo.vendas.R;
14 import com.danilo.vendas.ui.ClienteActivity;
15 import com.danilo.vendas.ui.ClienteListaActivity;
16 import com.danilo.vendas.ui.NovoClienteActivity;
17 import com.danilo.vendas.ui.PedidoActivity;
18 import com.danilo.vendas.ui.ProdutoActivity;
19 import java.util.Calendar;
20 import java.util.Date;
21 /**
22  * Fragment Principal, responsável pela composição da tela inicial
23  */
24 public class MainFragment extends Fragment implements View.OnClickListener {
25     SharedPreferences prefs;
26     private CardView cdParceiro;
27     private CardView cdPedido;
28     private CardView cdEnviarDados;
29     private CardView cdReceberDados;
30     private CardView cdNovaVenda;
31     private CardView cdNovoCliente;
32     private CardView cdProduto;
33     private TextView tvSaudacao;
34     public MainFragment() {
35         // Required empty public constructor
36     }
37     @Override
38     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
39         View rootView = inflater.inflate(R.layout.fragment_main, container, false);
40         //Restaura as preferencias gravadas
41         prefs = BaseActivity.getPreferences(getActivity());
42         //Recuperando referências dos componentes
43         cdParceiro = (CardView) rootView.findViewById(R.id.cdCliente);
44         cdPedido = (CardView) rootView.findViewById(R.id.cdVenda);
45         cdEnviarDados = (CardView) rootView.findViewById(R.id.cdEnviarDados);
46         cdReceberDados = (CardView) rootView.findViewById(R.id.cdReceberDados);
47         cdNovaVenda = (CardView) rootView.findViewById(R.id.cdNovaVenda);
48         cdNovoCliente = (CardView) rootView.findViewById(R.id.cdNovoCliente);
49         cdProduto = (CardView) rootView.findViewById(R.id.cdProduto);
50         tvSaudacao = (TextView) rootView.findViewById(R.id.tvSaudacao);
51         //Ouvintes de ação
52         cdParceiro.setOnClickListener(this);
53         cdPedido.setOnClickListener(this);
54         cdEnviarDados.setOnClickListener(this);
55         cdReceberDados.setOnClickListener(this);
56         cdNovaVenda.setOnClickListener(this);
57         cdNovoCliente.setOnClickListener(this);
58         cdProduto.setOnClickListener(this);
59         //Populando TextView Saudação
60         Calendar calendar = Calendar.getInstance();
61         Date date = calendar.getTime();

```

```
62     int hora = date.getHours();
63     if (hora >= 18 && hora < 24) {
64         tvSaudacao.setText("Boa Noite, Demo 1");
65     } else {
66         tvSaudacao.setText("Bom Dia, Demo 1");
67     }
68     return rootView;
69 }
70 @Override
71 public void onClick(View v) {
72     Intent intent;
73     switch (v.getId()) {
74         case R.id.cdCliente:
75             intent = new Intent(getActivity(), ClienteActivity.class);
76             startActivity(intent);
77             break;
78         case R.id.cdVenda:
79             intent = new Intent(getActivity(), PedidoActivity.class);
80             startActivity(intent);
81             break;
82         case R.id.cdEnviarDados:
83             Toast.makeText(getActivity(), "Enviando dados para o Servidor...!",
Toast.LENGTH_SHORT).show();
84             //Método responsável pela transmissão de dados
85             break;
86         case R.id.cdReceberDados:
87             Toast.makeText(getActivity(), "Recebendo dados do Servidor...!",
Toast.LENGTH_SHORT).show();
88             //Método responsável pelo recebimento
89             break;
90         case R.id.cdNovaVenda:
91             intent = new Intent(getActivity(), ClienteListaActivity.class);
92             startActivity(intent);
93             break;
94         case R.id.cdNovoCliente:
95             intent = new Intent(getActivity(), NovoClienteActivity.class);
96             startActivity(intent);
97             break;
98         case R.id.cdProduto:
99             intent = new Intent(getActivity(), ProdutoActivity.class);
100             startActivity(intent);
101             break;
102     }
103 }
```

Activity (Classe) Novo Pedido:

```
1 package com.danilo.vendas.ui;
2 import android.content.Intent;
3 import android.content.SharedPreferences;
4 import android.os.Bundle;
5 import android.support.design.widget.Snackbar;
6 import android.text.Editable;
7 import android.text.TextWatcher;
8 import android.view.Menu;
9 import android.view.MenuItem;
10 import android.view.View;
11 import android.widget.AdapterView;
12 import android.widget.Button;
13 import android.widget.EditText;
14 import android.widget.Spinner;
15 import android.widget.TextView;
16 import android.widget.Toast;
17 import com.danilo.vendas.dao.ItemPedidoDao;
18 import com.danilo.vendas.model.ItemPedido;
19 import com.danilo.vendas.model.Pedido;
20 import com.github.javiersantos.bottomdialogs.BottomDialog;
21 import com.danilo.vendas.BaseActivity;
22 import com.danilo.vendas.R;
23 import com.danilo.vendas.dao.PedidoDao;
24 import com.danilo.vendas.dao.ProdutoDao;
25 import com.danilo.vendas.dao.RentabilidadeDao;
26 import com.danilo.vendas.model.Produto;
27 import com.danilo.vendas.model.Rentabilidade;
28 import java.text.SimpleDateFormat;
29 import java.util.Calendar;
30 import java.util.Date;
31 /**
32  * Classe responsável pela manipulação de dados
33  */
34 public class NovoPedidoActivity extends BaseActivity implements View.OnClickListener
35 {
36     private Spinner spnUnidadePedido;
37     private EditText edtValorVenda;
38     private EditText edtQuantidade;
39     private TextView tvDescricaoProduto;
40     private TextView tvEstoque;
41     private TextView tvRentabilidadeItem;
42     private TextView tvValorTabela;
43     private TextView tvValorTotalPedido;
44     private TextView tvValorTotalItem;
45     private TextView tvQuantidadeTotal;
46     private Button btnAumentar;
47     private Button btnReduzir;
```

