

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luana Monteiro Vieira

**Desenvolvimento de sistemas de gestão para
plataformas móveis apoiado em conceitos de
engenharia de software e teorias sobre a
interação humano-computador: um estudo de
caso prático**

Uberlândia, Brasil

2015

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luana Monteiro Vieira

**Desenvolvimento de sistemas de gestão para plataformas
móveis apoiado em conceitos de engenharia de software
e teorias sobre a interação humano-computador: um
estudo de caso prático**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Ciência da Computação.

Orientador: Professor Fabiano Azevedo Dorça

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2015

Luana Monteiro Vieira

Desenvolvimento de sistemas de gestão para plataformas móveis apoiado em conceitos de engenharia de software e teorias sobre a interação humano-computador: um estudo de caso prático

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 18 de fevereiro de 2015:

Professor Fabiano Azevedo Dorça
Orientador

Professor

Professor

Uberlândia, Brasil
2015

Agradecimentos

Agradeço primeiramente a Deus, por ter me fortalecido em todos os momentos da minha vida até aqui.

Agradeço aos meus pais por terem sido não só um referencial de persistência para mim como também me apoiado em todas as minhas desejos e anseios.

Agradeço aos amigos e familiares por também terem me apoiado e acreditado em mim.

Agradeço ao meu príncipe, meu companheiro, amigo, confidente fiel, meu namorado que sempre esteve me apoiando e me ajudando.

Ainda que eu falasse as línguas dos homens e dos anjos, e não tivesse amor, seria como o metal que soa ou como o sino que tine. E ainda que tivesse o dom de profecia, e conhecesse todos os mistérios e toda a ciência, e ainda que tivesse toda a fé, de maneira tal que transportasse os montes, e não tivesse amor, nada seria. E ainda que distribuísse toda a minha fortuna para sustento dos pobres, e ainda que entregasse o meu corpo para ser queimado, e não tivesse amor, nada disso me aproveitaria. ICoríntios 13:1-3

Resumo

Este trabalho consiste de um estudo de caso prático para aplicação de técnicas de Engenharia de Software, programação para dispositivos móveis e técnicas de Interação Humano-Computador e tem por objetivo mostrar resultados plausíveis da aplicação dessas teorias e técnicas, além disso gerar um aplicativo funcional que, futuramente, poderá ser amplamente usado pelo público alvo: consultoras de cosméticos.

Pretende-se mostrar a efetividade destas técnicas implantando, no cotidiano, uma aplicação de fluxo de caixa. Esta, por sua vez, construída seguindo o padrão de projeto chamado Processo Unificado, bem como a estruturação em camadas. Sempre dando ênfase também na interação do usuário com o sistema, ou seja gerando uma aplicação de fácil uso, estruturada e documentada.

O trabalho foi concluído com êxito, com algumas dificuldades que serão especificadas na conclusão, contudo pode-se observar a eficiência e boa estruturação proveniente das técnicas e teorias aplicadas, resultando em um estudo de caso prático, além de uma aplicação promissora para o mercado real.

Palavras-chave: Engenharia de Software, Interação Humano-Computador, Modelagem de Software, Android

Lista de ilustrações

Figura 1 – Tabela de atividades do método de avaliação heurística, retirada da referência (BARBOSA, 2010), pag. 318	26
Figura 2 – Diagrama UML: Módulos do sistema	31
Figura 3 – Diagrama de Modelo: Módulos do sistema	36
Figura 4 – Diagrama DER: Projeto Meu Estoque Rosa (Completo)	37
Figura 5 – Diagrama DER: Projeto Meu Estoque Rosa (Agenda x Cliente)	38
Figura 6 – Diagrama DER: Projeto Meu Estoque Rosa (Financeiro x Cliente) . . .	39
Figura 7 – Diagrama DER: Projeto Meu Estoque Rosa (Financeiro x Produto) . .	39
Figura 8 – Diagrama MER: Projeto Meu Estoque Rosa	40
Figura 9 – Diagrama de Atividades para o caso de uso de cadastro de pedido e itens do pedido.	43
Figura 10 – Diagrama de Sequência para o caso de uso de cadastro de pedido e itens do pedido.	44
Figura 11 – Protótipo Hi-Fi : Primeira tela principal criada para a aplicação Meu Estoque Rosa, levando em consideração as características do usuário final.	45
Figura 12 – Protótipo Hi-Fi : Tela principal.	46
Figura 13 – Protótipo Hi-Fi :Tela de cadastro de clientes.	47
Figura 14 – Tela principal final.	50
Figura 15 – Tela financeira final.	51
Figura 16 – Tela de cadastro de sessões final.	52
Figura 17 – Screenshot tirada do framework eclipse configurado com o ambiente de desenvolvimento Android, destacando a pasta de layout que é a camada de visão no Android	54
Figura 18 – Diagrama de Atividades para o caso de uso de cadastro de clientes, categorias, sessões e compromissos.	96
Figura 19 – Diagrama de Atividades para o caso de uso de alteração e exclusão de clientes, categorias, sessões e compromissos.	97
Figura 20 – Diagrama de Atividades para o caso de uso de alteração e exclusão de pedido e itens do pedido.	98
Figura 21 – Diagrama de Sequência para o caso de uso cadastro de clientes.	99
Figura 22 – Diagrama de Sequência para o caso de uso cadastro de categorias. . .	100
Figura 23 – Diagrama de Sequência para o caso de uso cadastro de subcategorias .	101
Figura 24 – Diagrama de Sequência para o caso de uso cadastro de produtos. . . .	102
Figura 25 – Diagrama de Sequência para o caso de uso cadastro de sessões.	103
Figura 26 – Diagrama de Sequência para o caso de uso cadastro de compromissos.	104

Figura 27 – Diagrama de Sequência para o caso de uso cadastro de metas.	105
Figura 28 – Diagrama de Sequência para o caso de uso cadastro de pedidos e itens do pedido.	106
Figura 29 – Diagrama de Sequência para o caso de uso exclusão de itens(Obs: o único caso em que não a exclusão é o de estoque , uma vez que o estoque é excluído junto com o produto).	107
Figura 30 – Protótipo Hi-Fi : Tela de cliente.	109
Figura 31 – Protótipo Hi-Fi : Tela de cadastro de clientes mostrando os campos obrigatórios.	110
Figura 32 – Protótipo Hi-Fi : Tela de cadastro de clientes mostrando os campos preenchidos sem a data.	111
Figura 33 – Protótipo Hi-Fi : Tela de cadastro de clientes mostrando os campos preenchidos com a data no formato adequado.	112
Figura 34 – Protótipo Hi-Fi : Mensagem de sucesso para o cadastro de clientes. . .	113
Figura 35 – Protótipo Hi-Fi : Tela de pesquisa de clientes.	114
Figura 36 – Protótipo Hi-Fi : Tela de pesquisa de clientes, filtro de pesquisa. . . .	115
Figura 37 – Protótipo Hi-Fi : Tela de calendário do módulo agenda, seleção de um dia aleatório	116

Lista de tabelas

Tabela 1 – Modelo de Caso de Uso	18
Tabela 2 – Exemplo de avaliação heurística.	26
Tabela 3 – Questionário com resultados obtidos.	48
Tabela 4 – Tabela de notas e significado.	48
Tabela 5 – Avaliação Heurística por módulos	49
Tabela 6 – Caso de Teste: Cadastro de Pedidos	58
Tabela 7 – Caso de Teste: Cadastro de Itens do Pedido	58
Tabela 8 – Questionário de avaliação final da aplicação desenvolvida: Meu Estoque Rosa	60
Tabela 9 – Caso de Teste: Cadastro de Clientes	127
Tabela 10 – Caso de Teste: Cadastro de Categorias	127
Tabela 11 – Caso de Teste: Cadastro de Subcategorias	128
Tabela 12 – Caso de Teste: Cadastro de Produtos	128
Tabela 13 – Caso de Teste: Cadastro de Sessões	129
Tabela 14 – Caso de Teste: Cadastro de Compromissos	129
Tabela 15 – Caso de Teste: Cadastro de Metas	130
Tabela 16 – Caso de Teste: Cadastro de Pedidos e Itens do pedido	130
Tabela 17 – Caso de Teste: Adicionar ao Estoque	131
Tabela 18 – Caso de Teste: Pesquisar Clientes	131
Tabela 19 – Caso de Teste: Pesquisar Categorias	131
Tabela 20 – Caso de Teste: Pesquisar Subcategorias	132
Tabela 21 – Caso de Teste: Pesquisar Produtos	132
Tabela 22 – Caso de Teste: Visualizar Calendário	132
Tabela 23 – Caso de Teste: Pesquisar Compromissos	133
Tabela 24 – Caso de Teste: Pesquisar Metas	133
Tabela 25 – Caso de Teste: Resumo Financeiro	133
Tabela 26 – Caso de Teste: Pesquisar Estoque	134
Tabela 27 – Caso de Teste: Pesquisar Produtos em Falta	134
Tabela 28 – Caso de Teste: Pesquisar Sessões	135
Tabela 29 – Caso de Teste: Pesquisar Pedidos	135
Tabela 30 – Caso de Teste: Pesquisar Itens do Pedido	135
Tabela 31 – Caso de Teste: Alteração de Clientes	136
Tabela 32 – Caso de Teste: Alteração de Categorias	136
Tabela 33 – Caso de Teste: Alteração de Subcategorias	137
Tabela 34 – Caso de Teste: Alteração de Produtos	138
Tabela 35 – Caso de Teste: Alteração de Sessões	138

Tabela 36 – Caso de Teste: Alteração de Compromissos	139
Tabela 37 – Caso de Teste: Alteração de Pedidos	139
Tabela 38 – Caso de Teste: Alteração de Itens do Pedido	140
Tabela 39 – Caso de Teste: Alteração de Metas	140
Tabela 40 – Caso de Teste:Excluir Metas	141
Tabela 41 – Caso de Teste:Excluir Clientes	141
Tabela 42 – Caso de Teste:Excluir Categorias	141
Tabela 43 – Caso de Teste:Excluir Subcategorias	141
Tabela 44 – Caso de Teste:Excluir Produtos	142
Tabela 45 – Caso de Teste:Excluir Sessões	142
Tabela 46 – Caso de Teste:Excluir Compromissos	142
Tabela 47 – Caso de Teste:Excluir Pedidos	142
Tabela 48 – Caso de Teste:Excluir Itens do pedido	143

Lista de abreviaturas e siglas

IHC	Iteração Humano Computador
PU	Processo Unificado
UML	Linguagem de Modelagem Unificada
MVC	<i>Model, View, Control</i>
DAO	<i>Data Access Object</i>
POJO	<i>Plain Old Java Objects</i>
UI	<i>User Interface</i>

Sumário

1	INTRODUÇÃO	13
	Introdução	13
1.1	Sobre a proposta	13
1.2	Objetivos	13
1.3	Contexto do domínio	14
1.3.1	Módulos	14
1.4	Justificativa	15
1.5	Descrição dos Capítulos	15
2	REFERENCIAL TEÓRICO	16
2.1	Processo Unificado e Características	16
2.1.1	Concepção	16
2.1.1.1	Artefatos da Fase de Concepção	17
2.1.1.2	O Modelo de Casos de Uso	17
2.1.2	Elaboração	18
2.1.2.1	Artefatos da Elaboração	19
2.1.2.2	Iterações	19
2.1.2.3	Modelo de Domínio no projeto	20
2.1.2.4	Diagramação e Arquitetura do Projeto	20
2.1.3	Construção	21
2.1.4	Transição	21
2.1.4.1	Testes	22
2.2	Interação Humano Computador	22
2.2.1	Conceitos	23
2.2.2	Avaliação Heurística	24
2.2.3	Prototipação	26
2.3	Android	27
2.4	Padrão MVC + DAO	27
3	TRABALHOS CORRELATOS	29
4	METODOLOGIA	31
4.1	Concepção	31
4.2	Elaboração	32
4.3	Construção	32

4.4	Transição	34
5	DESENVOLVIMENTO	35
5.1	Concepção	35
5.2	Elaboração	36
5.3	Construção	52
5.4	Transição	57
6	CONCLUSÃO	61
	Referências	63
	APÊNDICES	65
	APÊNDICE A – REGRAS DE NÉGOCIO	66
	APÊNDICE B – DOCUMENTO DE ARQUITETURA DO SOFTWARE	68
B.1	Cliente	68
B.2	Produtos	68
B.3	Agenda	69
B.4	Financeiro	70
	APÊNDICE C – MODELO DE CASOS DE USO	72
C.1	Módulo Cliente	72
C.2	Módulo Produto	75
C.3	Módulo Agenda	85
C.4	Módulo Financeiro	88
	APÊNDICE D – MODELO DO PROJETO	95
D.1	Diagramas de Atividades	95
D.2	Diagramas de sequência	95
	APÊNDICE E – PROTÓTIPO HI-FI	108
	APÊNDICE F – EXEMPLOS DE CÓDIGOS-FONTE EM JAVA	117
	APÊNDICE G – DESCRIÇÃO DOS CASOS DE TESTE UNITÁRIOS	127

1 Introdução

1.1 Sobre a proposta

O ramo de desenvolvimento de software tem crescido muito durante os anos e as inovações tem ganhado espaço no mercado e um ramo que tem sido cada vez mais visado pelas empresas é o de desenvolvimento para plataformas móveis. As aplicações, por sua vez, são produzidas seguindo padrões de projeto e estruturas próprias.

Muitas destas aplicações tem sido criadas seguindo rigorosos padrões de qualidade e com estruturas bem definidas, já outras nem tanto. Portanto visando um mercado de público específico, acredita-se que é possível criar um aplicativo seguindo um padrão de projeto específico, com ênfase na sua estrutura e *design*.

Logo, tendo em vista o mercado de cosméticos, este trabalho tem como objetivo aplicar um padrão de projeto ao desenvolvimento de uma aplicação para a plataforma Android. A ideia é produzir um produto que seja útil tanto no aspecto de Interação Humano-Computador (IHC) como na sua base estrutural, visando a programação em camadas.

O produto final deve ser uma aplicação capaz de gerenciar o fluxo de caixa de uma consultora de cosméticos, feito em uma estrutura de camadas bem definidas, aliando conceitos de Engenharia de Software e Padrões de Projeto, bem como IHC.

A seguir serão especificados os objetivos deste trabalho e como serão abordados os pontos mais importantes deste estudo de caso.

1.2 Objetivos

Os objetivos deste projeto são:

- desenvolver um sistema de software para gerenciamento do fluxo de caixa de uma revendedora, que contenha módulos bem definidos e coesos;
- usar técnicas de engenharia de software para moldar sua estrutura e implementá-la;
- usar alguns conceitos de IHC para adequar a interface e componentes ao ramo de negócio específico.

Este software contém os seguintes módulos: Financeiro, Produtos, Agenda e Cliente. Estes módulos estão conectados uns aos outros.

- implantar um banco de dados local, sendo acessível em qualquer lugar pelo dispositivo móvel, no caso usando a plataforma Android.
- aplicar todo o conhecimento adquirido no curso relevante ao projeto, tais técnicas e padrões de projeto da Engenharia de Software, *design* de interfaces da IHC, modelagem de dados e requisitos da Modelagem de Software e a codificação e práticas da disciplina de Computação Móvel.
- após a conclusão do projeto, implantar o aplicativo em um ambiente real para que seja testado e avaliado, assim podendo ser aprimorado de acordo com as avaliações finais dos usuários.

1.3 Contexto do domínio

Este projeto contempla um nicho de mercado específico e está baseado em uma das várias marcas de cosméticos que existem, a marca Mary Kay.

O público-alvo são as consultoras e consultores da marca, ou seja, as pessoas que fazem o trabalho de revenda de produtos e também o trabalho de instruir o cliente sobre os produtos, e como usá-los.

Os produtos da marca são divididos em categorias, algumas destas sendo subdivididas em subcategorias, assim sendo uma das regras de negócio abordadas neste projeto. Isto é para facilitar a organização dos produtos.

As pessoas que trabalham com a marca também fazem sessões com os clientes, ou seja, vão a casa de cada cliente, no mínimo uma vez, para demonstrar alguns produtos e instruir o cliente sobre o uso correto dos mesmos. O cliente pode chamar outras pessoas para participar destas sessões, assim ajudando na divulgação da marca e obtenção de novos clientes para a consultora.

1.3.1 Módulos

Os módulos serão essenciais para o sistema, cada um deles apresenta características que o diferenciam:

- **Módulo Financeiro:** Tem como objetivo gerenciar toda transação financeira relacionada ao estoque, como gerenciar entrada e saída de produtos, controle de trocas entre consultoras, e geração de relatórios diários, mensais, trimestrais e o resumo anual.
- **Módulo Produtos:** O objetivo deste módulo é gerenciar todas as operações relacionadas aos produtos, no caso podendo ser remoção, atualização e cadastro de itens,

sempre indicando ao usuário o resultado da operação feita. Este módulo é importante para o controle do cadastro de todos os produtos, bem como suas categorias e subcategorias.

- **Módulo Agenda:** O objetivo deste módulo é ter todos os compromissos e sessões dispostos em um calendário, para que seja possível visualizar quais compromissos estarão marcados em cada dia e também visualizar a frequência de compromissos no mês. Este módulo gerencia o cadastro de compromissos, ou seja, entregas, reuniões, eventos e etc, e de sessões, está fortemente relacionado com o módulo Cliente.
- **Módulo Cliente:** Neste módulo são tratadas todas as questões de cadastro de clientes, e atualização de informações de acordo com a necessidade pessoal de cada consultora.

1.4 Justificativa

Com a popularização do uso de dispositivos móveis baseados na plataforma Android, ficou mais fácil desenvolver aplicações móveis e testá-las. Este software foi desenvolvido visando uma melhora para o negócio de revenda de cosméticos. A ideia principal é ajudar no dia a dia de uma consultora, usando um aplicativo que forneça as ferramentas necessárias para controlar toda entrada e saída de produtos, como também seus compromissos e o agendamento de visitas aos clientes, que é o ponto forte da revenda para este nicho de mercado.

Por isso desenvolveu-se este software com módulos bem divididos e interface amigável, para que o uso do mesmo fosse fácil e rápido agilizando todo processo referente ao trabalho.

1.5 Descrição dos Capítulos

No próximo capítulo serão tratados os trabalhos relacionados; no Capítulo 4, será explicada a metodologia do projeto; no Capítulo 5, será abordado o desenvolvimento do projeto bem como os seus métodos e testes; por fim uma conclusão do trabalho é feita no capítulo 6.

2 Referencial Teórico

2.1 Processo Unificado e Características

O Processo Unificado ou PU é um processo de desenvolvimento de software baseado na seguinte premissa: que deve-se desenvolver um software de forma iterativa e evolutiva, deixando a parte de documentação para ser escrita durante todo o projeto e focando em desenvolver o software como um todo junto de sua documentação (LARMAN, 2007).

Para a realização do PU, é necessário seguir alguns procedimentos e utilizar algumas ferramentas específicas. As fases de um PU e os seus instrumentos serão detalhados nas próximas subseções.

Um projeto baseado no PU tem como característica principal ser iterativo e evolutivo, isso porque o PU é desenvolvido por meio de iterações programadas de pouca duração, em cada iteração se faz a análise, modelagem e implementação além dos testes (LARMAN, 2007). Cada iteração é organizada em quatro fases principais:

- **Concepção:** oferece uma visão de alto nível, dos casos de negócio, escopo do problema e estimativas.
- **Elaboração:** é uma visão mais detalhada dos requisitos, junto com uma implementação da arquitetura de modo iterativo, bem como escopo e estimativas mais bem definidas, tendo em vista uma análise mais detalhada dos elementos de risco.
- **Construção:** nesta fase os requisitos considerados de menor risco são escalados para implementação de forma iterativa, bem como o projeto é preparado para ser implantado.
- **Transição:** são realizados os testes das edições e a implantação do projeto.

2.1.1 Concepção

A Fase da concepção como o próprio nome diz trata a concepção do produto, bem como o seu escopo e ideia principal, tem por objetivo gerar uma visão de alto nível do projeto e impulsioná-lo, como também decidir se é ou não viável, de acordo com as regras de negócio estabelecidas para o projeto.

Essa fase possui algumas características e instrumentos essenciais, demonstrados a seguir:

2.1.1.1 Artefatos da Fase de Concepção

Uma característica do PU é o uso de artefatos para organização dos requisitos, baseando no livro (LARMAN, 2007) pode-se citar os artefatos que mais serão utilizados nesse projeto:

- **Modelo de Casos de Uso:** Um conjunto de Casos de Uso descritos, podendo ser de alto nível no começo do projeto e mais baixo nível no final, representando os principais requisitos funcionais ou não-funcionais.
- **Glossário:** Define as características relativas aos dados e atributos, bem como dicionário de dados.
- **Visão:** Contém a visão geral do projeto em alto nível, agrupando os casos de uso tratados no modelo e nas regras de negócio tratadas no projeto.
- **Regras de Negócio:** Contém todas as regras de negócio, requisitos relacionados como a política do projeto e características governamentais, especificação de gerenciamento e leis da empresa ou qual linha de projeto precisa seguir.
- **Plano de Iteração:** Planejamento da primeira iteração.

Com isso planeja-se nesta fase da concepção a abordagem destes artefatos, e seus requisitos, fazendo um planejamento de iterações para as próximas fases do PU.

2.1.1.2 O Modelo de Casos de Uso

Em um projeto PU é importante ressaltar que os requisitos funcionais do projeto são representados e caracterizados em um modelo de Casos de Uso, sempre lembrando que não se deve fazer a documentação total dos casos previamente, sendo então diferente de um projeto em cascata. Logo, nesta seção será descrito como cada caso de uso será abordado e implementado, sendo então um documento textual com alguns diagramas.

Como dito os casos de uso podem ser funcionais ou não-funcionais, ou seja, podem definir funções que serão implementadas no projeto, ou podem definir características que o projeto deve conter. Logo para esta fase do projeto, são tratados de acordo com o decorrer das iterações, todos os casos de uso importantes.

Através de um modelo de casos de uso, pode-se definir como serão abordados os mesmos e com qual prioridade, tendo devida atenção com os casos de uso que ao decorrer do processo são adicionados.

Opcionalmente pode-se usar também um diagrama de contexto para definir estes casos, seus atores, nomes, atributos e relacionamentos.

A seguir têm-se um exemplo da caso de uso para este projeto, baseado na proposta do PU, os casos de uso mais importante devem ser descritos em forma de relatório. É através dessa breve documentação que se tem uma visão da estrutura do sistema.

O exemplo a seguir é uma representação de como podem ser documentados os casos de uso, de acordo com (LARMAN, 2007), e será utilizado como uma base para a produção de uma breve documentação dos principais casos de uso.

Seção do Caso de Uso	Comentário
Nome do Caso de Uso	Começar com um verbo.
Escopo	O sistema em projeto.
Nível	"Objetivo do usuário" ou "sub-função".
Ator principal	Chama o sistema para fornecer os serviços
Interessados e interesses	Quem se importa com esse caso de uso e o que eles desejam ?
Pré-condições	O que precisa ser verdade de início e precisa ser dito ao leitor.
Garantia de sucesso	O que precisa ser verdade e quando da finalização bem sucedida e se vale a pena dizer ao leitor.
Cenário de sucesso principal	Um caminho típico, incondicional e otimista do cenário de sucesso.
Extensões	Cenários alternativos de sucesso ou fracasso.
Requisitos especiais	Requisitos não funcionais relacionados.
Listas de variantes tecnológicas e de dados	Métodos de entrada e saída e formatos de dados variáveis.
Frequência de ocorrência	Influencia a investigação, teste e oportunidade da implementação.
Diversos	Como, por exemplo, pontos em aberto.

Tabela 1 – Modelo de Caso de Uso

É importante que nessa fase seja decidido se o projeto é viável ou não, quais recursos serão utilizados para tal feito, como também o orçamento previsto para a conclusão do projeto.

2.1.2 Elaboração

Nesta fase tem-se uma visão melhorada dos requisitos, estabelecendo uma arquitetura para tratá-los, assim depois da fase de concepção tem-se o início da implementação. A etapa da elaboração não trata todos os requisitos, alguns deles são incluídos a medida em que novas iterações são iniciadas.

Com os casos de uso mais importantes declarados em alto nível, inicia-se o trabalho sobre as iterações de acordo com os casos de uso, sempre associando a cada iteração apenas alguns casos de uso e os resultados obtidos na iteração anterior. É uma boa prática de

projeto PU que essas iterações sejam curtas, durando apenas algumas semanas, e que sejam tratados os requisitos de maior importância e risco na fase de elaboração.

O resultado dessas iterações são partes já funcionando do projeto todo, e estas partes são testadas para buscar falhas arquiteturais e novos requisitos importantes que não foram tratados ainda.

2.1.2.1 Artefatos da Elaboração

Nesta fase também podem inserir artefatos para melhor elaboração da arquitetura geral do projeto. Baseado na proposta de trabalho, alguns artefatos essenciais são tratados:

- **Modelo de domínio:** Compreende os requisitos de domínio do sistema e oferece uma visão dos conceitos mais importantes relacionados. Oferecerá uma visão do domínio do problema, e conceitos essenciais para entendimento e construção do projeto.
- **Modelo de Projeto:** Modelo do projeto, contém os diagramas referentes a ele dando a visão lógica e arquitetural.
- **Documento de Arquitetura de Software:** Documento que resume todos os requisitos chaves e problemas, para auxiliar na construção da arquitetura e na solução do projeto em si.
- **Modelo de banco de Dados:** Possui as especificações dos dados e o banco que será usado, e mapeamento das relações.
- **Modelo de teste:** Modelo dos caso de teste com suas especificação de como serão aplicados.

2.1.2.2 Iterações

Para as iterações foi necessário basear-se nas primeiras noções do projeto apresentadas na concepção, sempre prestando atenção na visão principal do projeto, guardando as mudanças feitas e os novos casos de uso encontrados, para futuras iterações. Seguindo então para os outros módulos, até todos terem sido tratados e elaborados, para então construir os detalhes finais na fase de construção.

Cada iteração na fase de elaboração teve como objetivo retornar uma parte do projeto já pronta, mesmo que seja uma parte mínima, nesta fase inicia-se a implementação. Como boa prática PU a implementação e testes tiveram início o mais cedo possível, sempre documentando todo o processo na busca por novos requisitos e detalhes não vistos antes(LARMAN, 2007).

Para cada iteração é preciso três itens importantes, de acordo com o livro (LARMAN, 2007) é importante estar a par dos riscos, tanto como de complexidade no projeto implementado como também em mudanças de negócio ou regras. Deve-se prestar atenção na cobertura, ou seja, nas primeiras iterações, deve se tratar no mínimo uma parte de todos os módulos do sistema. Também deve se atentar a criticidade, ou seja, deve-se considerar os requisitos de mais alto valor para o negócio.

Logo há uma necessidade de escalar de acordo com a prioridade de implementação os casos de uso, definindo os de alta, média e baixa prioridade. A cada iteração esta lista é atualizada até o término do projeto. Portanto, deve-se na concepção obter-se uma lista de casos de uso e suas classificações, esta lista deve ser usada aqui como base para a sequência de implementação.

2.1.2.3 Modelo de Domínio no projeto

O modelo de domínio é um passo importante Orientado a Objetos que estabelece os conceitos existentes, dando uma visão de alto nível do projeto como um todo, podendo ser representado por um diagrama de classes (LARMAN, 2007). Este diagrama representará as classes conceituais, atributos e relações existentes entre elas.

Algumas definições são essenciais para o modelo de domínio, como já dito ele trata de conceitos e das classes conceituais, que são uma ideia e como o nome diz um conceito. Também uma classe conceitual pode ser definida de acordo com (LARMAN, 2007) pelo seu símbolo, intenção e extensão. Os símbolos são palavras ou imagens aplicadas as classes, intenção remete a definição do objetivo da classe e extensão aos exemplos relacionados a classe.

Então é importante fazer uma análise do projeto para encontrar as classes conceituais e moldar um esquema destas classes com suas relações, usando o diagrama de classes para definir a estrutura e os atributos e o diagrama de sequência, explicados a seguir.

2.1.2.4 Diagramação e Arquitetura do Projeto

Na fase da elaboração é comum o uso de diagramas para representar as relações existentes tanto nos módulos como seus atributos, relações entre requisitos e casos de uso. Contudo não serão representadas todas as relações em diagramas, uma boa prática PU é colocar em diagramas apenas o mais importante, e o que deve se ter mais atenção na implementação.

Lembrando que o PU é iterativo (LARMAN, 2007), estes diagramas serão produzidos a medida que o projeto estiver em andamento, sendo assim no final teremos uma documentação que poderá ajudar em atualizações futuras ou até mesmo em manutenções. Portanto no PU o foco é na documentação e implementação construídas juntas desde o

começo, sendo que no final temos um sistema completo e documentado.

Assim um diagrama importante tratado primeiro foi o diagrama de sequência. O diagrama de sequência é usado para representar a sequência de mensagens mandadas entre as entidades de uma determinada classe, ou de um caso de uso. Nele são demonstradas as entradas e saídas dos objetos na relação representada. Nesse diagrama representou-se também os atores externos do sistema e então a representação das chamadas de eventos e respostas dadas à eles.

Outro diagrama importante é o diagrama de classes, que também faz a representação mas das relações entre classes e a representações dos atributos e métodos. Este diagrama define a arquitetura do sistema, e pode ser usado também para representar domínio conceitual, e é usado para se ter uma visão estática dos objetos e seus atributos, como também associações e relações.

2.1.3 Construção

Nesta fase tem-se toda uma arquitetura montada a partir dos principais requisitos tratados na elaboração, logo é nesta fase que ocorre a construção do projeto em si, é também que trata-se os detalhes do projeto, bem como requisitos de baixo risco. Portanto esta é a fase mais longa do projeto PU, contando com mais iterações.

O número de iterações é maior devido ao volume de código a ser programado, mas sempre de acordo com o objetivo básico do PU, de ser iterativo e evolutivo, a cada iteração desta fase obtém-se documentos gerados nela e uma parte maior do projeto já concluída.

Em cada iteração, além de serem tratados e testados os casos de uso, também é feito o planejamento da próxima iteração, de acordo com os resultados obtidos. Em todas as iterações há novos testes e novos atributos ou casos de uso, contudo é na fase de elaboração que se encontram quase todos os requisitos e atributos a serem inseridos no projeto.

2.1.4 Transição

A fase de transição é dita como a fase de testes e manutenções, é nela que é feita a implantação do projeto pronto e testam-se todos os componentes. Contudo, os testes são feitos durante todo o projeto, e por isso, define-se que os testes feitos nesta fase são os testes finais.

Nesta fase implanta-se e avalia-se o software junto ao cliente final para detecção de erros e *bugs*, sendo então uma fase de poucas iterações e mais de testes de competência do software.

2.1.4.1 Testes

Testes são muito importantes para a avaliação da eficácia e eficiência de um projeto, são eles que determinam quando um componente ou unidade está funcionando de maneira correta e atendendo corretamente os requisitos.

Serão definidos aqui os testes que serão usados no projeto, na construção e finalização do trabalho, serão usados testes para validar e aprimorar cada vez mais os componentes e unidades criados, como também o sistema como um todo.

Pode-se definir então os tipos de testes a serem usados neste projeto seguindo a ideia proposta pelo autor Pressman em seu livro ([FILHO, 2009](#))

- **Teste de aceitação:** Teste feito em um sistema a fim de determinar se é aceitável ou não, pode se capacitar um cliente ou usuário para aplicá-lo.
- **Teste de desenvolvimento:** Teste realizado durante o desenvolvimento, podendo ser formal ou informal e realizado pelo desenvolvedor em um componente, ou no sistema como um todo.
- **Teste de unitário:** Teste feito para cada de uso específico do projeto, geralmente seguindo as descrições dos casos de teste.
- **Teste de caixa-branca:** Teste realizado no sistema interno e componentes contido nele.
- **Teste de caixa-cinza:** Este teste é aplicado nos componente mas focando nas saídas obtidas de acordo com as entradas e especificações, sem levar em conta o interior.
- **Teste de caixa-preta:** Este teste é aplicado no sistema mas focando nas saídas obtidas de acordo com as entradas e especificações, sem levar em conta o interior.

2.2 Interação Humano Computador

A disciplina Interação Humano Computador (IHC) tem como objetivo fazer a ligação entre o computador e o usuário de forma mais usual possível, ou seja, criar interfaces e dispositivos mais acessíveis e de fácil entendimento para o usuário, diminuindo a dificuldade de uso do software bem como sua compreensão. ([BARBOSA, 2010](#))

De acordo com essa disciplina o foco é melhorar o entendimento dos programas inserindo ferramentas e mudanças pertinentes, aumentando a facilidade para manipulação dos mesmos, para isso existem alguns conceitos que devem ser compreendidos ([BARBOSA, 2010](#)).

2.2.1 Conceitos

Existem conceitos que são muito importantes na disciplina de IHC, a seguir explicaremos alguns.

- **Interação:** A interação trata a relação do usuário com o computador, sendo assim é dita como um processo de comunicação, manipulação e troca de informações. Existem diferentes perspectivas de interação, e a que será abordada neste trabalho será a interação na perspectiva de ferramenta.

A interação, na perspectiva de ferramenta, define o diálogo entre usuário e sistema como ferramenta de auxílio à tarefas do usuário, e como uma ferramenta deve ter as seguintes qualidades : a facilidade de uso e de boas funcionalidades para o negócio tratado([BARBOSA, 2010](#)).

- **Interface:** A interface consiste em todo componente de software ou hardware que faz contato com o usuário tanto recebendo estímulos do usuário, como oferecendo recursos e mostrando informações ao usuário. Por isso deve ser de fácil uso e desenvolvido de acordo com o perfil do usuário alvo da aplicação.

Portanto, a interface deve estar de acordo com os padrões e especificações do perfil do usuário, além disso deve estar de acordo com as regras de negócio envolvidas no projeto.

- **Affordance:** Conjunto de propriedades de software e hardware visíveis ao usuário, às operações que podem ser realizadas com o sistema, bem como às formas de manipulá-lo. Assim a *affordance* de interface diz como o usuário pode usar o sistema e o que o sistema pode oferecer a ele.

Por exemplo, a *affordance* de um botão pode ser a capacidade dele de ser pressionável, ou também de ser atribuído a uma ação de pesquisa, alteração, entre outras.

- **Usabilidade :** Usabilidade e experiência do usuário são critérios de qualidade essenciais para a IHC. A usabilidade é definida pela norma ([ISO, 1991](#)) como:

"Um conjunto de atributos relacionados com o esforço necessário para o uso de um sistema interativo, e relacionados com a avaliação individual de tal uso, por um conjunto específico de usuários."

Outra definição vemos na norma ([ISO, 1998](#)) :

"O grau em que um produto é usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico."