```

47 private Button btnFinalizar;
48 private Double quantidadeTotalItem = 0.0;
49 private Double quantidadeTotalPedido = 0.0;
50 private Double valorTotalPedido = 0.0;
51 private Double valorTotalItem = 0.0;
52 private Double valorCusto = 0.0;
53 SharedPreferences prefs;
54 private long lastBackPressedTime = 0;
55 @Override
56 protected void onCreate(Bundle savedInstanceState) {
57     enterFromBottomAnimation();
58     super.onCreate(savedInstanceState);
59     setContentView(R.layout.activity_novo_pedido);
60     //Restaura as preferencias gravadas
61     prefs = BaseActivity.getPreferences(this);
62     //Recuperando referências dos componentes
63     spnUnidadePedido = (Spinner) findViewById(R.id.spnUnidadePedido);
64     edtValorVenda = (EditText) findViewById(R.id.edtValorVenda);
65     edtQuantidade = (EditText) findViewById(R.id.edtQuantidade);
66     tvDescricaoProduto = (TextView) findViewById(R.id.tvDescricaoProduto);
67     tvEstoque = (TextView) findViewById(R.id.tvEstoque);
68     tvRentabilidadeItem = (TextView) findViewById(R.id.tvRentabilidadeItem);
69     tvValorTabela = (TextView) findViewById(R.id.tvValorTabela);
70     tvValorTotalPedido = (TextView) findViewById(R.id.tvValorTotalPedido);
71     tvValorTotalItem = (TextView) findViewById(R.id.tvValorTotalItem);
72     tvQuantidadeTotal = (TextView) findViewById(R.id.tvQuantidadeTotal);
73     btnAumentar = (Button) findViewById(R.id.btnAumentar);
74     btnReduzir = (Button) findViewById(R.id.btnReduzir);
75     btnFinalizar = (Button) findViewById(R.id.btnFinalizar);
76     //Toolbar
77     setUpToolbarWithTitle(getString(R.string.ui_nova_venda));
78     //Spinner
79     ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.array_unidade_pedido, android.R.layout.simple_spinner_item);
80
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
81     spnUnidadePedido.setAdapter(adapter);
82     //Ouvintes de ação
83     btnAumentar.setOnClickListener(this);
84     btnReduzir.setOnClickListener(this);
85     btnFinalizar.setOnClickListener(this);
86 }
87 @Override
88 public void onResume(){
89     super.onResume();
90     if(prefs.getInt("idVenda", 0) > 1) {
91         ItemPedidoDao itemPedidoDao = new ItemPedidoDao(getApplicationContext());
92         for (ItemPedido i : itemPedidoDao.getAll("SELECT
idItemVenda,SUM(quantidade) as quantidade,valorUnitario,SUM(valorTotal) as

```

```

valorTotal,idVenda,idProduto FROM itemVenda WHERE idVenda =
"+String.valueOf(prefs.getInt("idVenda", 0))+ " ") {
93     valorTotalPedido = i.getValorTotal();
94     quantidadeTotalPedido = i.getQuantidade();
95 }
96 }
97 //Pegando dados do produto selecionado
98 ProdutoDao dao = new ProdutoDao(getApplicationContext());
99 for (Produto p : dao.getAll("SELECT * FROM produto WHERE idProduto = " +
String.valueOf(prefs.getInt("idProduto", 0)) + " ")) {
100     //Definindo valor de venda
101     tvValorTabela.setText("R$ " + p.getValorTabela().toString() + "0");
102     edtValorVenda.setText(p.getValorTabela().toString());
103     tvDescricaoProduto.setText(p.getNomeProduto().toString());
104     valorTotalItem = p.getValorTabela();
105     quantidadeTotalItem = 1.0;
106     edtQuantidade.setText("1");
107     tvValorTotalItem.setText("R$ " + String.valueOf(valorTotalItem) + "0");
108     tvValorTotalPedido.setText("R$ " + String.valueOf(valorTotalPedido +
valorTotalItem) + "0");
109
tvQuantidadeTotal.setText(String.valueOf((int)(quantidadeTotalPedido+quantidadeTotalItem
))+ " CP");
110     tvEstoque.setText(p.getEstoque().toString() + "0 KG");
111 }
112 edtQuantidade.requestFocus();
113 RentabilidadeDao dao2 = new RentabilidadeDao(getApplicationContext());
114 for (Rentabilidade r : dao2.getAll("SELECT * FROM rentabilidade WHERE
idProduto = " + String.valueOf(prefs.getInt("idProduto", 0)) + " AND idCliente = " +
String.valueOf(prefs.getInt("idCliente", 0)) + "")) {
115     //Definindo valor de venda
116     valorCusto = r.getCusto();
117 }
118 if (valorTotalItem < valorCusto) {
119     tvRentabilidadeIem.setText("Pessíma");
120 } else {
121     tvRentabilidadeIem.setText("Ótima");
122 }
123 edtQuantidade.addTextChangedListener(new TextWatcher() {
124     @Override
125     public void beforeTextChanged(CharSequence s, int start, int count, int after) {
126     }
127     @Override
128     public void onTextChanged(CharSequence s, int start, int before, int count) {
129     }
130     @Override
131     public void afterTextChanged(Editable s) {
132         if (edtQuantidade.getText().toString().equals("")) {
133             edtQuantidade.setText("1");
134         }

```

```

135                                     quantidadeTotalItem =
Double.parseDouble(edtQuantidade.getText().toString());
136                                     valorTotalItem =
(Double.parseDouble(edtValorVenda.getText().toString().replace("R$ ", "")) *
Double.parseDouble(edtQuantidade.getText().toString()));
137         tvValorTotalItem.setText("R$ " + String.valueOf(valorTotalItem) + "0");
138         tvValorTotalPedido.setText("R$ " + String.valueOf(valorTotalPedido +
valorTotalItem) + "0");
139
tvQuantidadeTotal.setText(String.valueOf((int)(quantidadeTotalPedido+quantidadeTotalItem
))+" CP");
140     }
141 });
142     edtValorVenda.addTextChangedListener(new TextWatcher() {
143         @Override
144         public void beforeTextChanged(CharSequence s, int start, int count, int after) {
145         }
146         @Override
147         public void onTextChanged(CharSequence s, int start, int before, int count) {
148         }
149         @Override
150         public void afterTextChanged(Editable s) {
151             if (!edtValorVenda.getText().toString().isEmpty()) {
152                 if (Double.valueOf(edtValorVenda.getText().toString().replace("R$ ", "")) <
valorCusto) {
153                     tvRentabilidadeItem.setText("Pessíma");
154                 } else {
155                     tvRentabilidadeItem.setText("Ótima");
156                 }
157             }
158         }
159     });
160 }
161 public void salvaItem() {
162     ItemPedidoDao dao = new ItemPedidoDao(getApplicationContext());
163     ItemPedido vo = new ItemPedido();
164     int ultimoId = 0;
165     for (ItemPedido i : dao.getAll("SELECT * FROM itemVenda WHERE idItemVenda
= (SELECT MAX(idItemVenda) FROM itemVenda)")) {
166         ultimoId = i.getIdItemVenda() + 1;
167     }
168     vo.setIdItemVenda(ultimoId);
169     vo.setQuantidade(Double.parseDouble(edtQuantidade.getText().toString()));
170
vo.setValorUnitario(Double.parseDouble(edtValorVenda.getText().toString().replace("R$ ",
""))));
171
vo.setValorTotal(Double.parseDouble(edtQuantidade.getText().toString().replace("R$ ", ""))
* Double.parseDouble(edtValorVenda.getText().toString().replace("R$ ", "")));
172     vo.setIdVenda(prefs.getInt("idVenda", 0));

```