De acordo com a norma, eficácia se restringe à capacidade de uso adequado do sistema, eficiência quanto aos recursos exigidos para interação com o sistema de

forma correta e o grau de satisfação se refere ao quanto o sistema tem beneficiado seus usuários.

De acordo com (BARBOSA, 2010) os fatores que influenciam na usabilidade são : facilidade de aprendizado, facilidade de recordação, eficiência, segurança de uso e satisfação do usuário.

- **Acessibilidade:** Esse fator está associado à capacidade de o usuário usar e usufruir do sistema sem qualquer dificuldade associada. De acordo com o cenário onde o sistema é implantado a acessibilidade é um fator essencial, uma vez que a interação com o usuário deve ser feita levando em conta suas necessidades e dificuldades.
- **Comunicabilidade:** Está relacionada à capacidade de comunicação entre o usuário e a interface, é preciso ter estratégias bem definidas para que isso ocorra. De acordo com (BARBOSA, 2010) se o usuário consegue entender as funcionalidades da interface, ele pode ter mais chances de usá-la de forma efetiva.

Para obter um grau maior de comunicabilidade o desenvolvedor da interface pode fazer uso de analogias ao construir suas interfaces, uma vez que a analogia ajuda o usuário a estabelecer hipóteses sobre o funcionamento da interface, mediante experiências anteriores com outros sistemas.

2.2.2 Avaliação Heurística

A avaliação heurística consiste em uma avaliação de *design* levando como base um conjunto de heurísticas, proposta por Jakob Nielsen e Rolf Molich na década de 1990, estas por sua vez tentam de maneira eficaz encontrar problemas de usabilidade no *design* (BARBOSA, 2010).

Sua proposta é de ser rápida e de baixo custo sendo feita pelos próprios desenvolvedores usando um conjunto de diretrizes(heurísticas), estas são o resultado da análise de mais de 240 problemas encontrados em outros protótipos, por Nielsen em 1993 (BARBOSA, 2010) .

A seguir, são apresentadas as 10 heurísticas propostas por Nielsen e suas características (BARBOSA, 2010):

- **Visibilidade do estado do sistema:** o sistema deve manter o usuário a par de toda ação ou mudança no sistema, apresentando sempre notificações.
- **Correspondência entre o sistema e o mundo real:** devem ser usadas palavras e informações de acordo com a linguagem real usada pelo usuário, sem jargões ou termos técnicos usados pelos desenvolvedores, oferecendo um ambiente adequado.

- **Controle e liberdade do usuário:** um sistema com boa usabilidade deve possuir maneiras de desfazer ações possibilitando o maior controle possível do usuário na aplicação.
- **Consistência e padronização:** o sistema deve manter o padrão em todas as funções e componentes, evitando confusão do usuário por conta de algum desentendimento.
- **Reconhecimento em vez de memorização:** o sistema deve ter seu *design* projetado para que o usuário não precise se lembrar de informações e também reconheça sem precisar de ajuda o que cada componente faz.
- **Flexibilidade e eficiência de uso:** o sistema deve conter componentes que flexibilizam o serviço, não sendo perceptível tanto para novatos como para usuários experientes.
- **Projeto estético e minimalista:** a interface do sistema deve conter o essencial previsto para sua função específica, sendo assim essencial que não contenha muitas informações irrelevantes ou desnecessárias.
- **Prevenção de erros:** o sistema deve mostrar mensagens de erro e assegurar que não ocorram problemas por conta de inserção, seleção ou edição de dados.
- **Ajuda aos usuários no reconhecimento, diagnóstico e recuperação de erros:** na ocorrência de um erro, a mensagem exibida deve ser concisa e mostrar a correção do erro, sendo expressa de maneira simples e objetiva.
- **Ajuda e documentação:** é essencial que o sistema tenha opções de ajuda para o usuário, com passos e demonstrações, deve também ter uma documentação para eventuais consultas e dúvidas. (BARBOSA, 2010)

A partir destas diretrizes faz-se a avaliação do *design* da aplicação gerando um documento de retorno com as informações sobre supostas discordâncias, severidade e como proceder para adequação do *design*. Esta avaliação é feita de acordo com atividades que são explicadas na Tabela 1. A severidade citada acima pode seguir a seguinte classificação também proposta por Nielsen(1994a) :

1: *problema cosmético* - um problema que não precisa ser consertado caso o projeto esteja com atrasos.

2: *problema pequeno* - precisa ser consertado, contudo não tem prioridade alta.

3: *problema grande* - é importante que seja consertado e lhe é atribuído alta prioridade, problemas grandes tendem a afetar o bom funcionamento do programa causando falhas de usabilidade.

4: *problema catastrófico* - é de máxima prioridade e deve ser consertado o quanto antes, este pode danificar o funcionamento da aplicação e gerar erros graves no momento que o usuário estiver usando.

avaliação heurística	
atividade	tarefa
Preparação	<i>Todos os avaliadores:</i> ▪ aprendem sobre a situação atual: usuários, domínio etc. ▪ selecionam as partes da interface que devem ser avaliadas
Coleta de dados	<i>Cada avaliador, individualmente:</i> ▪ inspeciona a interface para identificar violações das heurísticas ▪ lista os problemas encontrados pela inspeção, indicando local, gravidade, justificativa e recomendações de solução
Interpretação	
Consolidação dos resultados	<i>Todos os avaliadores:</i> ▪ revisam os problemas encontrados, julgando sua relevância, gravidade, justificativa e recomendações de solução ▪ geram um relatório consolidado
Relato dos resultados	

Figura 1 – Tabela de atividades do método de avaliação heurística, retirada da referência (BARBOSA, 2010), pag. 318

Assim pela proposta de Nielsen a esta avaliação segue a forma mostrada na Tabela (2)

Problema	Descreve o problema encontrado protótipo em avaliação
Heurística violada	Mostra qual das heurísticas esta sendo violada
Explicação	Diz o porque da violação da heurística
Gravidade	Nível da gravidade, proposta por Nielsen que vai de 0 a 4, bem como a descrição do que acontece se o problema persistir
Possível Solução	Propõe uma solução para o problema

Tabela 2 – Exemplo de avaliação heurística.

2.2.3 Prototipação

A prototipação é uma conceito importante no desenvolvimento da interface gráfica e na arquitetura de um sistema, uma vez que oferece um meio de avaliação antes da entrega do mesmo. O resultado da avaliação possibilita a adequação do sistema às necessidades do cliente e a possibilidade de melhorias durante o projeto. A seguir são apresentadas as características de cada tipo de protótipo e suas vertentes.

- *Low-Fidelity (Lo-Fi)*: Prototipação de baixo nível, ou baixa fidelidade, e sem muitos detalhes técnicos, geralmente usada no início do projeto pelos desenvolvedores. Dentro desse nível temos vários tipos de protótipos, entre os mais usados estão os desenhos em papel das interfaces gráficas, com diferentes abordagens - que não serão aprofundadas.

Assim pode-se ter uma visão de como poderia ser a interface gráfica e seus componentes a partir de um protótipo no papel.(PREECE YVONNE ROGERS, 2005)

- *High-Fidelity (Hi-Fi)*: Prototipação de alto nível, ou alta fidelidade, e com muitos detalhes técnicos, este nível de prototipação é garantido após a implementação de boa parte do projeto, sendo assim um protótipo mais fiel e eficiente do sistema, já contendo algumas funções finais. Este geralmente é para avaliação com o cliente.

Geralmente os protótipos Hi-Fi são uma prévia do sistema com algumas funções prontas ou apenas a interface com os componentes, podendo ser feito em uma ferramenta específica ou sendo o próprio sistema mas sem todas as funções. (PREECE YVONNE ROGERS, 2005)

2.3 Android

A tecnologia Android foi criada pela empresa *Open Handset Alliance*, baseada no núcleo do Linux e desenvolvida para ser um sistema operacional para plataforma mobile de código aberto. Esta tecnologia foi comprada pela Google e continuou sendo aperfeiçoada e melhorada com o tempo. (WIKIPÉDIA, 2015a)

Hoje a Android está em sua versão 5.0 e apelidado de *Lollipop*, um dos seus antecessores o *Ice Cream Sandwich* de versão 4.0, deverá ser a base de programação para este projeto. (WIKIPÉDIA, 2015b)

Para todos os celulares baseados na tecnologia Android, são disponibilizadas as atualizações de sistema, sendo que uma nova atualização não interfere na anterior em quesito de programas e adaptações, logo uma aplicação criada para a versão 4.0 funcionará da mesma forma para as versões posteriores.

2.4 Padrão MVC + DAO

O padrão *Model-View-Control* (MVC) é um padrão de arquitetura em que são utilizadas as camadas, visão, controle e modelo, pode ser aplicado em qualquer projeto e aliado a camada *Data Access Object* (DAO) têm-se um forte acoplamento (FILHO, 2009). A seguir, serão apresentadas as características de cada camada.

- **Modelo:** Camada responsável pelo modelo de dados, bem como pelas regras de negócio envolvidas no sistema, tem comunicação com o controle e a visão sendo que é a camada central usada para passagem de dados.
- **Controle:** Camada responsável pelo controle das funções tanto da camada de visão, controlando as solicitações do usuário, como também das atribuições usando a camada modelo.

- **Visão:** Camada visível ao usuário, composta pelas interfaces gráficas do sistema, faz comunicação com o controle no momento da requisição de uma função pelo usuário e usa a camada de modelo para passagem de dados.

Para auxiliar a persistência dos dados é comum o uso da camada DAO, que nada mais é que a camada de banco de dados e classes de persistência dos dados. Esta camada é uma extensão da camada de modelo, não deixando de ser essencial a projetos grandes que necessitem de uma boa base de dados, ou seja, que mantenha a persistência de dados e a segurança, não afetando a eficiência da aplicação (SOUZA, 2014).

Como se trata uma aplicação para plataforma Android, essas camadas são tratadas com nomes e formas diferentes, não distanciando das suas características. Assim a camada de modelo é garantida pela camada POJO (Plain Old Java Objects) que nada mais é que uma camada de objetos Java simplificada - por se tratar de um sistema desenvolvido na linguagem nativa do Android.

Para fazer a separação da persistência com o modelo foi adotada a camada DAO, mesmo que essa seja parte da camada de modelo. Assim, usando o banco de dados SQLite que também é nativo do Android.

A camada de Controle e Visão é esta na camada UI (User Interface), no Android a visão está garantida pelos arquivos de extensão .xml que sempre estarão em um pacote separado criado pelo próprio *framework* do Android e dentro da pasta nomeada de *layout*, estas fazem a comunicação com o usuário, todo o controle destes arquivos é garantido por arquivos com extensão .java contidos na camada UI da aplicação, assim estabelecendo o padrão MVC (DEVELOPERS, 2015).

3 Trabalhos Correlatos

Na busca de soluções parecidas com a proposta deste trabalho, foram encontrados muitos aplicativos de controle de fluxo de caixa. Contudo, esses aplicativos são direcionados para o controle mais pessoal e sem muito envolvimento com algum tipo de negócio.

O mais parecido com a ideia deste trabalho é o aplicativo Controle de Vendas ([PACHECO, 2014](#)), que foi desenvolvido pelo desenvolvedor Thiago Pacheco e disponibilizado para download na [textitWeb](#). Este aplicativo foi criado com o intuito de auxiliar vendedores autônomos, fazendo o gerenciamento do fluxo de vendas e de forma simples o cadastro e gerenciamento de finanças, clientes e produtos. Neste aplicativo é também possível fazer relatórios e backup dos dados, contudo não foi especificada uma regra de negócio fixa, muito menos houve um tratamento específico de clientes e compromissos. Também não há evidência de Agenda, que é um ponto chave em nosso projeto.

O aplicativo Mint.com Personal Finance ([INTUIT, 2014](#)) é um software também para Android que já ganhou vários prêmios, e é voltado para o gerenciamento de finanças pessoais, reconhecido no mercado exterior, tem boas ferramentas para indicação de dívidas, saldo financeiro, balanço geral de gastos, monitoramento de finanças e gráficos para visualização de gastos no mês. Mas nosso foco será no controle de estoque e caixa do negócio em questão, logo esse aplicativo será um bom referencial, mesmo tendo sido concebido para área pessoal.

Outro aplicativo que tem como proposta gerenciar o fluxo de caixa pessoal é o Fluxo de Caixa ([DANTAS, 2014](#)), aplicativo desenvolvido por Danilo Dadonas e tem por objetivo gerenciar e controlar finanças pessoais. Porém é um aplicativo voltado para o gerenciamento financeiro pessoal, sem levar em conta uma empresa ou situação de mercado específica.

Todos os outros aplicativos encontrados e disponibilizados são referentes ao gerenciamento pessoal sem algum tipo de especificação quanto a regra de negócio, sendo assim de certa forma modelos genéricos, logo a proposta deste projeto é melhorar esta abordagem e trabalhar um caso específico de negócio.

Pretende-se usando as mesmas ideias destes aplicativos, construir um software mais elaborado e com mais recursos para controle, visando a melhora significativa no dia a dia das consultoras, que são os clientes esperados para este aplicativo.

O aplicativo criado focou no controle de todo o estoque das consultoras, tratando os pedidos e o cadastro de produtos, clientes e eventos, seu diferencial maior é que se tratou de uma aplicação mais específica e aplicada as regras de negócio das consultoras.

Assim, o aplicativo conta com funcionalidades específicas como a marcação de sessões e compromissos e o estilo de cadastro de produtos, separados em categorias e subcategorias.

Além das funcionalidades o *design* do mesmo e sua estrutura de funções foram baseadas em um estudo do público alvo, assim usando técnicas conhecidas da IHC, pode-se criar um sistema com uma interface mais amigável e intuitiva melhorando o valor do aplicativo.

Uma funcionalidade importante é o controle financeiro feito de acordo com a meta cadastrada pelo usuário, essa por sua vez se baseando nas regras de negócio especificadas pelas consultoras, mais um diferencial desse sistema. Esse controle financeiro oferece um resumo de toda movimentação do aplicativo e dos ganhos financeiros da consultora.

4 Metodologia

Levando em consideração os objetivos e a ideia geral sobre o software, a proposta é usar o Processo Unificado como modelo de desenvolvimento e análise. Este Processo é um modelo de desenvolvimento de software que segue um processo iterativo para construção de sistemas orientado a objetos, de acordo com (LARMAN, 2007).

O modelo PU é bastante flexível e suscetível a práticas vistas em alguns modelos ágeis e que serão necessárias na concretização desse projeto. É um modelo baseado no processo iterativo e evolutivo.

Após realizar um estudo de caso que gerou a introdução deste trabalho, pode-se definir as fases e suas iterações, que serão feitas no decorrer do projeto. Essas fases serão especificadas a seguir.

4.1 Concepção

A fase de concepção trata das primeiras definições do projeto para criação de uma visão de alto nível do projeto, assim podendo delimitar o escopo do produto, e estimar os casos de risco. É definido um diagrama da UML para representar o projeto em alto nível destacando-se módulos contidos no projeto (LARMAN, 2007).

A figura 2 representa de maneira geral a estrutura central da aplicação, com os módulos principais e suas interdependências, que estão representadas pelas linhas pontilhadas. O módulo Cliente trata do cadastro e controle dos clientes para eventuais consul-

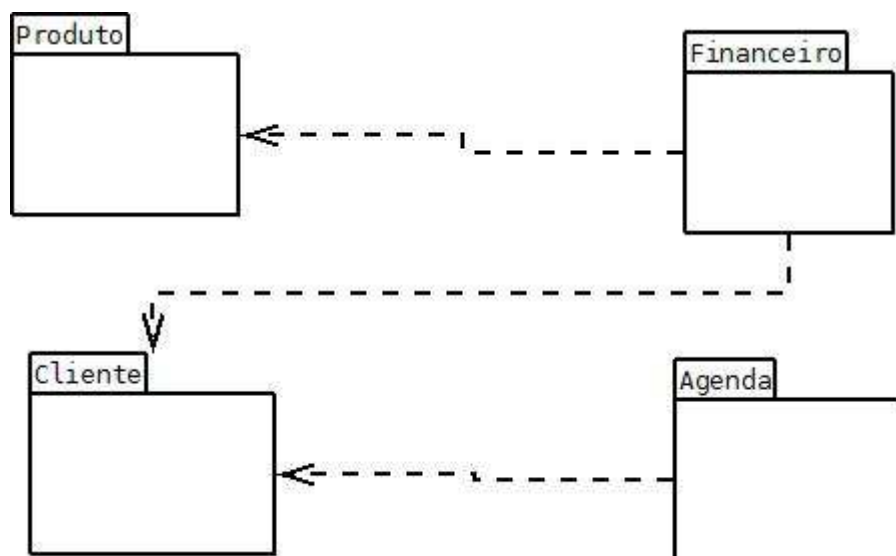


Figura 2 – Diagrama UML: Módulos do sistema

tas, o módulo Produto como o próprio nome diz controla os produtos mas também suas categorias e subcategorias, podendo ser feito cadastros e edições nos mesmos.

O módulo Agenda trata do controle de eventos de uma consultora, no caso das sessões e compromissos, os quais serão explicados com mais detalhes no documento de modelo de casos de uso e regras de negócio. Este módulo faz uso e depende do módulo Cliente.

O módulo de maior risco e talvez o de maior complexidade é o Financeiro. Nele há o controle de Pedidos, Estoque e Trocas.

A partir do diagrama da figura 2 há a definição da ordem de implementação, baseando na interdependência dos módulos, é comum que essa fase seja pequena e de pouca duração, logo tendo duração de 2 dias caso fosse feita em uma equipe.