```

173     vo.setIdProduto(prefs.getInt("idProduto", 0));
174     try {
175         dao.insert(vo);
176         Toast.makeText(getApplicationContext(), "Item adicionado: " +
String.valueOf(prefs.getInt("idProduto", 0)), Toast.LENGTH_SHORT).show();
177     } catch (Exception e) {
178         Toast.makeText(getApplicationContext(), "Erro ao salvar: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
179     }
180 }
181 public void salvaBaseVenda(){
182     PedidoDao dao = new PedidoDao(getApplicationContext());
183     Pedido vo = new Pedido();
184     int ultimoId = 0;
185     for (Pedido p : dao.getAll("SELECT * FROM venda WHERE idVenda = (SELECT
MAX(idVenda) FROM venda)")) {
186         ultimoId = p.getIdVenda() + 1;
187     }
188     vo.setIdVenda(ultimoId);
189     vo.setIdCliente(prefs.getInt("idCliente", 1));
190     Calendar calendar = Calendar.getInstance();
191     Date date = calendar.getTime();
192     SimpleDateFormat format1 = new SimpleDateFormat("dd/MM/yyyy");
193     vo.setDataVenda(format1.format(date).toString());
194
vo.setValorTotal(Double.parseDouble(tvValorTotalPedido.getText().toString().replace("R$ ",
""))));
195     try {
196         dao.insert(vo);
197         //Salvando ultimo idVenda cadastrado
198         NovoPedidoActivity.this.prefs.edit().putInt("idVenda", ultimoId).apply();
199     } catch (Exception e) {
200         Toast.makeText(getApplicationContext(), "Erro ao salvar: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
201     }
202 }
203 public void finalizaVenda() {
204     PedidoDao dao = new PedidoDao(getApplicationContext());
205     Pedido vo = new Pedido();
206     vo.setIdVenda(prefs.getInt("idVenda", 0));
207     vo.setIdCliente(prefs.getInt("idCliente", 0));
208     Calendar calendar = Calendar.getInstance();
209     Date date = calendar.getTime();
210     SimpleDateFormat format1 = new SimpleDateFormat("dd/MM/yyyy");
211     vo.setDataVenda(format1.format(date).toString());
212
vo.setValorTotal(Double.parseDouble(tvValorTotalPedido.getText().toString().replace("R$ ",
""))));
213     try {
214         dao.update(vo);

```

```

215     //Limpa as preferencias
216     prefs.edit().clear().apply();
217 } catch (Exception e) {
218     Toast.makeText(getApplicationContext(), "Erro ao salvar: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
219 }
220 }
221 @Override
222 public boolean onCreateOptionsMenu(Menu menu) {
223     getMenuInflater().inflate(R.menu.menu_adicionar, menu);
224     return super.onCreateOptionsMenu(menu);
225 }
226 @Override
227 public boolean onOptionsItemSelected(MenuItem item) {
228     switch (item.getItemId()) {
229         case R.id.menu_adicionar:
230             if(prefs.getInt("idVenda", 0) == 0) {
231                 salvaBaseVenda();
232             }
233             salvaItem();
234             Intent intent = new Intent(NovoPedidoActivity.this, ProdutoActivity.class);
235             NovoPedidoActivity.this.startActivity(intent);
236             finish();
237             break;
238         }
239     return super.onOptionsItemSelected(item);
240 }
241 @Override
242 protected void onPause() {
243     exitToBottomAnimation();
244     super.onPause();
245 }
246 @Override
247 public void onClick(View v) {
248     switch (v.getId()) {
249         case R.id.btnAumentar:
250             //Populando quantidadeTotalItem
251             quantidadeTotalItem = Double.parseDouble(edtQuantidade.getText().toString())
+ 1;
252             edtQuantidade.setText(String.valueOf((int)(quantidadeTotalItem+0)));
253             valorTotalItem = (Double.parseDouble(edtValorVenda.getText().toString()) *
quantidadeTotalItem);
254             tvValorTotalItem.setText("R$ " + String.valueOf(valorTotalItem) + "0");
255             tvValorTotalPedido.setText("R$ " + String.valueOf(valorTotalPedido +
valorTotalItem) + "0");
256             tvQuantidadeTotal.setText(String.valueOf((int)(quantidadeTotalPedido+quantidadeTotalItem
))+" CP");
257             break;
258         case R.id.btnReduzir:

```



```

259         //Populando quantidadeTotalItem
260         if (quantidadeTotalItem > 1) {
261                                     quantidadeTotalItem =
Double.parseDouble(edtQuantidade.getText().toString()) - 1;
262             edtQuantidade.setText(String.valueOf((int)(quantidadeTotalItem+0)));
263             valorTotalItem = (Double.parseDouble(edtValorVenda.getText().toString()) *
quantidadeTotalItem);
264             tvValorTotalItem.setText("R$ " + String.valueOf(valorTotalItem) + "0");
265             tvValorTotalPedido.setText("R$ " + String.valueOf(valorTotalPedido +
valorTotalItem) + "0");
266             tvQuantidadeTotal.setText(String.valueOf((int)(quantidadeTotalPedido+quantidadeTotalItem
))+" CP");
267         }
268         break;
269     case R.id.btnFinalizar:
270         hideKeyboard(findViewById(android.R.id.content));
271         finalizaVenda();
272         new AlertDialog.Builder(this)
273             .setTitle("Pedido Finalizado!")
274             .setContent("Deseja realizar outro pedido?")
275             .setPositiveText("SIM")
276             .setNegativeText("NÃO")
277             .setIcon(R.drawable.ic_base)
278             .setCancelable(false)
279             .onPositive(new AlertDialog.ButtonCallback() {
280                 @Override
281                 public void onClick(DialogInterface dialog) {
282                     Intent intent = new Intent(getApplicationContext(),
ClienteListaActivity.class);
283                     startActivity(intent);
284                     finish();
285                 }
286             })
287             .onNegative(new AlertDialog.ButtonCallback() {
288                 @Override
289                 public void onClick(DialogInterface dialog) {
290                     Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
291                     startActivity(intent);
292                     finish();
293                 }
294             }).show();
295         break;
296     }
297 }
298 @Override
299 public void onBackPressed() {
300     if (lastBackPressedTime + 2000 > System.currentTimeMillis()) {
301         super.onBackPressed();

```

```

302     finish();
303     } else {
304         Snackbar bar = Snackbar.make(findViewById(android.R.id.content), "Pressione
novamente para sair! \nAtenção, os dados não serão salvos!", Snackbar.LENGTH_SHORT);
305         bar.setDuration(2000);
306         bar.show();
307         lastBackPressedTime = System.currentTimeMillis();
308     }
309 }

```

Activity (Classe) Base Activity:

```

1 package com.danilo.vendas;
2 import android.content.Context;
3 import android.content.SharedPreferences;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.support.v7.widget.Toolbar;
7 import android.view.View;
8 import android.view.WindowManager;
9 import android.view.inputmethod.InputMethodManager;
10 import com.danilo.vendas.R;
11 /**
12  * Classe responsável pela manipulação de dados
13  */
14 public class BaseActivity extends AppCompatActivity {
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         // Previne que o teclado apareça no início
19
20         getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_HI
DDEN);
21     }
22     protected void setUpToolbarWithTitle(String title){
23         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
24         setSupportActionBar(toolbar);
25         if(getSupportActionBar() != null) {
26             getSupportActionBar().setTitle(title);
27             getSupportActionBar().setDisplayHomeAsUpEnabled(true);
28             getSupportActionBar().setDisplayShowHomeEnabled(true);
29             toolbar.setNavigationOnClickListener(new View.OnClickListener() {
30                 @Override
31                 public void onClick(View v) {
32                     onBackPressed();
33                 }
34             });
35     }
36 }

```

```

35 }
36 protected void enterFromBottomAnimation(){
37     overridePendingTransition(R.anim.activity_open_translate_from_bottom,
R.anim.activity_no_animation);
38 }
39 protected void exitToBottomAnimation(){
40     overridePendingTransition(R.anim.activity_no_animation,
R.anim.activity_close_translate_to_bottom);
41 }
42 //Metodo para esconder o teclado
43 public void hideKeyboard(View view) {
44     InputMethodManager inputMethodManager
=(InputMethodManager) getSystemService(AppCompatActivity.INPUT_METHOD_SERVIC
E);
45     inputMethodManager.hideSoftInputFromWindow(view.getWindowToken(), 0);
46 }
47 //Método para obter as preferências salvas
48 public static SharedPreferences getPreferences(Context context) {
49     return context.getSharedPreferences("preferencias", Context.MODE_PRIVATE);
50 }

```

Manifest XML:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="com.danilo.vendas">
5     <!-- Permissões -->
6     <uses-permission android:name="android.permission.INTERNET" />
7     <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
8     <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
9     <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
10    <uses-permission android:name="android.permission.READ_SYNC_SETTINGS" />
11    <uses-permission android:name="android.permission.READ_SYNC_STATS" />
12    <uses-permission android:name="android.permission.VIBRATE" />
13    <application
14        android:allowBackup="true"
15        android:icon="@drawable/ic_base"
16        android:label="@string/app_name"
17        android:supportRtl="true"
18        android:theme="@style/AppTheme">
19        <!-- activities -->
20        <activity
21            android:name="com.danilo.vendas.ui.SplashActivity"
22            android:theme="@style/FullscreenTheme">
23            <intent-filter>
24                <action android:name="android.intent.action.MAIN" />
25                <category android:name="android.intent.category.LAUNCHER" />

```

```
26     </intent-filter>
27 </activity>
28 <activity
29     android:name="com.danilo.vendas.ui.LoginActivity"
30     android:configChanges="orientation"
31     android:theme="@style/AppThemeTranslucentStatus"
32     android:windowSoftInputMode="adjustResize" />
33 <activity
34     android:name="com.danilo.vendas.ui.ServidorActivity"
35     android:configChanges="orientation"
36     android:theme="@style/AppThemeTranslucentStatus"
37     android:windowSoftInputMode="adjustResize" />
38 <activity
39     android:name="com.danilo.vendas.ui.MainActivity"
40     android:configChanges="orientation"
41     android:label="@string/app_name"
42     android:windowSoftInputMode="adjustResize" />
43 <activity
44     android:name="com.danilo.vendas.ui.NovoPedidoActivity"
45     android:parentActivityName="com.danilo.vendas.ui.MainActivity"
46     android:screenOrientation="portrait"
47     android:windowSoftInputMode="adjustPan|adjustResize" />
48 <activity
49     android:name="com.danilo.vendas.ui.ClienteListaActivity"
50     android:parentActivityName="com.danilo.vendas.ui.MainActivity"
51     android:screenOrientation="portrait"
52     android:windowSoftInputMode="adjustPan|adjustResize" />
53 <activity
54     android:name="com.danilo.vendas.ui.ClienteActivity"
55     android:configChanges="orientation"
56     android:parentActivityName="com.danilo.vendas.ui.MainActivity"
57     android:windowSoftInputMode="adjustResize" />
58 <activity
59     android:name="com.danilo.vendas.ui.ProdutoActivity"
60     android:configChanges="orientation"
61     android:parentActivityName="com.danilo.vendas.ui.MainActivity"
62     android:windowSoftInputMode="adjustResize" />
63 <activity
64     android:name="com.danilo.vendas.ui.PedidoActivity"
65     android:configChanges="orientation"
66     android:parentActivityName="com.danilo.vendas.ui.MainActivity"
67     android:windowSoftInputMode="adjustResize" />
68 <activity
69     android:name="com.danilo.vendas.ui.ConsultaClienteActivity"
70     android:configChanges="orientation"
71     android:parentActivityName="com.danilo.vendas.ui.MainActivity"
72     android:windowSoftInputMode="adjustResize" />
73 <activity
74     android:name="com.danilo.vendas.ui.NovoClienteActivity"
75     android:configChanges="orientation"
```

```

76     android:parentActivityName="com.danilo.vendas.ui.MainActivity"
77     android:windowSoftInputMode="adjustResize" />
78 </application>

```

Layout Main:

```

1 <?xml version="1.0" encoding="utf-8"?>
2     <android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:fitsSystemWindows="true"
8     tools:context="com.danilo.vendas.ui.MainActivity">
9     <RelativeLayout
10        android:layout_width="match_parent"
11        android:layout_height="match_parent">
12        <include
13            android:id="@+id/toolbar"
14            layout="@layout/toolbar" />
15        <View
16            android:layout_width="match_parent"
17            android:layout_height="2dp"
18            android:layout_below="@id/toolbar"
19            android:background="@drawable/toolbar_dropshadow" />
20        <FrameLayout
21            android:id="@+id/fragment_container"
22            android:layout_width="match_parent"
23            android:layout_height="match_parent"
24            android:layout_below="@id/toolbar"
25            android:layout_marginBottom="60dp" />
26        <com.roughike.bottombar.BottomBar
27            android:id="@+id/bottomBar"
28            android:layout_width="match_parent"
29            android:layout_height="60dp"
30            android:layout_alignParentBottom="true"
31            app:bb_tabXmlResource="@xml/bottombar_tabs" />
32    </RelativeLayout>

```

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:card_view="http://schemas.android.com/apk/res-auto"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6     <ScrollView

```