Serão usados os artefatos da fase de concepção para aprimorar o projeto e organizar os requisitos existentes, bem como todas as ferramentas disponíveis nessa fase. Para melhor impulsionar o projeto e concluir sua visão geral.

4.2 Elaboração

Nesta fase serão tratados os requisitos de maior prioridade, como já descrito na Seção 4.1, usando parte da elaboração da visão do projeto bem como sua estrutura inicial para planejar as iterações e quais casos de uso tratar (LARMAN, 2007). Para esta fase então estima-se a utilização de 1 a 3 semanas para tratar todos os detalhes desta fase de acordo com o projeto, já começando a implementá-lo.

Como as duas primeiras iterações serão deixadas para a fase de concepção, estima-se que nesta fase precisa-se de no mínimo 5 semanas sujeitas a mudanças de acordo com os requisitos que forem sendo encontrados e de acordo com a prioridade de tratamento destes.

A implementação dos principais requisitos, no caso a elaboração deles, é feita nesta fase. Respeitando as dependências dos módulos serão tratados os principais casos de uso de cada módulo de forma rápida criando uma estrutura de alto nível, que será tratada com mais detalhes na fase de construção. Aqui o foco será tratar um pouquinho de cada módulo focando nos requisitos mais importantes.

4.3 Construção

Na fase de construção já tem-se em mão uma estrutura de projeto, uma base implementada e uma pequena documentação dos casos de uso e relações para ajudar a

construir todo o restante do software (LARMAN, 2007). Neste momento, tratam-se os detalhes, bem como a interface do software.

É na fase de construção que também se dá o desenvolvimento dos componentes do software, bem como o uso de ferramentas para produzir o software, que será explicado na Seção 5. Desde já, enfatiza-se que para construir este software, será usada a plataforma Android com ajuda do Sistema de Banco de Dados SQLite.

Pode-se então, baseado no modelo da figura 2]do software completo com seus módulos, fazer uma previsão de como serão tratados os módulos de acordo com a prioridade e as relações existente entre eles. De acordo com a figura pode-se tratar o restante dos requisitos seguindo a ordem de módulos:

1. **Módulo Cliente:** É um módulo importante e não depende de nenhum módulo, contudo é pequeno e de fácil implementação precisando talvez de apenas uma semana.
2. **Módulo Produtos:** Por ser um módulo necessário na maioria dos outros, deve-se tratá-lo por segundo, logo é separado uma semana para o mesmo, visto também que sua implementação não será tão complexa.
3. **Módulo Agenda:** Talvez este módulo tenha tanta importância quanto o financeiro, pois através dele que o usuário do software poderá estar a par dos seus compromissos e sessões de demonstração. É um módulo pouco complexo mas de muitos detalhes logo usando 3 semanas.
4. **Módulo Financeiro:** Depende de um módulo já tratado que é o de produtos e é um módulo grande e propenso a falhas, logo deve-se garantir mais iterações para tal, sendo então pre vistas de 2 a 4 semanas.

Haverá uma preocupação para os testes de desenvolvimento, uma vez que auxiliarão nas iterações mostrando quais aspectos devem ser tratados naquele momento, sendo então testes informais e sem muito planejamento. Contudo alguns módulos requerem tratamento mais adequado e um planejamento dos testes, indicando todos os casos a testar e itens para análise, seguindo a especificação de planejamento estudado em (FILHO, 2009)

Nesta fase também haverá uma preocupação com a interface produzida para o software, logo será nesta fase a introdução dos conceitos de IHC para gerar uma interface mais amigável com o cliente, melhorando o aprendizado e uso do mesmo.

A interface será implementada de maneira estratégica visando o maior grau de usabilidade, acessibilidade e comunicabilidade, sempre prestando atenção à interação do usuário e a *affordance* dos componentes na interface. Antes de concluída, também será

testada e avaliada em um ambiente real, gerando documentos de avaliação e resultados de questionários aplicados.

4.4 Transição

Com o projeto implementado e completo nesta fase é implantado o software nos celulares e feitos testes com os usuários para analisar a eficiência do software em um ambiente real, o software que terá o nome de Meu Estoque Rosa , será implantado em no mínimo cinco aparelhos que suportem o sistema Android 4.0, e serão feitos testes de caixa-preta, caixa-cinza e caixa-branca, bem como testes unitários e de qualidade para finalização do projeto.

Nesse momento, será importante focar nas saídas obtidas pelo software, de acordo com o uso do mesmo no ambiente de trabalho diário, sendo preciso fazer testes de aceitação e operacional.

Nesta fase, o foco será na avaliação junto ao cliente final, definindo a eficiência e eficácia do projeto e concluindo com os resultados obtidos, sendo então gastos no mínimo umas quatro semanas para avaliação e coleta de dados finais. A avaliação consistirá de entrevistas com os usuários e testes de sistema, onde serão colhidos dados para verificação.

Os resultados obtidos nessa fase serão avaliados e a conclusão serão discutidos no capítulo de Conclusão, Seção (6)

5 Desenvolvimento

Neste capítulo serão detalhados todos os procedimentos de criação do software, bem como toda a documentação específica de acordo com a metodologia escolhida.

Usando o PU como base da organização e desenvolvimento, a seguir será explicado cada passo de cada fase, e todas as implementações feitas e ferramentas usadas. (LARMAN, 2007).

5.1 Concepção

Nesta fase, foi definida como seria a arquitetura do sistema, usando como referência o diagrama dos módulos da Figura 2 e o diagrama de classes de alto nível do projeto completo da figura 3. De acordo com essa primeira especificação foi montado um documento de arquitetura do software do apêndice B, detalhando as partes mais importantes e de maior risco.

Nesta fase também foram introduzidos alguns diagramas de classes de alto nível para definir como seriam abordados os módulos e os componentes que seriam implementados primeiro, pois para este projeto foi de extrema importância definir a relação das tabelas no banco de dados, uma vez que estas são a base do nosso sistema, sendo a primeira camada abordada. Estes diagramas de classes definiram as colunas das tabelas principais.

Assim pode-se definir um modelo de domínio especificado: No módulo Cliente temos o controle de clientes e apenas a classe relacionada a ele, portanto não diagramada.

No módulo Produtos visto na Figura 3, temos o controle de Produtos, Categoria e Sub Categoria, uma vez que o produto pode estar em uma categoria e/ou numa sub categoria, portanto como esta especificado no diagrama, bem como a sub categoria deve estar contida em uma categoria.

No módulo Financeiro temos o controle do estoque (Figura 3), relacionando com os produtos, bem como controle de pedidos e uma lista de produtos para cada pedido. Nele temos também a meta Trimestral, que marca a meta diária e mensal. Ele faz uso do módulo produtos e do módulo clientes para associar os pedido.

No módulo da Agenda temos as Sessões (Figura 3) que são marcadas com data, hora e relacionados com um cliente, e os Compromissos que são marcados com data, hora, observações, descrição e prioridade, além de poder relacionar com um cliente caso seja um compromisso de entrega.

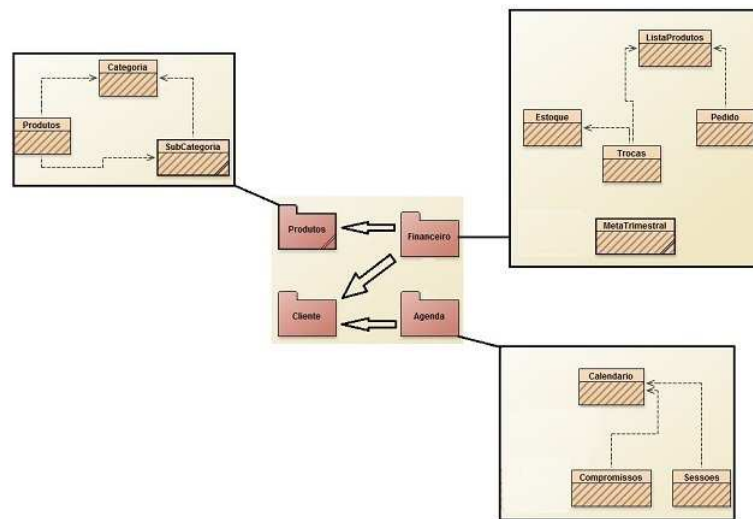


Figura 3 – Diagrama de Modelo: Módulos do sistema

Nesta etapa foram definidas as regras de negócio para o projeto, que são muito importantes na implementação do módulo financeiro e da agenda, estes estão fortemente baseados nas regras e métodos do dia a dia de uma consultora, como por exemplo, como é feita a meta trimestral da consultora e o que é preciso para a computação dos dados para manter a consultora ciente de sua condição dentro da meta estipulada por ela , as regras podem ser vistas no Apêndice A.

5.2 Elaboração

Na etapa de elaboração, o foco foi em como e por que de ser projetado o sistema como foi projetado, pensando nos aspectos mais importantes e mais impactantes do projeto, exigindo um tempo maior para tal.

A arquitetura escolhida para esse projeto foi a MVC , que foi adequada para o ambiente Android, assim também contando com a camada de banco de dados chamada DAO. Assim a camada de visão e controle estão associadas com as classes UI, a camada de modelo está associada ao pacote POJO com as classes (nome da classe)VO.

Neste momento também aliada a arquitetura escolhida foi projetado como seria o banco de dados e o modelo relacional, bem como o diagrama entidade relacionamento (Figura 4). O banco de dados é uma parte importante do projeto e garante o armazenamento de todos os dados trafegados dentro da aplicação, logo foi preciso trabalhar bem a relação das tabelas para evitar informações redundantes e a facilidade de acesso aos dados.

A Figura 4 representa o diagrama entidade-relacionamento completo da aplicação

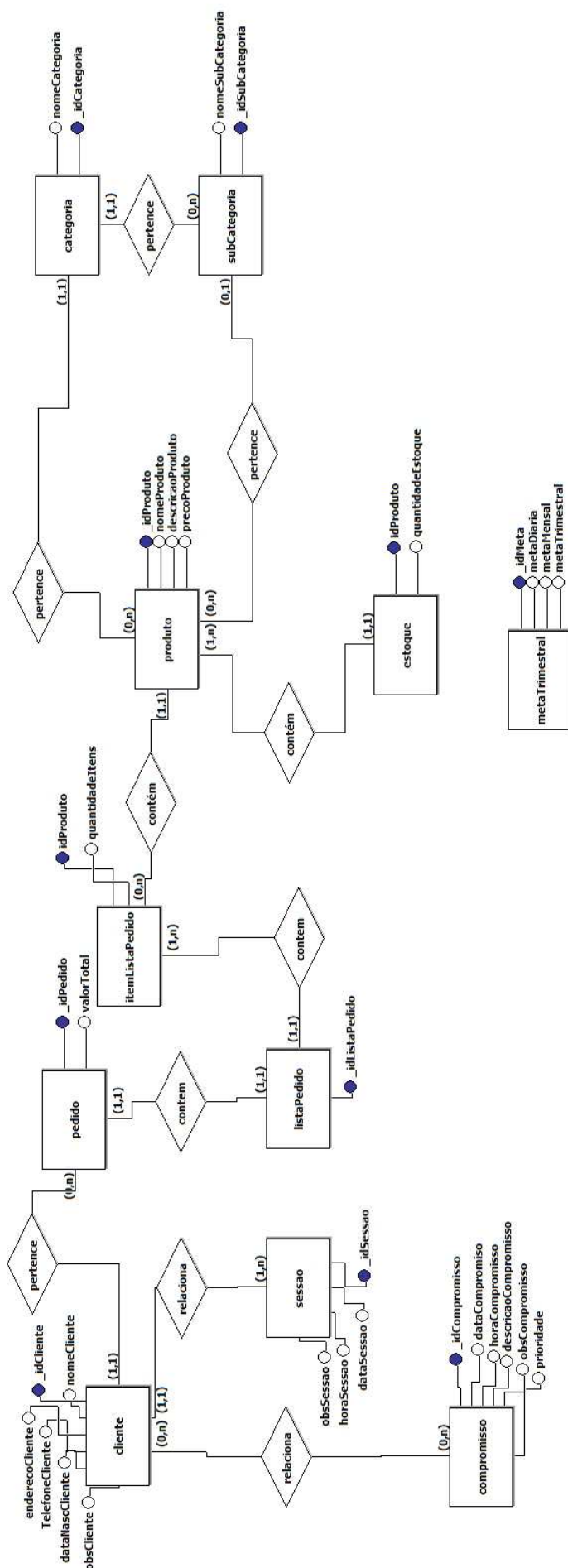


Figura 4 – Diagrama DER: Projeto Meu Estoque Rosa (Completo)

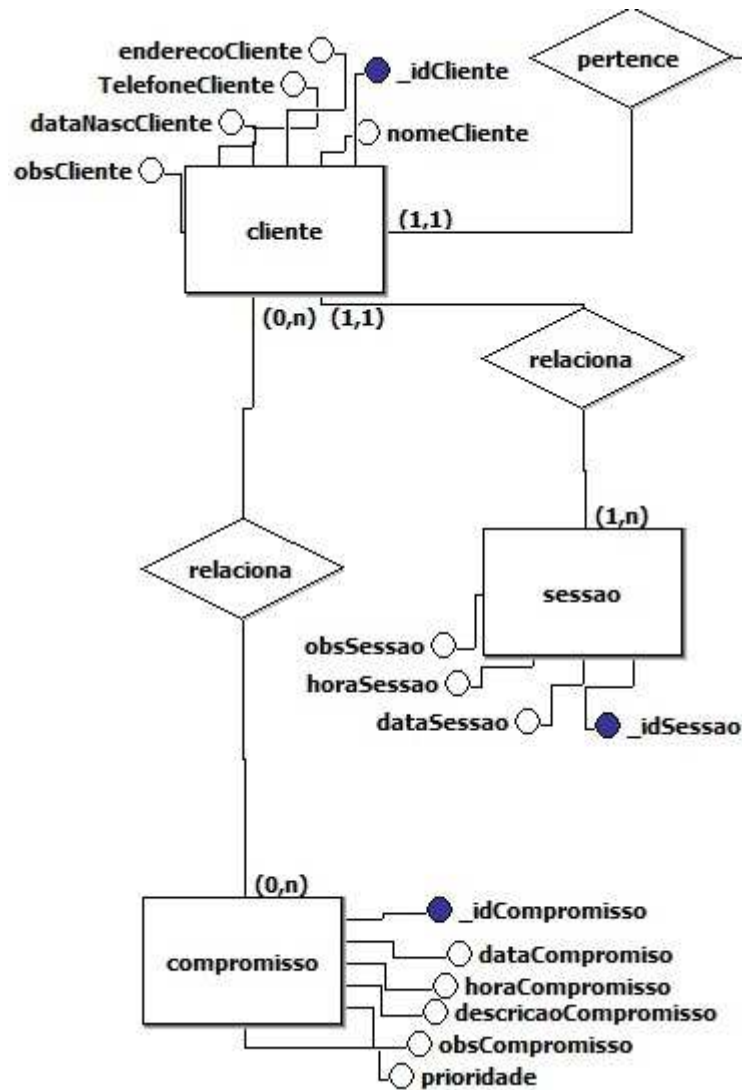


Figura 5 – Diagrama DER: Projeto Meu Estoque Rosa (Agenda x Cliente)

e mostra todas as relações que compõem a estrutura de banco de dados, pode-se também observar como é a relação entre o módulo Agenda e Cliente na Figura 5, que mostra como estão relacionadas as classes cliente, sessão e compromisso e qual grau de relacionamento delas. Outra visão de relações também pode ser vista na Figura 6 que mostra o relacionamento do módulo financeiro com o cliente e como são as relações entre as classes que os compõem. Por fim pode-se mostrar a relação entre o módulo financeiro e o de produtos na Figura 7, todas esses diagramas são essenciais para a elaboração do modelo entidade-relacionamento que pode ser visto na Figura 8.

É importante ressaltar que o SQLite garante o armazenamento de dados sem gastar muita memória e também as relações de dependência asseguradas por *foreign keys*, assim pode-se assegurar que o modelo de banco de dados criado não irá impactar na performance e na memória do celular (HIPP, 2014).

Neste momento, foram definidas as interfaces, no caso quais seriam os atributos

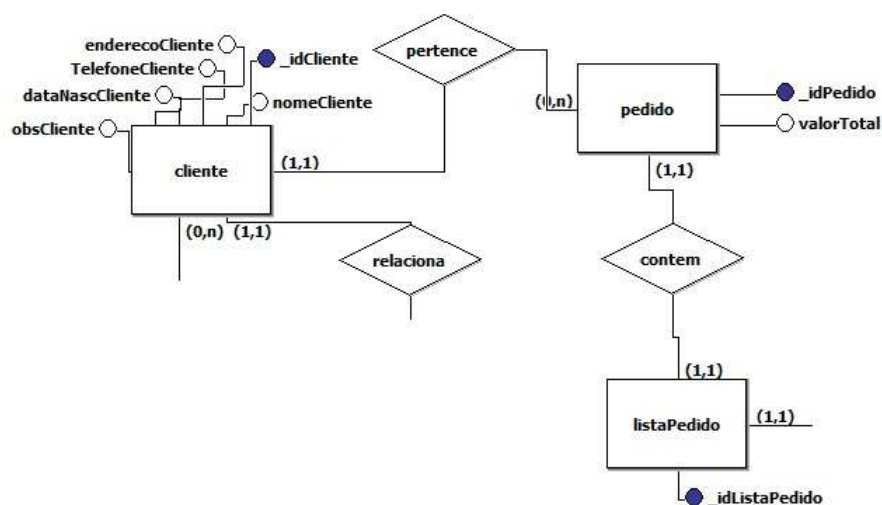


Figura 6 – Diagrama DER: Projeto Meu Estoque Rosa (Financeiro x Cliente)

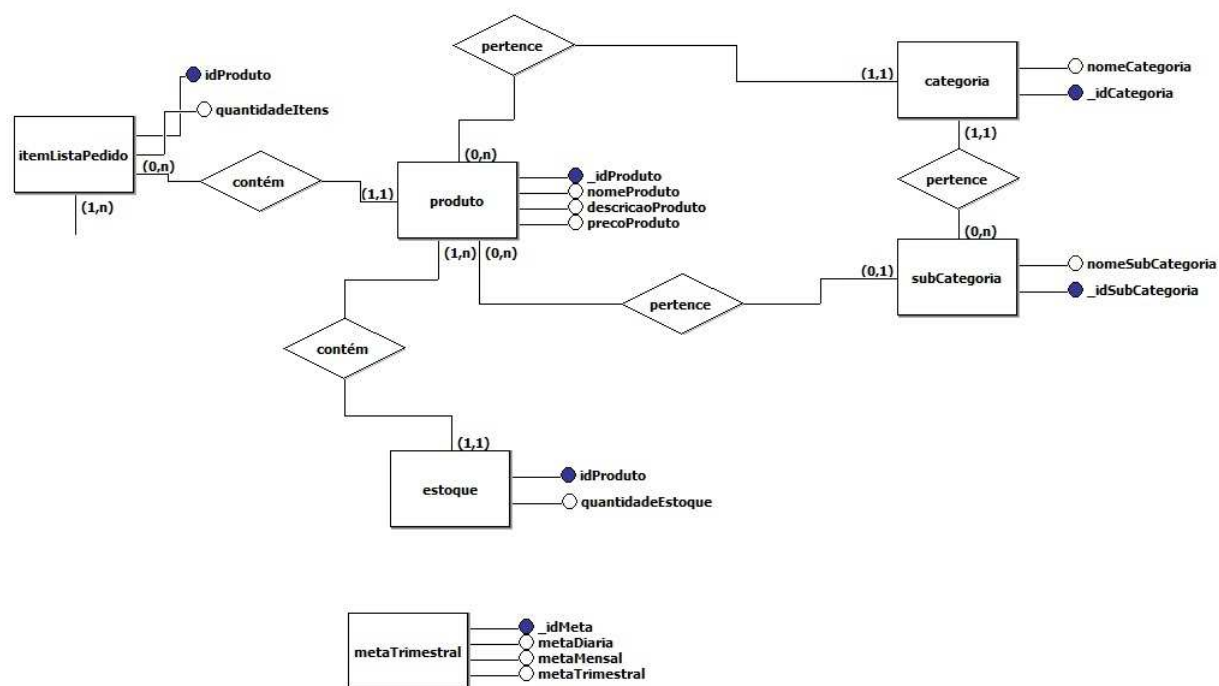


Figura 7 – Diagrama DER: Projeto Meu Estoque Rosa (Financeiro x Produto)

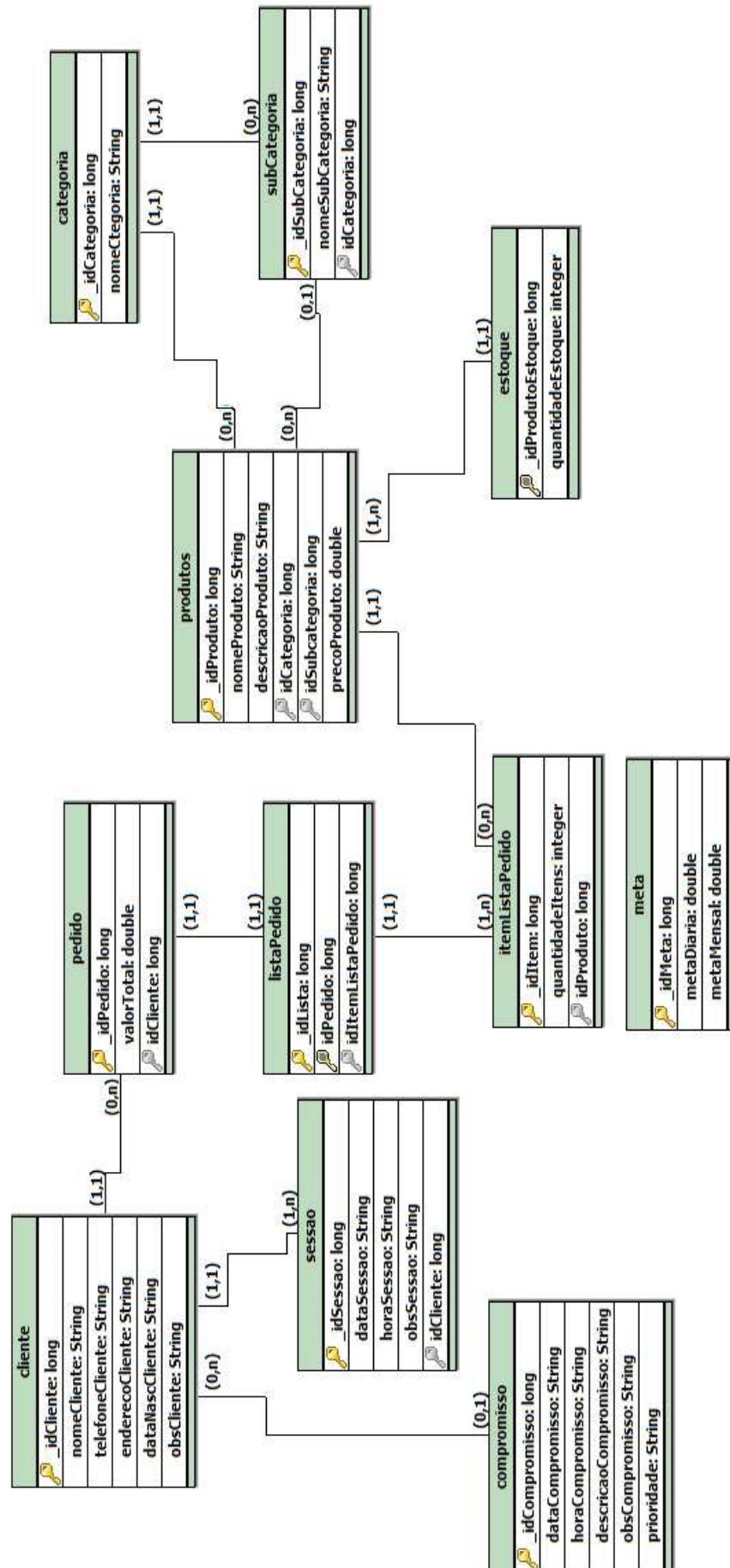


Figura 8 – Diagrama MER: Projeto Meu Estoque Rosa

necessários a estas, foi feita uma análise simples do perfil das consultoras, constatando que são na maioria mulheres, portanto sendo necessário manter cores mais agradáveis e de preferência que lembrem o padrão feminino do negócio. Na MaryKay são comuns as cores rosa, vermelho, azul cobalto e verde esmeralda, logo a cor escolhida padrão das interfaces foi rosa.

Tendo isso em mãos logo foi possível a implementação de algumas interfaces gráficas, também do banco de dados da aplicação, com as tabelas de produtos e a de clientes.

De acordo com as regras de negócio, o modelo de domínio e o modelo do banco de dados, podem-se definir os casos de uso do sistema, detalhados por uma descrição textual. Para cada função foi feito um detalhamento de seu fluxo e como seriam as interações, para exemplificar tem-se o caso de uso: Cadastro de Pedido e Itens.

Cadastrar pedido e itens

Escopo: O sistema deverá cadastrar o pedido e também os itens do pedido podendo ser cadastrado separadamente em dois fluxos de atividades .

Pré-Condição: Ter clientes cadastrados no sistema e produtos no estoque.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso o pedido foi ou não cadastrado bem como os itens. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de pedidos cadastrados atualizada.

Cenário de sucesso Principal:

Fluxo principal : Cadastro do Pedido

1. O usuário clica no botão cadastrar pedido da tela financeira.
2. O sistema redireciona para a tela de cadastro de pedido com os campos: Nome Cliente, Valor Total, Numero de Parcelas, Valor Parcela.
3. O usuário escolhe o cliente e clica no botão salvar caso queira salvar o cadastro ou no botão adicionar produtos caso queira salvar e adicionar itens ao pedido.
4. O sistema insere no banco retornando a mensagem de sucesso para o usuário.
5. O sistema redireciona para a tela financeira caso tenha clicado em salvar, mostrando uma mensagem de sucesso.

Fluxo secundário : Cadastro de Itens do Pedido

1. Após o preenchimento dos campos o usuário clica em adicionar itens.
2. O sistema salva o pedido retornando mensagem de sucesso e redireciona para a tela de cadastro de itens do pedido.
3. Na tela o usuário preenche os seguintes campos: Nome do Produto(Selecionável) e quantidade.
4. O usuário clica em adicionar.
5. O sistema verifica os campos e salva no banco mostrando mensagem de Item adicionado, retorna pra mesma tela mas com os campos limpos.

Esta declaração mostra com detalhes como é o fluxo de informações e dados no

sistema. Para a obtenção de uma visão mais simples podem-se usar diagramas de atividades que mostram as atividades feitas durante o fluxo, além do diagrama de sequência que mostra todas as trocas de mensagens entre as classes e as camadas do sistema. Um exemplo de diagrama de atividades está ilustrado na Figura 9 que mostra a diagramação do caso de uso acima.

Um exemplo de diagrama de sequência que exemplifica o caso de uso cadastro de pedidos e itens esta no diagrama da Figura 10.

Assim tem-se os diagramas necessários na construção da aplicação, o restante dos casos de uso podem ser vistos na Seção de Apêndices C, os diagramas de atividades e de sequência podem ser vistos na Seção de Apêndices D. Esta fase segue junto a construção uma vez que os casos de uso vão sendo especificados e elaborados durante o projeto, nesta fase também há a elaboração dos testes de IHC, a avaliação por questionário e a avaliação heurística.

Uma questão muito importante é o protótipo do sistema, de acordo com (PRE-ECE YVONNE ROGERS, 2005) podem-se fazer protótipos de baixo e alta fidelidade. Considerando-se o tempo reduzido para a realização do projeto foi escolhido fazer direto o protótipo de alta fidelidade para que o acompanhamento com o usuário fosse mais agradável. O ambiente de desenvolvimento para Android, oferece suporte à criação de interfaces gráficas não só por código mas também pelo editor gráfico.

Usando dados obtidos por meio de conversas com o usuário foi possível estimar como seria a interface gráfica do sistema como pode ser visto na Figura 11, contudo de acordo com as técnicas de observação do usuário e do ambiente de trabalho pode-se formular uma interface mais amigável e intuitiva (BARBOSA, 2010) como pode ser vista na figura 12, que é a mesma ideia da primeira figura contudo mais trabalhada e com ícones intuitivos.

Para fins de avaliação tanto da interface gráfica, como também da interatividade das telas e componentes com suas devidas funções, foi realizada uma avaliação com 10 consultoras usando um protótipo do sistema já com algumas funções implementadas e finalizadas.

O processo de avaliação foi feito da seguinte forma, usando um celular com Android de versão 4.0 e com a aplicação incompleta instalada, as consultoras usaram a aplicação dando ênfase a interface gráfica. Logo depois responderam um questionário com 9 perguntas sobre a interface e suas funcionalidades avaliando de 1 a 5 de acordo com a legenda do questionário da tabela 4.

A interface gráfica que foi usada para avaliação foi a das Figuras 12 e 13 que representam respectivamente a tela principal e a tela de cadastro de cliente, outros exemplos podem ser vistos na Seção do Apêndice E.

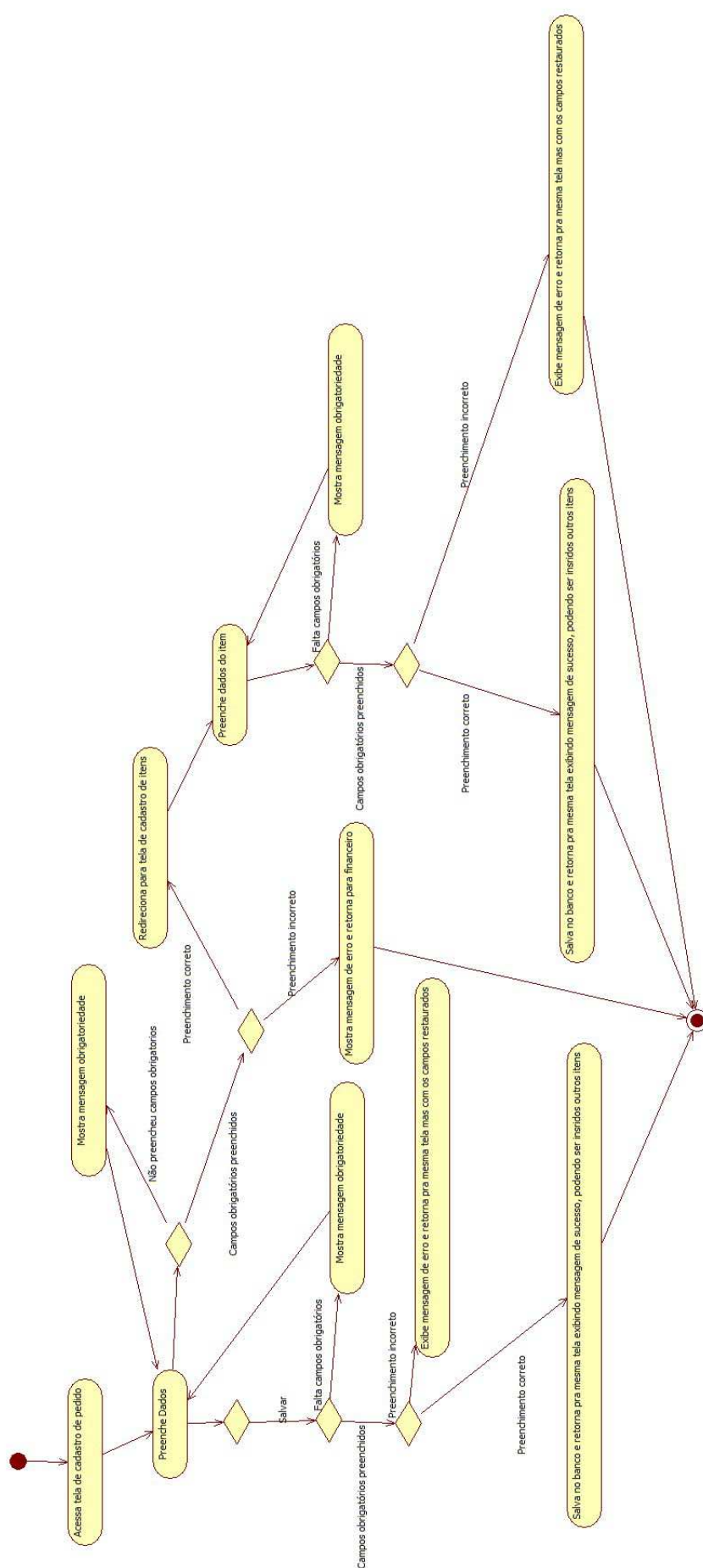


Figura 9 – Diagrama de Atividades para o caso de uso de cadastro de pedido e itens do pedido.

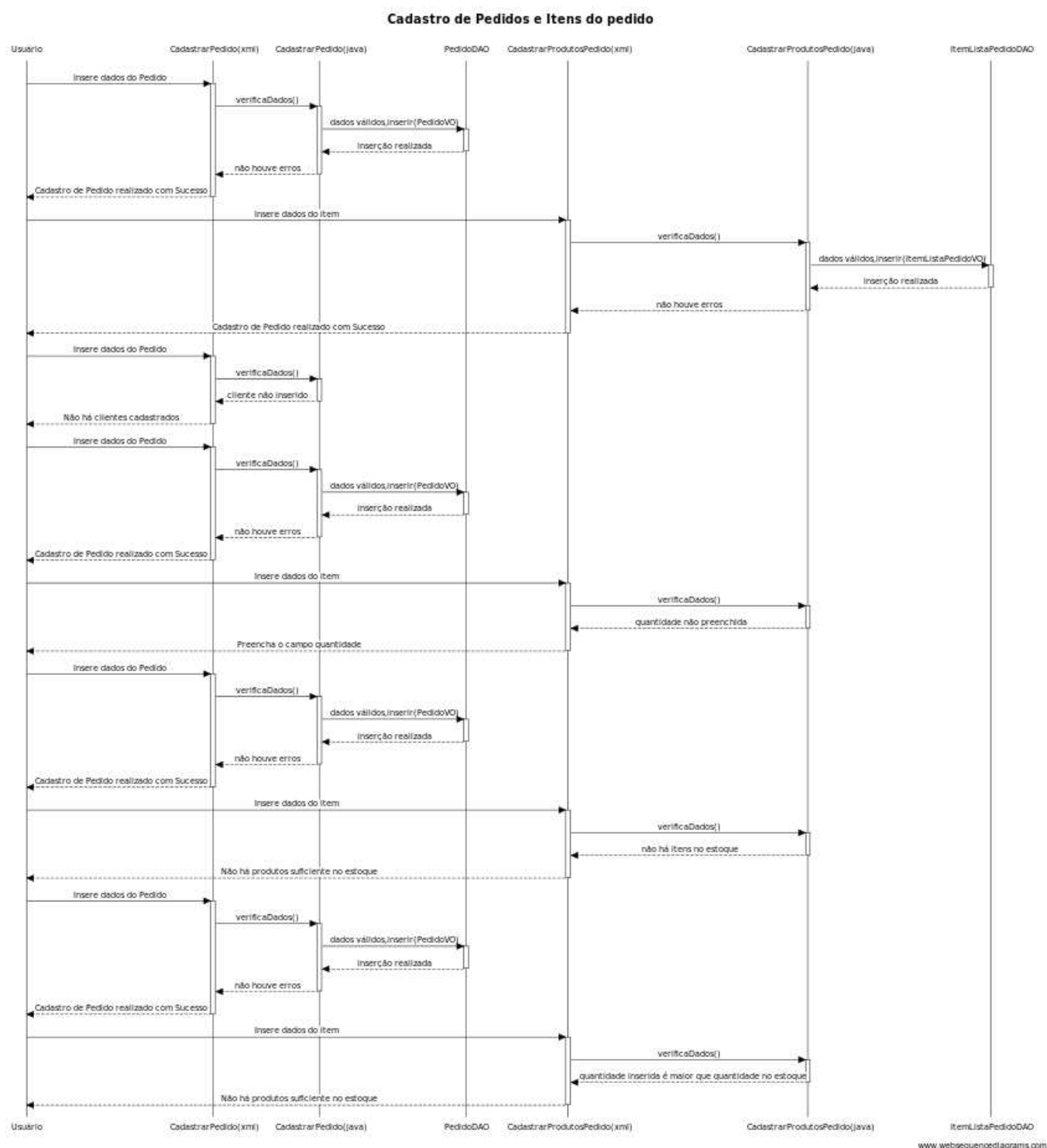


Figura 10 – Diagrama de Sequência para o caso de uso de cadastro de pedido e itens do pedido.