```

7   android:layout_width="match_parent"
8   android:layout_height="match_parent"
9   android:layout_below="@id/toolbar">
10  <LinearLayout
11      android:layout_width="match_parent"
12      android:layout_height="match_parent"
13      android:orientation="vertical">
14      <TextView
15          android:id="@+id/tvSaudacao"
16          android:layout_width="match_parent"
17          android:layout_height="wrap_content"
18          android:layout_marginBottom="10dp"
19          android:layout_marginLeft="13dp"
20          android:layout_marginRight="13dp"
21          android:layout_marginTop="10dp"
22          android:text=""
23          android:textColor="@color/colorPrimary"
24          android:textSize="16dp"
25          android:textStyle="bold" />
26      <LinearLayout
27          android:layout_width="match_parent"
28          android:layout_height="match_parent"
29          android:orientation="vertical">
30          <LinearLayout
31              android:layout_width="match_parent"
32              android:layout_height="match_parent"
33              android:layout_marginLeft="4dp"
34              android:layout_marginRight="4dp"
35              android:orientation="horizontal">
36
37              <android.support.v7.widget.CardView
38                  xmlns:android="http://schemas.android.com/apk/res/android"
39                  android:id="@+id/cdNovaVenda"
40                  android:layout_width="wrap_content"
41                  android:layout_height="wrap_content"
42                  android:layout_margin="2dp"
43                  android:layout_marginLeft="1dp"
44                  android:layout_marginRight="1dp"
45                  android:clickable="true"
46                  android:elevation="80dp"
47                  android:foreground="?attr/selectableItemBackground"
48                  android:orientation="vertical">
49                  <LinearLayout
50                      android:layout_width="match_parent"
51                      android:layout_height="wrap_content"
52                      android:orientation="vertical"
53                      android:padding="2dp">
54                      <RelativeLayout
55                          android:layout_width="wrap_content"
56                          android:layout_height="wrap_content">
57                          <ImageView

```

```

56         android:layout_width="wrap_content"
57         android:layout_height="wrap_content"
58         android:src="@drawable/ic_card_base" />
59     <ImageView
60         android:layout_width="50dp"
61         android:layout_height="50dp"
62         android:layout_centerHorizontal="true"
63         android:layout_centerVertical="true"
64         android:elevation="5dp"
65         card_view:srcCompat="@drawable/ic_novo_pedido" />
66     </RelativeLayout>
67     <TextView
68         android:id="@+id/nameTextViewCliente"
69         android:layout_width="match_parent"
70         android:layout_height="25dp"
71         android:layout_marginBottom="4dp"
72         android:layout_marginTop="8dp"
73         android:gravity="center"
74         android:text="Novo Pedido"
75
android:textAppearance="@style/TextAppearance.AppCompat.Body1"
76         android:textStyle="bold" />
77     </LinearLayout>
78     </android.support.v7.widget.CardView>
79         <android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
80         android:id="@+id/cdCliente"
81         android:layout_width="wrap_content"
82         android:layout_height="wrap_content"
83         android:layout_margin="2dp"
84         android:layout_marginLeft="1dp"
85         android:layout_marginRight="1dp"
86         android:clickable="true"
87         android:elevation="80dp"
88         android:foreground="?attr/selectableItemBackground"
89         android:orientation="vertical">
90     <LinearLayout
91         android:layout_width="match_parent"
92         android:layout_height="wrap_content"
93         android:orientation="vertical"
94         android:padding="2dp">
95     <RelativeLayout
96         android:layout_width="wrap_content"
97         android:layout_height="wrap_content">
98     <ImageView
99         android:layout_width="wrap_content"
100        android:layout_height="wrap_content"
101        android:src="@drawable/ic_card_base" />
102    <ImageView
103        android:layout_width="50dp"

```

```

104         android:layout_height="50dp"
105         android:layout_centerHorizontal="true"
106         android:layout_centerVertical="true"
107         android:elevation="5dp"
108         card_view:srcCompat="@drawable/ic_cliente" />
109     </RelativeLayout>
110     <TextView
111         android:id="@+id/nameTextViewCliente"
112         android:layout_width="match_parent"
113         android:layout_height="25dp"
114         android:layout_marginBottom="4dp"
115         android:layout_marginTop="8dp"
116         android:gravity="center"
117         android:text="Clientes"
118
119         android:textAppearance="@style/TextAppearance.AppCompat.Body1"
120         android:textStyle="bold" />
121     </LinearLayout>
122     </android.support.v7.widget.CardView>
123     <android.support.v7.widget.CardView
124         xmlns:android="http://schemas.android.com/apk/res/android"
125         android:id="@+id/cdVenda"
126         android:layout_width="wrap_content"
127         android:layout_height="wrap_content"
128         android:layout_margin="2dp"
129         android:layout_marginLeft="1dp"
130         android:layout_marginRight="1dp"
131         android:clickable="true"
132         android:elevation="80dp"
133         android:foreground="?attr/selectableItemBackground"
134         android:orientation="vertical">
135         <LinearLayout
136             android:layout_width="match_parent"
137             android:layout_height="wrap_content"
138             android:orientation="vertical"
139             android:padding="2dp">
140             <RelativeLayout
141                 android:layout_width="wrap_content"
142                 android:layout_height="wrap_content">
143                 <ImageView
144                     android:layout_width="wrap_content"
145                     android:layout_height="wrap_content"
146                     android:src="@drawable/ic_card_base" />
147                 <ImageView
148                     android:layout_width="50dp"
149                     android:layout_height="50dp"
150                     android:layout_centerHorizontal="true"
151                     android:layout_centerVertical="true"
152                     android:elevation="5dp"
153                     card_view:srcCompat="@drawable/ic_pedido" />

```



```

200         android:elevation="5dp"
201         card_view:srcCompat="@drawable/ic_novo_cliente" />
202     </RelativeLayout>
203     <TextView
204         android:id="@+id/nameTextViewCliente"
205         android:layout_width="match_parent"
206         android:layout_height="25dp"
207         android:layout_marginBottom="4dp"
208         android:layout_marginTop="8dp"
209         android:gravity="center"
210         android:text="Novo Cliente"
211
212         android:textAppearance="@style/TextAppearance.AppCompat.Body1"
213         android:textStyle="bold" />
214     </LinearLayout>
215     </android.support.v7.widget.CardView>
216     <android.support.v7.widget.CardView
217         xmlns:android="http://schemas.android.com/apk/res/android"
218         android:id="@+id/cdProduto"
219         android:layout_width="wrap_content"
220         android:layout_height="wrap_content"
221         android:layout_margin="2dp"
222         android:layout_marginLeft="1dp"
223         android:layout_marginRight="1dp"
224         android:clickable="true"
225         android:elevation="80dp"
226         android:foreground="?attr/selectableItemBackground"
227         android:orientation="vertical">
228     <LinearLayout
229         android:layout_width="match_parent"
230         android:layout_height="wrap_content"
231         android:orientation="vertical"
232         android:padding="2dp">
233     <RelativeLayout
234         android:layout_width="wrap_content"
235         android:layout_height="wrap_content">
236     <ImageView
237         android:layout_width="wrap_content"
238         android:layout_height="wrap_content"
239         android:src="@drawable/ic_card_base" />
240     <ImageView
241         android:layout_width="50dp"
242         android:layout_height="50dp"
243         android:layout_centerHorizontal="true"
244         android:layout_centerVertical="true"
245         android:elevation="5dp"
246         card_view:srcCompat="@drawable/ic_produto" />
247     </RelativeLayout>
248     <TextView
249         android:id="@+id/nameTextViewCliente"

```

```

248         android:layout_width="match_parent"
249         android:layout_height="25dp"
250         android:layout_marginBottom="4dp"
251         android:layout_marginTop="8dp"
252         android:gravity="center"
253         android:text="Produtos"
254
android:textAppearance="@style/TextAppearance.AppCompat.Body1"
255         android:textStyle="bold" />
256     </LinearLayout>
257 </android.support.v7.widget.CardView>
258         <android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
259         android:id="@+id/cdReceberDados"
260         android:layout_width="wrap_content"
261         android:layout_height="wrap_content"
262         android:layout_margin="2dp"
263         android:layout_marginLeft="1dp"
264         android:layout_marginRight="1dp"
265         android:clickable="true"
266         android:elevation="80dp"
267         android:foreground="?attr/selectableItemBackground"
268         android:orientation="vertical">
269     <LinearLayout
270         android:layout_width="match_parent"
271         android:layout_height="wrap_content"
272         android:orientation="vertical"
273         android:padding="2dp">
274     <RelativeLayout
275         android:layout_width="wrap_content"
276         android:layout_height="wrap_content">
277     <ImageView
278         android:layout_width="wrap_content"
279         android:layout_height="wrap_content"
280         android:src="@drawable/ic_card_base" />
281     <ImageView
282         android:layout_width="50dp"
283         android:layout_height="50dp"
284         android:layout_centerHorizontal="true"
285         android:layout_centerVertical="true"
286         android:elevation="5dp"
287         card_view:srcCompat="@drawable/ic_receber_dados" />
288     </RelativeLayout>
289     <TextView
290         android:id="@+id/nameTextView"
291         android:layout_width="match_parent"
292         android:layout_height="25dp"
293         android:layout_marginBottom="4dp"
294         android:layout_marginTop="8dp"
295         android:gravity="center"

```

```

296         android:text="Receber Dados"
297
android:textAppearance="@style/TextAppearance.AppCompat.Body1"
298         android:textStyle="bold" />
299     </LinearLayout>
300 </android.support.v7.widget.CardView>
301 </LinearLayout>
302 <LinearLayout
303     android:layout_width="match_parent"
304     android:layout_height="match_parent"
305     android:layout_marginLeft="4dp"
306     android:layout_marginRight="4dp"
307     android:orientation="horizontal">
308                                     <android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
309         android:id="@+id/cdEnviarDados"
310         android:layout_width="wrap_content"
311         android:layout_height="wrap_content"
312         android:layout_margin="2dp"
313         android:layout_marginLeft="1dp"
314         android:layout_marginRight="1dp"
315         android:clickable="true"
316         android:elevation="80dp"
317         android:foreground="?attr/selectableItemBackground"
318         android:orientation="vertical">
319     <LinearLayout
320         android:layout_width="match_parent"
321         android:layout_height="wrap_content"
322         android:orientation="vertical"
323         android:padding="2dp">
324     <RelativeLayout
325         android:layout_width="wrap_content"
326         android:layout_height="wrap_content">
327     <ImageView
328         android:layout_width="wrap_content"
329         android:layout_height="wrap_content"
330         android:src="@drawable/ic_card_base" />
331     <ImageView
332         android:layout_width="50dp"
333         android:layout_height="50dp"
334         android:layout_centerHorizontal="true"
335         android:layout_centerVertical="true"
336         android:elevation="5dp"
337         card_view:srcCompat="@drawable/ic_enviar_dados" />
338 </RelativeLayout>
339 <TextView
340     android:id="@+id/nameTextView"
341     android:layout_width="match_parent"
342     android:layout_height="25dp"
343     android:layout_marginBottom="4dp"

```

```

344         android:layout_marginTop="8dp"
345         android:gravity="center"
346         android:text="Enviar Dados"
347
android:textAppearance="@style/TextAppearance.AppCompat.Body1"
348         android:textStyle="bold" />
349     </LinearLayout>
350 </android.support.v7.widget.CardView>
351 </LinearLayout>
352 </LinearLayout>
353 </LinearLayout>
354 </ScrollView>

```

Layout Novo Pedido:

```

1 <?xml version="1.0" encoding="utf-8"?>
2         <android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:fitsSystemWindows="true"
7     tools:context="com.danilo.vendas.ui.NovoPedidoActivity">
8     <RelativeLayout
9         android:layout_width="match_parent"
10        android:layout_height="match_parent">
11        <include
12            android:id="@+id/toolbar"
13            layout="@layout/toolbar" />
14        <View
15            android:layout_width="match_parent"
16            android:layout_height="2dp"
17            android:layout_below="@id/toolbar"
18            android:background="@drawable/toolbar_dropshadow" />
19        <ScrollView
20            android:layout_width="match_parent"
21            android:layout_height="match_parent"
22            android:layout_below="@id/toolbar">
23            <LinearLayout
24                android:layout_width="match_parent"
25                android:layout_height="wrap_content"
26                android:layout_marginLeft="16sp"
27                android:layout_marginRight="16sp"
28                android:orientation="vertical">
29                <LinearLayout
30                    android:layout_width="fill_parent"
31                    android:layout_height="wrap_content"
32                    android:layout_marginTop="20dp"
33                    android:orientation="horizontal"

```

```

34     android:weightSum="2">
35     <android.support.design.widget.TextInputLayout
36         android:layout_width="0dp"
37         android:layout_height="fill_parent"
38         android:layout_weight="1">
39         <TextView
40             android:layout_width="fill_parent"
41             android:layout_height="fill_parent"
42             android:fontFamily="sans-serif"
43             android:text="Rentabilidade Item"
44             android:textSize="@dimen/text_size_large" />
45     </android.support.design.widget.TextInputLayout>
46     <android.support.design.widget.TextInputLayout
47         android:layout_width="0dp"
48         android:layout_height="fill_parent"
49         android:layout_weight="1">
50         <TextView
51             android:id="@+id/tvRentabilidadeIem"
52             android:layout_width="fill_parent"
53             android:layout_height="fill_parent"
54             android:fontFamily="sans-serif"
55             android:text=""
56             android:textSize="@dimen/text_size_large" />
57     </android.support.design.widget.TextInputLayout>
58 </LinearLayout>
59 <LinearLayout
60     android:layout_width="fill_parent"
61     android:layout_height="wrap_content"
62     android:layout_marginTop="15dp"
63     android:orientation="horizontal"
64     android:weightSum="2">
65     <android.support.design.widget.TextInputLayout
66         android:layout_width="0dp"
67         android:layout_height="fill_parent"
68         android:layout_weight="1">
69         <TextView
70             android:layout_width="fill_parent"
71             android:layout_height="fill_parent"
72             android:fontFamily="sans-serif"
73             android:text="Estoque"
74             android:textSize="@dimen/text_size_large" />
75     </android.support.design.widget.TextInputLayout>
76     <android.support.design.widget.TextInputLayout
77         android:layout_width="0dp"
78         android:layout_height="fill_parent"
79         android:layout_weight="1">
80         <TextView
81             android:id="@+id/tvEstoque"
82             android:layout_width="fill_parent"
83             android:layout_height="fill_parent"

```