Figura 11 – Protótipo Hi-Fi : Primeira tela principal criada para a aplicação Meu Estoque Rosa, levando em consideração as características do usuário final.



Figura 12 – Protótipo Hi-Fi : Tela principal.

O questionário foi feito baseando-se na interface e em como ela comportaria, visando o entendimento e a facilidade de uso da aplicação, o resultado bem como o questionário aplicado é mostrado a seguir na Tabela 3. O questionário foi respondido por sete consultoras da marca Mary Kay.

Fazendo uma análise dos resultados, pode-se concluir que no geral o sistema tem uma boa aceitação pelo público-alvo. Pela análise das afirmações 1, 3 e 9 pode-se dizer que a interface gráfica atende bem ao propósito, talvez trabalhando um pouco mais os componentes e a visão da mesma, agradaria mais o usuário final.

Pelas afirmações 2 e 7 pode-se dizer que as telas cumprem seu propósito e o usuário não precisaria de um curso ou ajuda para usar o software, poupando tempo e gastos com aprendizado.

Levando em consideração a média obtida pelas afirmações 5 e 8, pode-se prever que o aplicativo agradará o consumidor e será muito usado uma vez que teve notas altas

2G 25% 7h55

Cadastro Clientes

Preencha os Campos

Nome: _____

Telefone: _____

Endereço: _____

Data de Nascimento: _____

Observações: _____

Aceitar Cancelar Voltar

QuickMemo

Figura 13 – Protótipo Hi-Fi :Tela de cadastro de clientes.

nestas afirmações.

Levando em consideração as afirmações 4 e 6, pode-se dizer que o aplicativo é bom, contudo com o tempo poderia ser melhorado para aumentar sua eficiência e atender a necessidade mais específica do usuário.

Neste mesmo questionário foi dado ao usuário a possibilidade de sugerir uma melhoria, resultando em uma sugestão que seria importante para próximas versões do mesmo, bem como para esta, que seria fazer a junção de dados com a agenda do Google e notificar às consultoras sobre o programa 2+2+2, que trata-se de um programa de acompanhamento com o cliente.

A avaliação heurística feita sobre o protótipo de alta fidelidade parcialmente finalizado detectou algumas inconformidades, estas por sua vez são especificadas seguindo o método de avaliação proposto por Nielsen, assim pode-se ver o resultado na Tabela 5 .

Com os resultados da avaliação heurística pode-se melhorar um pouco a interface

Afirmações	Média de Nota (5)	Avaliação da média
A disposição das telas é intuitiva, eficiente e atende minhas expectativas.	4,4	Concordam
Entendo todas as funcionalidades do aplicativo.	4,3	Concordam
Gosto das cores e me identifico com as telas e disposição do conteúdo.	4,7	Concordam Plenamente
O aplicativo atende as minhas necessidades.	4,5	Concordam
Usaria com frequência o aplicativo.	4,7	Concordam Plenamente
Apresenta as mensagens de forma clara, notificando qualquer alteração nas informações.	4,3	Concordam
O aplicativo é de fácil uso e não tenho dificuldades em achar suas opções.	4,7	Concordam Plenamente
Indicaria o aplicativo a outra consultora.	4,9	Concordam Plenamente
Os nomes e ações do aplicativo, bem como as imagens são adequadas a sua funcionalidade específica.	4,5	Concordam

Tabela 3 – Questionário com resultados obtidos.

Nota	Significado
1	Discordo Plenamente
2	Discordo
3	Neutro
4	Concordo
5	Concordo Plenamente

Tabela 4 – Tabela de notas e significado.

aplicando as mudanças propostas, que podem ser vistas nas figuras 14, 15 e 16, como pode ser visto nestas imagens foram aplicadas os ícones de ajuda e uma forma diferente de fazer o cadastro de datas e horas para ficar mais dinâmica a tela.

Módulo Cliente				
Problema	Heurística violada	Explicação	Gravidade	Proposta de solução
Ao inserir a data, se a data é inválida o erro mostrado não é tratado, apresentando ao usuário um erro que ele não consegue identificar por se tratar de um erro técnico de banco de dados.	Ajuda no reconhecimento de erros, diagnóstico e recuperação de erros	Esse erro ocorre quando o sistema tenta fazer uma validação de data mas por ser inválida ele apresenta a mensagem de erro da aplicação, a qual é de entendimento apenas para os desenvolvedores.	3 Grave. Pois o usuário não saberá como proceder ao erro e poderá perder as informações inseridas anteriormente.	Tratamento dos erros gerados pela <i>exception</i> mostrando uma mensagem mais próxima ao usuário.
Na tela de Cliente, bem como nas telas acessadas a partir dela não há nenhum botão para ajuda ou explicação das funções	Ajuda e documentação	Para que o usuário saiba tudo sobre as funções e não se confunda na hora de usar o aplicativo, é essencial dispor de um modo de mostrar uma aba ou tela de ajuda para eventuais consultas ou dúvidas.	2 Pequeno. Pois o usuário não terá nenhuma fonte de ajuda e talvez não seja possível contactar um administrador.	Adicionar ícones de ajuda em todas as telas com as devidas colocações para as mesmas e observações.
Módulo Produtos				
O inserir o preço do produto erro mostrado não é tratado, apresentando ao usuário um erro que ele não consegue identificar por se tratar de um erro técnico de banco de dados.	Ajuda no reconhecimento de erros, diagnóstico e recuperação de erros	Esse erro ocorre quando o sistema tenta fazer uma inserção do valor do campo preço mas por ser inválido ele apresenta a mensagem de erro da aplicação, a qual é de entendimento apenas para os desenvolvedores.	3 Grave. Pois o usuário não saberá como proceder ao erro e poderá perder as informações inseridas anteriormente.	Tratamento dos erros gerados pela <i>exception</i> mostrando uma mensagem mais próxima ao usuário.
Na tela de Produtos, bem como nas telas acessadas a partir dela não há nenhum botão para ajuda ou explicação das funções	Ajuda e documentação	Para que o usuário saiba tudo sobre as funções e não se confunda na hora de usar o aplicativo, é essencial dispor de um modo de mostrar uma aba ou tela de ajuda para eventuais consultas ou dúvidas.	2 Pequeno. Pois o usuário não terá nenhuma fonte de ajuda e talvez não seja possível contactar um administrador.	Adicionar ícones de ajuda em todas as telas com as devidas colocações para as mesmas e observações.
Na tela de cadastro a ordem de cadastro sugere uma ordem de cadastro que não funcionaria, a ordem de cadastro de um produto se não tivesse nenhuma categoria e nenhuma subcategoria cadastrada seria cadastro de categoria, de subcategoria se houvesse e enfim do produto.	Reconhecimento em vez do lembrar	Para que o usuário não fique frustrado na hora de inserir um produto e se depare com o fato de não poder inserir o produto sem que tenha categorias cadastradas, é importante manter a ordem de inserção, assim não é preciso lembrar da ordem evitando erros por má inserção de dados.	1 Cosmético. Pois usuário pode não lembrar que deve inserir uma categoria e acabar ficando frustrado por não poder inserir um novo produto, contudo com o tempo o usuário pode aprender a ordem.	Ordenar os botões na ordem de inserção.
Módulo Agenda				
Possui as mesmas inconformidades que o módulo cliente quando se tratando de ajuda no reconhecimento de erros, diagnóstico e recuperação de erros, e documentação.				
Módulo Financeiro				
Possui as mesmas inconformidades que o módulo cliente quando se tratando de ajuda no reconhecimento de erros, diagnóstico e recuperação de erros, e documentação.				

Tabela 5 – Avaliação Heurística por módulos



Figura 14 – Tela principal final.



Figura 15 – Tela financeira final.

21 dez 2014

22 jan 2015

23 fev 2016

Nome Cliente:

Valor Total: 0

Nº Parcelas: 0 Calcular

Valor Parcela: 0

OutkMemo

Figura 16 – Tela de cadastro de sessões final.

5.3 Construção

Na fase de construção ocorreu a maior parte da implementação e descoberta de detalhes de desenvolvimento, nesta fase inicia-se a implementação das funções mais específicas, começando pelos dois módulos mais simples, o de produtos e de clientes, usando como base da construção as regras de negócio descritas na Seção de Apêndice A, a descrição das funções no documento de arquitetura que está na Seção de Apêndice B, a descrição sucinta do fluxo de ações dos casos de uso descritos na Seção de Apêndice C, seguindo também os diagramas de atividades e de sequência que estão na Seção de Apêndice D.

Estes módulos são apenas de cadastro e manutenção de informações simples, uma vez cadastrados não há muitas mudanças, logo são de fácil implementação. Contudo são a base dos outros dois módulos, portanto sendo de grande importância que sejam estáveis e apresentem as informações corretas.

A programação destes dois módulos foi rápida mas foi de grande aprendizado e

padronização do projeto, assim os módulos seguintes foram baseados na estrutura destes e aprimorados. A partir destes módulos foram padronizadas as funções de cadastro, pesquisa, atualização e remoção e replicados aos outros módulos, no que diz respeito às notificações, padrão de demonstração das mensagens e das funcionalidades e de cadastro.

Através destes dois módulos pode-se ter uma base de como seria a construção das tabelas do banco para os outros módulos como a programação do sistema de controle e funções, assim gerando um padrão.

Assim com o exemplo dado no site ([SOUZA, 2014](#)) pode se implementar o banco e parte do controle de cadastro, alteração, pesquisa e exclusão do módulo de produtos e clientes. Um trecho do código que faz a criação do banco bem como das regras de inserção das tabelas usando SQLite pode ser visto na Seção de Apêndice [F](#):

Com as classes criadas no banco é possível criar todo o fluxo de código para atender aos casos de uso dos módulos cliente e produtos. Com esta base de código pode se replicar ao restante das classe dos módulos tendo atenção à suas particularidades que cada uma tem.

Para fazer o controle das classes, bem como as pesquisas e instruções de inserir, remover e alterar, cada classe ou seja cada tabela conta com uma classe em Java que faz comunicação com o banco, todas possuindo o sufixo DAO.java como pode ser visto na Seção de Apêndice [F](#).

Para fazer o transporte entre a camada DAO e a camada de Visão temos os objetos para cada tabela tratada no banco, as classes com sufixo POJO.java são os objetos de transporte de dados e são a representação dessas tabelas no código-fonte, um exemplo dessas classe pode ser visto na Seção de Apêndice [F](#) .

Com estas classes implementadas temos a base para a implementação das funções e controles estabelecidos nos casos de usos. É importante ressaltar que a camada de Visão é o código apenas de visão ao usuário, e diferente de uma aplicação *desktop* essa é garantida pelos arquivos de extensão .xml . Estes por sua vez são gerados em uma pasta separada dos pacotes de classes Java como pode ser visto na Figura [17](#), nela também é possível ver um exemplo de código XML .

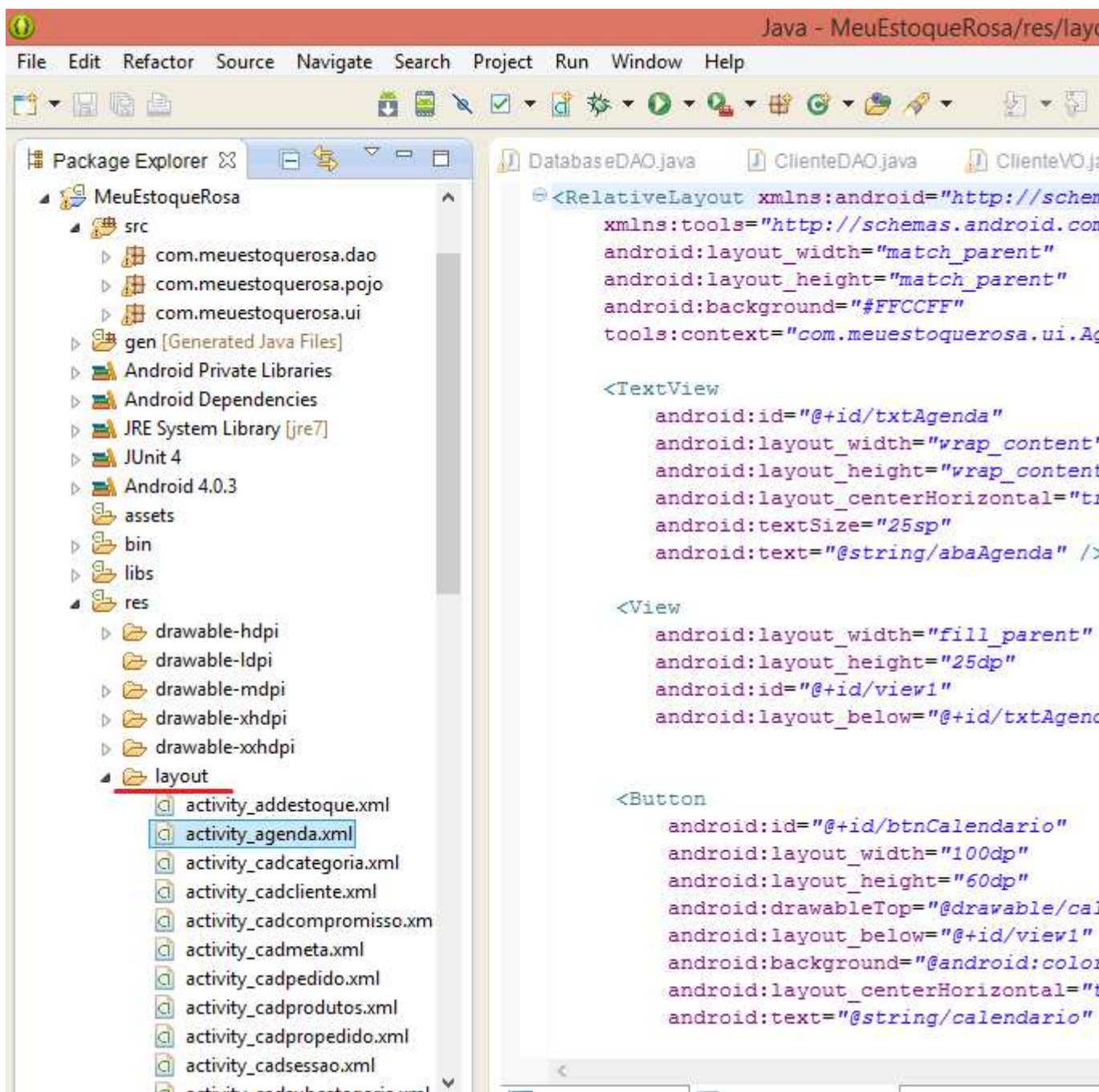


Figura 17 – Screenshot tirada do framework eclipse configurado com o ambiente de desenvolvimento Android, destacando a pasta de layout que é a camada de visão no Android

Para a implementação das mensagens mostradas ao usuário foram usados os diagramas de sequência, para cada caso de uso foi feito um diagrama e estes podem ser vistos na Seção de Apêndice D.

Houve um grande empenho na parte de cadastro de produtos, para que fosse feito de modo correto e para que as informações não fossem gravadas de forma errada arriscando a integridade do banco de dados, uma vez que este módulo conta com 3 tabelas

relacionadas umas as outras por meio de chaves estrangeira. A produção deste ofereceu base de programação para os outros módulos, que são fortemente acoplados. Por questões de exemplificação, na Seção de Apêndice F tem-se um exemplo de pesquisa ao banco.

O módulo agenda foi contemplado após o módulo produtos, baseando nas regras de negócio vistas na Seção de Apêndice A, na descrição das funções pelo documento de arquitetura descrita na Seção de Apêndice B e nos casos de uso do módulo da agenda que podem ser vistos na Seção de Apêndice C, assim como os diagramas de atividades e de para cadastro de sessões e compromissos vistos na seção de Apêndice D.