```

84         android:fontFamily="sans-serif"
85         android:text=""
86         android:textSize="@dimen/text_size_large" />
87     </android.support.design.widget.TextInputLayout>
88 </LinearLayout>
89 <LinearLayout
90     android:layout_width="fill_parent"
91     android:layout_height="wrap_content"
92     android:layout_marginTop="15dp"
93     android:orientation="horizontal"
94     android:weightSum="2">
95     <android.support.design.widget.TextInputLayout
96         android:layout_width="0dp"
97         android:layout_height="fill_parent"
98         android:layout_weight="1">
99         <TextView
100             android:layout_width="fill_parent"
101             android:layout_height="fill_parent"
102             android:fontFamily="sans-serif"
103             android:text="Valor Tabela"
104             android:textSize="@dimen/text_size_large" />
105     </android.support.design.widget.TextInputLayout>
106     <android.support.design.widget.TextInputLayout
107         android:layout_width="0dp"
108         android:layout_height="fill_parent"
109         android:layout_weight="1">
110         <TextView
111             android:id="@+id/tvValorTabela"
112             android:layout_width="fill_parent"
113             android:layout_height="fill_parent"
114             android:fontFamily="sans-serif"
115             android:text=""
116             android:textSize="@dimen/text_size_large" />
117     </android.support.design.widget.TextInputLayout>
118 </LinearLayout>
119 <LinearLayout
120     android:layout_width="match_parent"
121     android:layout_height="wrap_content"
122     android:layout_marginBottom="20sp"
123     android:layout_marginLeft="8dp"
124     android:layout_marginStart="8dp"
125     android:layout_marginTop="20dp"
126     android:orientation="vertical">
127     <RelativeLayout
128         android:layout_width="match_parent"
129         android:layout_height="wrap_content"
130         android:layout_marginTop="6dp"
131         android:clickable="true"
132         android:orientation="horizontal">
133         <LinearLayout

```

```

134         android:id="@+id/circle"
135         android:layout_width="28dp"
136         android:layout_height="28dp"
137         android:layout_alignParentLeft="true"
138         android:layout_alignParentStart="true"
139         android:layout_marginEnd="12dp"
140         android:layout_marginRight="12dp"
141         android:background="@drawable/oval_mail"
142         android:backgroundTint="@color/colorAccent"
143         android:orientation="vertical">
144         <TextView
145             android:layout_width="match_parent"
146             android:layout_height="match_parent"
147             android:fontFamily="sans-serif"
148             android:gravity="center"
149             android:text="1"
150             android:textColor="#ffffff"
151             android:textSize="16sp" />
152     </LinearLayout>
153     <TextView
154         android:id="@+id/tvDescricaoProduto"
155         android:layout_width="match_parent"
156         android:layout_height="28dp"
157         android:layout_toEndOf="@id/circle"
158         android:layout_toRightOf="@id/circle"
159         android:fontFamily="sans-serif"
160         android:gravity="center_vertical"
161         android:text="Produto"
162         android:textColor="#212121"
163         android:textSize="17sp" />
164 </RelativeLayout>
165 <LinearLayout
166     android:layout_width="match_parent"
167     android:layout_height="wrap_content"
168     android:layout_marginTop="12dp"
169     android:orientation="horizontal">
170     <LinearLayout
171         android:layout_width="4dp"
172         android:layout_height="match_parent"
173         android:layout_marginEnd="24dp"
174         android:layout_marginLeft="12dp"
175         android:layout_marginRight="24dp"
176         android:layout_marginStart="12dp"
177         android:background="#e0e0e0"
178         android:orientation="vertical" />
179     <LinearLayout
180         android:layout_width="match_parent"
181         android:layout_height="wrap_content"
182         android:orientation="vertical">
183     <LinearLayout

```



```

184     android:layout_width="match_parent"
185     android:layout_height="wrap_content"
186     android:orientation="vertical"
187     android:paddingTop="1dp">
188     <!-- Step content -->
189     <LinearLayout
190         android:layout_width="fill_parent"
191         android:layout_height="wrap_content"
192         android:layout_marginTop="15dp"
193         android:orientation="vertical">
194         <TextView
195             android:layout_width="match_parent"
196             android:layout_height="wrap_content"
197             android:layout_marginLeft="4sp"
198             android:text="Unidade"
199             android:textSize="14sp" />
200         <Spinner
201             android:id="@+id/spnUnidadePedido"
202             style="@style/Widget.AppCompat.Spinner.Underlined"
203             android:layout_width="match_parent"
204             android:layout_height="45sp"
205             android:spinnerMode="dialog" />
206     </LinearLayout>
207     <LinearLayout
208         android:layout_width="fill_parent"
209         android:layout_height="wrap_content"
210         android:layout_marginTop="15dp"
211         android:orientation="horizontal">
212         <android.support.design.widget.TextInputLayout
213             android:layout_width="0dp"
214             android:layout_height="wrap_content"
215             android:layout_weight="1">
216             <EditText
217                 android:id="@+id/edtValorVenda"
218                 android:layout_width="fill_parent"
219                 android:layout_height="wrap_content"
220                 android:hint="Valor Venda"
221                 android:inputType="textPersonName"
222                 android:singleLine="true"
223                 android:textColorHighlight="@color/colorPrimary"
224                 android:textIsSelectable="false"
225                 android:textSize="@dimen/text_size_activity" />
226             </android.support.design.widget.TextInputLayout>
227     </LinearLayout>
228     <LinearLayout
229         android:layout_width="fill_parent"
230         android:layout_height="wrap_content"
231         android:layout_marginTop="15dp"
232         android:orientation="horizontal">
233         <android.support.design.widget.TextInputLayout

```

```

234         android:layout_width="0dp"
235         android:layout_height="wrap_content"
236         android:layout_weight="1">
237         <EditText
238             android:id="@+id/edtQuantidade"
239             android:layout_width="match_parent"
240             android:layout_height="wrap_content"
241             android:hint="Quantidade"
242             android:inputType="numberDecimal"
243             android:singleLine="true"
244             android:textColorHighlight="@color/colorPrimary"
245             android:textIsSelectable="false"
246             android:textSize="@dimen/text_size_large" />
247     </android.support.design.widget.TextInputLayout>
248     <android.support.v7.widget.AppCompatButton
249         android:id="@+id/btnReduzir"
250         android:layout_width="wrap_content"
251         android:layout_height="wrap_content"
252         android:layout_marginEnd="8dp"
253         android:layout_marginRight="8dp"
254         android:text="-"
255         android:textColor="@color/white_pressed"
256         android:textSize="17sp"
257         android:textStyle="bold" />
258     <android.support.v7.widget.AppCompatButton
259         android:id="@+id/btnAumentar"
260         android:layout_width="wrap_content"
261         android:layout_height="wrap_content"
262         android:layout_marginEnd="8dp"
263         android:layout_marginRight="8dp"
264         android:text="+"
265         android:textColor="@color/white_pressed"
266         android:textSize="17sp"
267         android:textStyle="bold" />
268     </LinearLayout>
269 </LinearLayout>
270 </LinearLayout>
271 </LinearLayout>
272 </LinearLayout>
273 <LinearLayout
274     android:layout_width="fill_parent"
275     android:layout_height="wrap_content"
276     android:layout_marginTop="20dp"
277     android:orientation="horizontal"
278     android:weightSum="2">
279     <android.support.design.widget.TextInputLayout
280         android:layout_width="0dp"
281         android:layout_height="fill_parent"
282         android:layout_weight="1">
283         <TextView

```

```

284         android:layout_width="fill_parent"
285         android:layout_height="fill_parent"
286         android:fontFamily="sans-serif"
287         android:text="Valor Total Item"
288         android:textSize="@dimen/text_size_large" />
289     </android.support.design.widget.TextInputLayout>
290     <android.support.design.widget.TextInputLayout
291         android:layout_width="0dp"
292         android:layout_height="fill_parent"
293         android:layout_weight="1">
294         <TextView
295             android:id="@+id/tvValorTotalItem"
296             android:layout_width="fill_parent"
297             android:layout_height="fill_parent"
298             android:fontFamily="sans-serif"
299             android:text=""
300             android:textSize="@dimen/text_size_large" />
301     </android.support.design.widget.TextInputLayout>
302 </LinearLayout>
303 <LinearLayout
304     android:layout_width="fill_parent"
305     android:layout_height="wrap_content"
306     android:layout_marginTop="20dp"
307     android:orientation="horizontal"
308     android:weightSum="2">
309     <android.support.design.widget.TextInputLayout
310         android:layout_width="0dp"
311         android:layout_height="fill_parent"
312         android:layout_weight="1">
313         <TextView
314             android:layout_width="fill_parent"
315             android:layout_height="fill_parent"
316             android:fontFamily="sans-serif"
317             android:text="Quantidade Total"
318             android:textSize="@dimen/text_size_large" />
319     </android.support.design.widget.TextInputLayout>
320     <android.support.design.widget.TextInputLayout
321         android:layout_width="0dp"
322         android:layout_height="fill_parent"
323         android:layout_weight="1">
324         <TextView
325             android:id="@+id/tvQuantidadeTotal"
326             android:layout_width="fill_parent"
327             android:layout_height="fill_parent"
328             android:fontFamily="sans-serif"
329             android:text=""
330             android:textSize="@dimen/text_size_large" />
331     </android.support.design.widget.TextInputLayout>
332 </LinearLayout>
333 </LinearLayout>

```

```

334 </ScrollView>
335 <android.support.design.widget.BottomNavigationView
336     android:id="@+id/navigation"
337     android:layout_width="match_parent"
338     android:layout_height="wrap_content"
339     android:layout_alignParentBottom="true"
340     android:layout_gravity="bottom"
341     android:background="@color/colorAccent"
342     android:elevation="10dp">
343     <LinearLayout
344         android:layout_width="fill_parent"
345         android:layout_height="wrap_content"
346         android:layout_marginBottom="10dp"
347         android:layout_marginLeft="15dp"
348         android:layout_marginTop="10dp"
349         android:orientation="vertical">
350         <LinearLayout
351             android:layout_width="fill_parent"
352             android:layout_height="wrap_content"
353             android:layout_marginTop="10dp"
354             android:orientation="horizontal"
355             android:weightSum="2">
356             <android.support.design.widget.TextInputLayout
357                 android:layout_width="0dp"
358                 android:layout_height="fill_parent"
359                 android:layout_weight="1">
360                 <TextView
361                     android:layout_width="fill_parent"
362                     android:layout_height="fill_parent"
363                     android:text="Valor Total Pedido"
364                     android:textColor="@color/colorPrimary"
365                     android:textSize="@dimen/text_size_large"
366                     android:textStyle="bold" />
367             </android.support.design.widget.TextInputLayout>
368             <android.support.design.widget.TextInputLayout
369                 android:layout_width="0dp"
370                 android:layout_height="fill_parent"
371                 android:layout_weight="1">
372                 <TextView
373                     android:id="@+id/tvValorTotalPedido"
374                     android:layout_width="fill_parent"
375                     android:layout_height="fill_parent"
376                     android:text=""
377                     android:textColor="@color/colorPrimary"
378                     android:textSize="@dimen/text_size_large"
379                     android:textStyle="bold" />
380             </android.support.design.widget.TextInputLayout>
381             <android.support.v7.widget.AppCompatButton
382                 android:id="@+id/btnFinalizar"
383                 android:layout_width="wrap_content"

```

```

384         android:layout_height="25dp"
385         android:layout_weight="1"
386         android:background="@color/colorAccent"
387         android:text="FINALIZAR"
388         android:textColor="@color/colorPrimary"
389         android:textStyle="bold" />
390     </LinearLayout>
391 </LinearLayout>
392 </android.support.design.widget.BottomNavigationView>
393 <View
394     android:layout_width="match_parent"
395     android:layout_height="1dp"
396     android:layout_above="@id/navigation"
397     android:background="@drawable/toolbar_dropshadow" />
398 </RelativeLayout>

```

Build.gradle Aplicação:

```

1 apply plugin: 'com.android.application'
2 apply plugin: 'android-apt'
3 android {
4     compileSdkVersion 25
5     buildToolsVersion "26.0.2"
6     defaultConfig {
7         applicationId 'com.danilo.vendas'
8         minSdkVersion 16
9         targetSdkVersion 25
10        versionCode 1
11        versionName '1.0.0'
12        multiDexEnabled true
13        vectorDrawables.useSupportLibrary = true
14    }
15    repositories {
16        mavenCentral()
17        maven {
18            url "https://jitpack.io"
19        }
20        maven {
21            url "https://oss.sonatype.org/content/repositories/snapshots"
22        }
23    }
24    buildTypes {
25        release {
26            minifyEnabled true
27            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
28        }
29    }
30    lintOptions {

```

```
31     abortOnError false
32   }
33   dexOptions {
34     incremental = true
35     preDexLibraries = false
36     javaMaxHeapSize "4g"
37   }
38   productFlavors {
39   }
40}
41dependencies {
42   compile fileTree(include: ['*.jar'], dir: 'libs')
43   compile 'com.android.support:appcompat-v7:25.3.1'
44   compile 'com.android.support:design:25.3.1'
45   compile 'com.android.support:support-v4:25.3.1'
46   compile 'com.android.support:cardview-v7:25.3.1'
47   compile 'com.android.support:recyclerview-v7:25.3.1'
48   compile 'com.android.support:support-vector-drawable:25.3.1'
49   compile 'com.github.javiersantos:BottomDialogs:1.1'
50   compile 'com.afollestad.material-dialogs:core:0.8.6.1'
51   compile 'com.github.coyarzun89:fabtransitionactivity:0.2.0'
52   compile 'com.roughike:bottom-bar:2.3.1'
53
54   compile 'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.client:4.1.2'
55   testCompile 'junit:junit:4.12'
56   apt 'com.jakewharton:butterknife-compiler:8.2.1'
```