A implementação desse módulo fez uso de algumas funções para manipulação das datas em formato de *String*, se tratando do banco de dados, a construção deste módulo é simples, mas pelo código em Java ela se torna um pouco mais complexa. Assim para este módulo foi usado algumas referências importantes, como a do site da comunidade de desenvolvimento ([STACKOVERFLOW, 2014](#)) e o exemplo de calendário encontrado na comunidade Java Code Geeks ([ALIFERI, 2014](#)).

Para o retorno dos itens na pesquisa por intervalo de datas foi preciso manipular os dados retornados pelo banco, uma vez que a comparação por intervalo de datas não funcionava para todos os casos de pesquisa, a solução melhor para isso foi pegar todos os dados do banco equivalentes a classe em questão e fazer um filtro após a consulta, como pode ser visto no código nas linhas 14,16 e 17, este tratamento se repete no código e é usado para todas as manipulações de datas.

```
1 public List<SessaoVO> Consultar(String dataInicio,String dataFim)
   {
2 List<SessaoVO>
3 lstSessao = new ArrayList<SessaoVO>();
4 List<SessaoVO> lstAuxSessao = new ArrayList<SessaoVO>();
5 String consulta = "select * from "+DatabaseDAO.tblSessao+ " a
   inner join "+DatabaseDAO.tblCliente+" b on a."+ DatabaseDAO.
   idClienteFKSes+" = b."+DatabaseDAO.idCliente+" "
6 + " order by a."+DatabaseDAO.dataSessao;
7 Cursor cursor = database.rawQuery(consulta,null);
8 cursor.moveToFirst();
9 while (!cursor.isAfterLast()) {
10 SessaoVO lSessaoVO = cursorToSessao(cursor);
11 lstSessao.add(lSessaoVO);
12 cursor.moveToNext();
13 }
14 cursor.close();
15 if(dataInicio.equals("") && dataFim.equals("")){
16 for (SessaoVO item : lstSessao){
```

```
17 long data = Long.parseLong(getDateTime().replace("-", ""));
18 long dataItem = Long.parseLong(item.getDataSessao().replace("-",
    ""));
19 if(dataItem >= data){
20     lstAuxSessao.add(item);
21 }
22 }
23 }else if (!dataInicio.equals("") && dataFim.equals("")){
24     for (SessaoVO item : lstSessao){
25         long data = Long.parseLong(dataInicio.replace("-", ""));
26         long dataItem = Long.parseLong(item.getDataSessao().replace("-",
27             ""));
28         if(dataItem >= data){
29             lstAuxSessao.add(item);
30         }
31     }
32 }else if(dataInicio.equals("") && !dataFim.equals("")){
33     for (SessaoVO item : lstSessao){
34         long data = Long.parseLong(dataFim.replace("-", ""));
35         long dataItem = Long.parseLong(item.getDataSessao().replace("-",
36             ""));
37         if(dataItem <= data){
38             lstAuxSessao.add(item);
39         }
40     }
41 }else if(!dataInicio.equals("") && !dataFim.equals("")){
42     for (SessaoVO item : lstSessao){
43         long dataI = Long.parseLong(dataInicio.replace("-", ""));
44         long dataF = Long.parseLong(dataFim.replace("-", ""));
45         long dataItem = Long.parseLong(item.getDataSessao().replace("-",
46             ""));
47         if(dataItem >= dataI && dataItem <= dataF){
48             lstAuxSessao.add(item);
49         }
50     }
51 }
52 return lstAuxSessao;
53 }
```

O módulo financeiro foi abordado após o módulo agenda e é o módulo mais complexo do sistema, uma vez que ele é responsável pelo controle de estoque, pedidos, metas e

resumo financeiro. Assim este módulo levou mais tempo que os outros e contou com mais funções de controle tanto para datas e valores reais, como também para manutenção das informações dentro do banco. Baseou-se nos documentos básicos assim como os outros módulos e contou com a descrição dos casos de uso vistos na Seção de Apêndice C. Para construção do módulo financeiro foram usados como base os diagramas de atividades e de sequência que podem ser vistos na Seção de Apêndice D.

Este módulo conta com sete funções visíveis além das funções de edição ao clicar nas listas obtidas das pesquisas, nele foi implementada a função de adição de produtos ao estoque e pesquisa do mesmo, a função de cadastro de pedidos e nela também a de cadastro de itens do pedido, que está interligada tanto com o módulo cliente quanto o módulo produtos, também conta com a função de pesquisa de pedidos, itens e produtos em falta. Por ultimo tem-se o resumo financeiro e cadastro/pesquisa de metas.

No resumo financeiro tem-se um resumo dos ganhos do usuário, baseado nos pedidos feitos e de acordo com a meta cadastrada. Esta função calcula quanto falta ou o que está passando para a meta diária, mensal e trimestral, a meta é baseada de acordo com a data atual e o período, sendo que são quatro períodos no ano. Também é mostrado ao usuário o resumo do estoque e da agenda.

5.4 Transição

A fase de transição é a ultima fase do processo unificado e tem como objetivo a realização de testes e finalização da construção da aplicação e implantação, inicialmente começando com os testes unitários.

Para isto foram montados modelos de testes unitários para cada caso de uso descrito no modelo de casos de uso que está na Seção de Apêndice C, um exemplo de modelo de teste pode ser visto na Tabela 16, e na Tabela 7 que descrevem com detalhes como será aplicação do teste para o caso de uso cadastro de pedidos e o de itens do pedido, sendo os testes mais complexo por tratar vários objetos e ter pré-condições.

Cada caso de teste especificado possui um objetivo que é a descrição do que deve ser obtido e testado no caso de teste, descrição que descreve o fluxo de ações a serem feitas para aplicação do teste, pré-requisito que trata das informações que devem estar contidas no sistema antes da aplicação do teste e observação caso seja pertinente que geralmente se refere a algo que deve ser observado no teste ou feito em segundo plano.

Mas por questão de relações entre módulos e classes os testes foram feitos seguindo uma ordem de dependência. Os primeiros testes aplicados foram o do módulo de clientes, assim foi testada a função de cadastro de clientes, a função de pesquisa e edição e por ultimo a função de exclusão, lembrando que em todas as inserções pelo menos dois itens

Caso de uso: Cadastrar Pedido	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Pedido cadastrado com sucesso! , esta mensagem deve aparecer tanto no momento que o usuário clicar em Salvar, quanto ao clicar em Adicionar Produtos. A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Cadastrar Pedido, inserir os dados e confirmar as saídas
Pré Requisito	Ter clientes cadastrados.
Observação	Não há.

Tabela 6 – Caso de Teste: Cadastro de Pedidos

Caso de uso: Cadastrar Itens do Pedido	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Item cadastrado com sucesso! , esta mensagem deve aparecer no momento que o usuário clicar em Adicionar mas possibilitar a inserção de mais itens. A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Cadastrar Pedido, e depois no botão Adicionar Produtos, ou clicando no botão pesquisar pedido e novamente no pedido selecionado para escolher a opção adicionar itens.
Pré Requisito	Ter clientes cadastrados e pedidos, bem como produtos no estoque.
Observação	Deve -se observar se o produto adicionado saiu do estoque.

Tabela 7 – Caso de Teste: Cadastro de Itens do Pedido

foram deixados para que fosse usados em outros fluxos.

O próximo módulo abordado pelos testes foi o de produtos e assim como o de clientes seguiu a ordem de inserção, pesquisa, edição e exclusão. Para este módulo foi respeitada a ordem de inserção: categorias, subcategorias e por fim produtos. Para inserção de categorias, subcategorias e produtos . Para fazer o teste de pesquisa e edição o seguinte ordem foi seguida: categorias, subcategorias, e por fim produtos. Por fim tem se o teste de exclusão e seguindo a mesma ordem primeiramente foi feito com as categorias, com as

subcategorias e finalmente produtos .

O módulo agenda foi tratado após os dois primeiros módulos e seguiu a mesma ordem de testes dos outros. A classe sessão foi testada seguindo a ordem : cadastro de sessão ; pesquisa de sessões ; edição de sessão e exclusão. Da mesma forma a classe compromisso foi testada seguindo a ordem: cadastro de compromissos , pesquisa , edição e por fim exclusão. Lembrando que este módulo se relaciona com o módulo clientes e essa duas classe fazem uso da lista de clientes.

Neste mesmo módulo também tem-se a opção de calendário que nada mais é que uma pesquisa por dia selecionado e de acordo com o caso de teste específico foi possível realizar o teste desta função do sistema.

Logo após os testes com o módulo agenda foi possível realizar os teste do módulo financeiro. De acordo com a ordem de dependência pode se aplicar todos os casos de testes para os casos de uso deste módulo que são: cadastro de metas; pesquisa e edição de metas; exclusão de metas; também pesquisa do estoque e adicionar ao estoque, que são casos de uso relacionados diretamente com os produtos; cadastro, pesquisa, edição e exclusão de pedidos conforme;cadastro de itens do pedido, pesquisa, edição e exclusão de itens; pesquisar produtos em falta e finalmente o caso de uso de resumo financeiro.

Assim foram aplicados todos os testes unitários, seguindo os casos de teste com suas descrições e passos para realização especificados na Seção de Apêndice G. Depois de testado, o sistema foi implantado no celular de algumas consultoras dentre elas 3 diretoras.

O teste de caixa branca foi aplicado usando como base as descrições dos testes unitários e para cada caso de uso foi feito um teste observando todo o fluxo de processo no código fonte usando o *Debug*, ou seja visualizando passo a passo dentro do código o que acontece no sistema.

O teste de caixa preta foi feito com o usuário, por meio da implantação do sistema em alguns celulares, com diferentes versões do Android. Todas as funções do sistema foram testadas pelos usuário, sem seguir uma ordem específica para tal. Para fins avaliativos também foi entregue um questionário para o usuário (Tabela 8), para avaliação do sistema como um todo. As notas seguiram a seguinte premissa : 1 para discordo totalmente, 2 para discordo, 3 para neutro, 4 para concordo e 5 para concordo plenamente. A Tabela 8 está preenchida com a média das notas dadas pelos usuários.

Três diretoras de venda realizaram a avaliação da aplicação gerada, elas tiveram uma semana para testar todas as funções. Após o teste foram distribuídos os questionários e os resultados podem ser vistos também na Tabela 8, sendo resultados satisfatórios.

Afirmação	Nota
Usei a aplicação todos os dias.	3,6
Me acostumei com a aplicação facilmente	4,3
Espero que tenha continuidade.	5
Baixaria a aplicação do Google Play para uso próprio	5
Gostei das telas e do visual da aplicação.	3,3
Não tive dificuldades em usufruir dela.	4,3
Não achei muitos <i>bugs</i> ou falhas no sistema.	4,3
Os ícone de ajuda são bem explicativos.	4,3
Instalaria a nova versão da aplicação caso tivesse	5
Não precisei de ajuda para usar o sistema.	5
Tenho todo o meu estoque cadastrado no sistema.	4,3
Tenho todos os meus pedidos cadastrado no sistema.	4
Todos os meus clientes estão cadastrados no sistema.	4
Indicaria o sistema para outras consultoras.	5

Tabela 8 – Questionário de avaliação final da aplicação desenvolvida: Meu Estoque Rosa

6 Conclusão

O projeto teve início de uma ideia simples, a aplicação das teorias aprendidas durante o curso para construção de um software útil. O objetivo era construir uma aplicação bem estruturada, em uma plataforma muito valorizada no mercado atualmente que é a dos dispositivos móveis, garantindo que fosse de fácil uso e entendimento.

Usando técnicas propostas pelo padrão de projeto chamado Processo Unificado juntamente com as de Modelagem de software aliado a IHC, foi possível construir uma aplicação completa que foi usada como estudo de caso, garantindo sucesso ao trabalho. Alguns contratempos devido a complexidade foram observados e serão explicados a seguir.

Este trabalho foi concluído gerando como resultado um aplicativo útil e que foi testado em ambiente real. O resultado foi plausível e sugere que o software produzido é uma boa aplicação de mercado futuro. Contudo nem tudo foi concluído como o planejado visto que houve um grande esforço para a produção e controle do projeto.

Foram encontradas algumas dificuldades tanto na parte de planejamento e mais na parte de desenvolvimento do software, o que causou a simplificação de algumas funções e a retirada de outras que não se mostraram realmente necessárias no momento. Tudo isso devido a complexidade de se manter a integridade do banco de dados e aplicação na plataforma Android.

Um dificuldade logo encontrada no começo foi manter as camadas bem definidas da aplicação, aliando as próprias camadas já existentes nas aplicações móveis para Android. Houve um tempo para familiarização do desenvolvimento *desktop* e *web* para o desenvolvimento na plataforma móvel.

Outra dificuldade encontrada foi na parte de banco dados, mais precisamente em como manter as tabelas relacionadas por *foreign keys* sem perder o controle ao acesso dos dados e sua integridade, levando ao acoplamento ideal e o uso de chaves auto incrementadas. Assim algumas tabelas como a de produtos tiveram que ser adequadas para que funcionassem de acordo com a especificação técnica.

Houve também um grande empenho para fazer o controle de datas, já que o SQLite não tem suporte ao tipo *Date*, apenas a funções de manipulações de texto para datas. Assim todas as operações feitas com data e hora foram elaboradas com manipulação de *String*, o que ocasionou um trabalho maior de pesquisa e implementação de funções para adequar o sistema e não perder as funcionalidades.

Por ser mais complexa a função de trocas de produtos foi escalada para uma implementação futura assim como as notificações de datas e horários das sessões e com-

promissos, contudo as funções mais importantes foram mantidas. Outra função que foi retirada foram as marcações de sessões e compromissos no calendário, assim só é possível ver as sessões e compromissos marcados em um dia clicando na data desejada.

Pela avaliação final da aplicação junto com o usuário é certo afirmar que aplicação precisa de melhorias na interface gráfica e na parte de inserção de dados. De acordo com as opiniões dadas pelas três usuárias que preencheram o questionário, a aplicação melhoraria se fosse mais dinâmica e tivesse outras alternativas de inserção dos dados. Outra questão levantada é sobre as notificações de eventos como as sessões e os compromissos, no caso seria uma função interessante para uma versão melhorada do sistema.

Uma sugestão interessante para uma nova versão do aplicativo seria aprimorar a inserção de dados usando arquivos disponibilizados pela empresa ou planilhas de pedidos criadas pelo usuário, para melhorar o preenchimento de dados no sistema, tanto para inserção de estoque como para inserção de pedidos dos clientes. Outra sugestão dada foi sobre inserção dos itens no estoque, possibilitando a seleção produtos não só pelo nome deles mas também por categorias e subcategorias.

Para melhorar as notas relacionadas as interfaces poderiam ser implementadas na nova versão interfaces mais dinâmicas e com menus interativos, que possam ser baseadas nos gestos do usuários, como um deslizar do dedo, menus laterais, clicar e arrastar itens entre outros gestos que podem ser amplamente trabalhados na plataforma móvel.

Futuramente pretende-se aprimorar o sistema para agradar ainda mais o usuário, talvez lançar um versão que possua as funcionalidades acima citadas na avaliação final. O resultado final foi satisfatório, pode-se concluir que o uso das teorias aprendidas durante o curso foram importantes para garantir a qualidade no desenvolvimento do produto de software e para que o produto final atingisse os objetivos previstos, de acordo com os requisitos dos usuários.

Referências

- ALIFERI, C. 2014. JavaCodeGeeks. Disponível em: <<http://examples.javacodegeeks.com/android/core/widget/android-calendarview-example/>>. Acesso em: 24 nov. 2014. Citado na página 55.
- BARBOSA, B. S. d. S. S. D. J. *Interação Humano-Computador*. 1. ed. Rio de Janeiro: [s.n.], 2010. Citado 7 vezes nas páginas 6, 22, 23, 24, 25, 26 e 42.
- DANTAS, D. 2014. Blog AndroidPIT. Disponível em: <<http://www.androidpit.com.br/pt/android/market/apps/app/com.dadonas.fluxocaixa/Fluxo-de-Caixa>>. Acesso em: 5 fev. 2014. Citado na página 29.
- DEVELOPERS, S. de suporte A. 2015. Android Developers. Disponível em: <<http://developer.android.com/about/index.html>>. Acesso em: 15 jan. 2015. Citado na página 28.
- FILHO, W. de P. P. *Engenharia de Software: fundamentos, métodos e padrões*. 3. ed. Rio de Janeiro: [s.n.], 2009. Citado 3 vezes nas páginas 22, 27 e 33.
- HIPP, D. R. 2014. SQLite. Disponível em: <<http://www.sqlite.org/foreignkeys.html>>. Acesso em: 15 set. 2014. Citado na página 38.
- INTERNATIONAL ORGANIZATION FOR STANDARTIZATION. *Software engineering - Product quality*. [S.l.], 1991. Citado na página 23.
- INTERNATIONAL ORGANIZATION FOR STANDARTIZATION. *Ergonomic requirements for office work with visual display terminals(VTDs)Part 11:Guidance on Usability*. [S.l.], 1998. Citado na página 23.
- INTUIT, I. 2014. Blog AndroidPIT. Disponível em: <<http://www.androidpit.com.br/pt/android/market/apps/app/com.mint/Mint-com-Personal-Finance>>. Acesso em: 17 fev. 2014. Citado na página 29.
- LARMAN, C. *Utilizando UML e padrões: uma introdução á análise e ao projeto orientado a objetos e ao desenvolvimento iterativo*. 3. ed. Porto Alegre: [s.n.], 2007. Citado 9 vezes nas páginas 16, 17, 18, 19, 20, 31, 32, 33 e 35.
- PACHECO, T. 2014. Blog AndroidPIT. Disponível em: <<http://www.androidpit.com.br/pt/android/market/apps/app/br.thiagopacheco.vendas/Controle-de-Vendas>>. Acesso em: 5 fev. 2014. Citado na página 29.
- PREECE YVONNE ROGERS, H. S. J. *Design de Interação Além da interação homem-computador*. Porto Alegre: [s.n.], 2005. Citado 3 vezes nas páginas 26, 27 e 42.
- SOUZA, M. de. 2014. Blog Escola de Android. Disponível em: <<http://escoladeandroid.blogspot.com.br/2012/02/android-criando-uma-agenda-de-contatos.html>>. Acesso em: 10 set. 2014. Citado 2 vezes nas páginas 28 e 53.

STACKOVERFLOW, C. 2014. Stackoverflow. Disponível em: <<http://stackoverflow.com/questions/1975737/sqlite-datetime-comparison>>. Acesso em: 24 nov. 2014. Citado na página 55.

WIKIPÉDIA. 2015. Wikipédia, enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/Android>>. Acesso em: 15 jan. 2015. Citado na página 27.

WIKIPÉDIA. 2015. Wikipédia, enciclopédia livre. Disponível em: <http://pt.wikipedia.org/wiki/Histórico_de_versions_do_Android>. Acesso em: 15 jan. 2015. Citado na página 27.

Apêndices

APÊNDICE A – Regras de Negócio

Para complemento do projeto definimos aqui algumas regras de negócio que devem ser aplicadas ao sistema, por se tratar de um sistema simples, logo o número de regras é menor.

Relativo aos Clientes: Quanto ao cadastro, é importante manter a informação da data de nascimento, uma vez que neste ramo de negócio é importante presentear o cliente no aniversário ou no mínimo dar-lhe os parabéns, por isso deve ser armazenado a data e verificada se é válida.

É importante também manter o campo de Observações para o cliente, caso seja necessário guardar alguma informação pertinente ao cliente que seja relevante para a consultora.

Sobre os Produtos: Os produtos são cadastrados de acordo com uma categoria e talvez uma subcategoria, portanto para manter a organização deve-se manter o cadastro de produtos de acordo com a categoria e subcategoria, os mesmos podem ser obtido no site (colocar referencia).

Agenda: De fato o item mais importante da agenda será a marcação de sessões que são o melhor jeito de conquistar novas clientes e também além de demonstrar os produtos, vender. Portanto é preciso ter uma função de marcação de sessão com data, hora e relacionar ao novo cliente, é também importante informar com um dia de antecedência sobre a sessão que irá acontecer.

As reuniões de unidade são importantes também, logo é preciso ter uma função para marcar as reuniões, bem como eventos relacionados.

Compromissos como entrega de produtos ou algum acontecimento relacionado ao negócio também devem ser agendados.

Financeiro: O módulo financeiro será capaz de fazer o controle tanto do estoque de produtos como também dos pedidos e resumo de ganhos, assim deverá controlar quantos e quais produtos foram vendidos em cada pedido, dando baixa no estoque. Este módulo deverá contar com um controle de metas cadastradas pelo usuário e o resumo demonstrará os ganhos com relação as metas mostrando quanto falta ou quanto ultrapassou a meta cadastrada para aquele período do ano, para o mês e para o dia.

Cada pedido excluído deverá retornar a quantia de produtos no estoque, bem como cada item do pedido que for excluído. Mas mesmo que os produtos sejam excluídos os pedido continuam inalterados, uma vez que são parte do histórico de pedidos da consultora.

No resumo além dos ganhos do dia, mês e trimestre, deverá constar o número de sessões e compromisso marcados no mês, além disso o número de produtos no estoque e o valor em reais.

APÊNDICE B – Documento de Arquitetura do Software

O projeto foi estruturado para manter as funções básicas de um fluxo de caixa e de uma agenda, contando com suas particularidades de acordo com a regra de negócio, baseando nas regras de negócio especificadas no documento anterior estimamos os requisitos funcionais a seguir, separados por módulos.

B.1 Cliente

Para o módulo Cliente definimos os seguintes requisitos:

- **Cadastro de Clientes** É necessário que o sistema realize o cadastro do cliente informando o nome, telefone, endereço, data de nascimento e observações, assim que cadastrado retornar mensagem para o usuário de sucesso ou fracasso. Ao cancelar o cadastro também deve-se informar ao usuário que o cadastro foi cancelado, em ambos casos deve se retorna a tela de opções do cliente.
- **Pesquisa de Clientes com edição e exclusão** Nessa opção o sistema retornará a lista de clientes e um campo para pesquisa por nome, nesta tela também ao selecionar um cliente será fornecida a opção para o usuário de editar ou excluir o cliente. Ao selecionar a opção de editar deve se direcionar para uma tela de edição com os campos preenchidos idêntica ao do cadastro. Ao selecionar a opção excluir deve informar ao usuário da ação, se o mesmo aceitar deve-se excluir e retorna mensagem de sucesso, também retornar a tela de opções do cliente.

B.2 Produtos

Para o módulo Produtos definimos os seguintes requisitos:

- **Cadastro de Categoria** Na tela de produtos ao clicar na opção de cadastrar o sistema redirecionará para outra tela que terá a opção de cadastro de Categoria, a de Sub Categoria e a de Produtos, no cadastro de Categoria será salvo apenas o nome da categoria, informando ao usuário a mensagem de sucesso ou não.
- **Pesquisa de Categoria com edição e exclusão** A pesquisa da categoria será da mesma forma que a do cliente com os mesmos campos de pesquisa por nome

da categoria e ao clicar em um item da lista terá as opções de editar e excluir. Da mesma forma também retornará a tela de opções de cadastro.

- **Cadastro de Sub Categoria** Assim como a categoria, será cadastrada a sub categoria, porém na sub categoria haverá um campo a mais do tipo de lista pra selecionar a qual categoria pertence a sub categoria em questão.
- **Pesquisa de Sub Categoria com edição e exclusão** Idem a pesquisa de categoria.
- **Cadastro de Produtos** Proceder como nos outros cadastros, contudo o usuário deverá informar a descrição do produto, selecionar a categoria e a sub categoria se houver, e o preço do mesmo em reais.
- **Pesquisa de Produtos com edição e exclusão** Idem as outras pesquisas.

B.3 Agenda

Para o módulo Agenda definimos os seguintes requisitos:

- **Cadastro de Sessão** Na tela de sessão ao clicar na opção de cadastrar o sistema redirecionará para tela de cadastro contendo os campos Data, Hora, Obs. e Nomes dos clientes cadastrados, logo ao preencher corretamente os campos o sistema informará ao usuário sobre o sucesso do cadastro e a informação de quantidade de dias para aquela sessão. O sistema deverá fazer validação de data e hora verificando se já existe cadastrado uma sessão com mesma data e hora.
- **Pesquisa de Sessão com edição e exclusão** A pesquisa da sessão será feita clicando no botão pesquisar, o sistema redirecionará para tela de pesquisa já preenchida com uma lista de sessões marcadas desde o dia vigente até a ultima sessão marcada, desconsiderando as sessões em que a data e a hora ja passaram, na mesma tela também poderá fazer a pesquisa por intervalo de data. Ao clicar em qualquer item da lista demonstrada na pesquisa, o sistema abrirá um menu para edição ou exclusão do item clicado.
- **Cadastro de Compromisso** Parecido com o cadastro de sessão contudo por ser genérico conterá os campos Descrição e Prioridade, além de o campo de Nomes de cliente poderá ou não ser preenchido caso seja uma entrega de produtos. A prioridade deverá ser preenchida com baixa, média ou alta de acordo com a seleção do cliente e as sessões terão prioridade alta. O sistema deverá fazer validação de data e hora verificando se já existe cadastrado um compromisso com mesma data e hora.

- **Pesquisa de Sub Categoria com edição e exclusão** Idem a pesquisa de compromisso.
- **Calendário** Ao clicar no botão Calendário, o sistema redirecionará para uma tela contendo o calendário já marcando o dia vigente e também uma lista com todas as sessões e compromissos desse dia. Caso o usuário clique em qualquer dia a lista mudará para os compromissos e sessões marcados no dia selecionado.

B.4 Financeiro

Para o módulo Financeiro definimos os seguintes requisitos:

- **Pesquisar Estoque** Ao clicar no botão Pesquisar Estoque da tela financeira o sistema deve apresentar uma lista de todos os produtos no estoque bem como a quantidade estocada, possibilitando ao clicar em um item da lista a edição do mesmo. Nesta tela deve também ser possível filtrar os itens por nome de produto.
- **Adicionar ao Estoque** Ao clicar no botão Adicionar ao Estoque o sistema deve redirecionar para a tela de alteração de estoque com os devidos campos de categoria, subcategoria e produtos preenchidos em uma lista para o usuários escolher, os campos de adicionar ou remover para preenchimento de quantidade pelo usuário. Ao clicar em adicionar o sistema deve salvar retornando mensagem de sucesso ou insucesso ao usuário, bem como mensagens de erro no preenchimento.
- **Cadastro de Meta** Na tela de meta ao clicar na opção de cadastrar o sistema redirecionará para outra tela qua terá a opção de cadastro, será salvo a meta diária, mensal e trimestral, o dia do cadastro e o período daquela meta, informando ao usuário a mensagem de sucesso ou não.
- **Pesquisa de Meta com edição e exclusão** A pesquisa da meta será da mesma forma que a do cliente com os mesmos campos de pesquisa por intervalo de datas e ao clicar em um item da lista terá as opções de editar e excluir. O termino das ações retornará a tela de pesquisa.
- **Cadastro de Pedido e Itens do pedido** Clicando na opção cadastrar pedido da tela financeira, o sistema redirecionará para a tela de cadastro de pedido com o campo nome do cliente selecionável e data do pedido para preenchimento e os campos valor total, número de parcelas e valor da parcela para visualização preenchidos com zero. Ao clicar no botão salvar o sistema deve salvar apresentando as mensagens corretas e ao clicar no botão adicionar produtos o sistema deve salvar o pedido, mostrar a mensagem de sucesso e redirecionar para tela de cadastro de itens do

pedido mostrando apenas os produtos que estão no estoque para seleção. Ao salvar o item o sistema continua na tela de cadastro com os dados atualizados.

- **Pesquisa de Pedido e itens com edição e exclusão** Idem as outras pesquisas, contudo possibilitando, ao clicar na lista de pedidos, o redirecionamento para pesquisa de itens. Na exclusão os itens voltarão para o estoque.
- **Pesquisa de Produtos em falta** Idem a pesquisa de estoque contudo apresentará apenas o estoque com quantidade igual a zero.
- **Resumo** Apresentar ao usuário um resumo financeiro com o ganho diário, mensal e trimestral levando em consideração o período e a meta cadastrada no mesmo, também é importante ter um controle de quantos produtos há no estoque, bem como o valor total em reais do mesmo.

No resumo também deverá aparecer quantas sessões foram marcadas no mês bem como o número de compromissos, o número de clientes cadastrados também é importante.

APÊNDICE C – Modelo de Casos de Uso

De acordo com o Documento de Arquitetura do Software e das Regras de Negócio podemos então definir o Modelo de Casos de Uso, a seguir teremos a descrição dos casos de uso divididos por módulos e requisitos.

C.1 Módulo Cliente

- **Cadastrar um Cliente**

Escopo: O sistema deverá fazer o cadastro de clientes de acordo com os dados informados pelo usuário.

Pré-Condição: Não há.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso o cliente foi ou não cadastrado, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de clientes cadastrados.

Cenário de sucesso Principal: 1. O usuário clica no botão cadastrar da tela de cliente.

2. O sistema então redireciona para tela de cadastro com os campos para preenchimento : Nome, Telefone, Endereço, Data de Nascimento e Obs.

3. O usuário preenche os campos e clica no botão aceitar caso queira salvar o cadastro.

3.1. O sistema verifica os campos e os dados preenchidos e insere no banco retornando a mensagem de sucesso para o usuário.

3.2. O sistema redireciona para a tela de cliente.

4. O usuário preenche os campos mas clica no botão cancelar.

4.1 O sistema retorna mensagem de cadastro cancelado e redireciona para a tela de cliente

- **Pesquisar Clientes**

Escopo: O sistema deverá fazer a pesquisa de clientes, apresentando a lista dos clientes cadastrados e possibilitando a pesquisa por nome.

Pré-Condição: Ter clientes cadastrados.

Garantia de sucesso: Apresentar ao usuário a lista de clientes cadastrados e os clientes resultantes caso o usuário fizer uma pesquisa.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de cliente.

2. O sistema então redireciona para tela de pesquisa já preenchida com a lista e um campo para preenchimento.

3. O usuário preenche os campo no botão pesquisar caso queira escolher um ou mais clientes com um nome específico.

3.1. O sistema faz uma consulta e retorna a lista de clientes achados que correspondem ao nome inserido pelo usuário.

3.2. O sistema redireciona para a tela de cliente.

4. O usuário visualiza a lista e clica em voltar.

4.1 O sistema retorna para a tela de cliente.

- **Editar um Cliente**

Escopo: O sistema deverá fazer a edição de clientes de acordo com os dados informados pelo usuário e seleção do cliente na lista demonstrada na pesquisa.

Pré-Condição: Ter o cliente já cadastrado no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso o cliente foi ou não editado, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de clientes cadastrados atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de cliente.

2. O sistema então redireciona para tela de pesquisa .

3. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos clientes facilitando a seleção.

3.1 O sistema verifica o banco e retorna a lista resultante da consulta por nome.

3.2. O usuário seleciona o cliente clicando na lista.

3.3. O sistema mostra um menu com as opções Editar e Excluir .

3.4. O usuário seleciona a opção editar.

3.5. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.

3.6. O usuário faz as alterações necessárias e clica em aceitar.

3.7. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Cliente.

4. O usuário clica em um cliente da lista .

- 4.1. O sistema mostra um menu com as opções Editar e Excluir .
- 4.2. O usuário seleciona a opção editar.
- 4.3. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.
- 4.4. O usuário faz as alterações necessárias e clica em aceitar.
- 4.5. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Cliente.

● Excluir um Cliente

Escopo: O sistema deverá fazer a exclusão de clientes de acordo com os dados informados pelo usuário e seleção do cliente na lista demonstrada na pesquisa .

Pré-Condição: Ter o cliente já cadastrado no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso o cliente foi ou não excluído. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de clientes cadastrados atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de cliente.

2. O sistema então redireciona para tela de pesquisa .

3. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos clientes facilitando a seleção.

3.1. O sistema verifica o banco e retorna a lista resultante da consulta por nome.

3.2. O usuário seleciona o cliente clicando na lista.

3.3. O sistema mostra um menu com as opções Editar e Excluir .

3.4. O usuário seleciona a opção excluir.

3.5. O sistema informa o usuário da ação com dois botões com sim e não.

3.5.1. O usuário clica no botão sim.

3.5.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

3.6. O usuário clica no botão não. 3.6.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

4. O usuário seleciona o cliente clicando na lista mostrada.

4.1. O sistema mostra um menu com as opções Editar e Excluir .

4.2. O usuário seleciona a opção excluir.

4.3. O sistema informa o usuário da ação com dois botões com sim e não.

4.3.1. O usuário clica no botão sim.

4.3.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

4.4. O usuário clica no botão não.

4.4.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

C.2 Módulo Produto

• Cadastrar um Produto

Escopo: O sistema deverá fazer o cadastro de produtos de acordo com os dados informados pelo usuário.

Pré-Condição: Ter uma categoria cadastrada.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso o produto foi ou não cadastrado, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de produtos cadastrados.

Cenário de sucesso Principal: 1. O usuário clica no botão cadastrar da tela de produto.

2. O sistema então redireciona para tela de opções para cadastro, contendo os botões para cadastro de produtos, categorias e subcategorias.

3. O usuário clica no botão produtos.

4. O sistema redireciona para a tela de cadastro de produtos com os campos: Nome Produto, Descrição Produto, Categoria, Subcategoria e Preço Produto.

5. O usuário preenche os campos e clica no botão aceitar caso queira salvar o cadastro.

5.1. O sistema verifica os campos e os dados preenchidos e insere no banco retornando a mensagem de sucesso para o usuário.

5.2. O sistema redireciona para a tela de opções de cadastro.

6. O usuário preenche os campos mas clica no botão cancelar.

6.1 O sistema retorna mensagem de cadastro cancelado e redireciona para a tela de opções de cadastro.

• Pesquisar Produtos

Escopo: O sistema deverá fazer a pesquisa de produtos, apresentando a lista dos produtos cadastrados e possibilitando a pesquisa por nome.

Pré-Condição: Ter produtos cadastrados.

Garantia de sucesso: Apresentar ao usuário a lista de produtos cadastrados e os produtos resultantes caso o usuário fizer uma pesquisa.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de Produto.

2. O sistema então redireciona para tela de opções para pesquisa, contendo os botões para pesquisa de produtos, categorias e subcategoria

3. O usuário clica no botão produtos.

4. O sistema então redireciona para tela de pesquisa já preenchida com a lista e um campo para preenchimento.

5. O usuário preenche os campo no botão pesquisar caso queira escolher um ou mais produtos com um nome específico.

5.1. O sistema faz uma consulta e retorna a lista de produtos achados que correspondem ao nome inserido pelo usuário.

5.2. O sistema redireciona para a tela de opções de pesquisa.

6. O usuário visualiza a lista e clica em voltar.

6.1 O sistema retorna para a tela de opções de pesquisa.

● Editar um produto

Escopo: O sistema deverá fazer a edição de produtos de acordo com os dados informados pelo usuário.

Pré-Condição: Ter o produto já cadastrado no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso o produto foi ou não editado, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de produtos cadastrados atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de produto.

2. O sistema então redireciona para tela de opções para pesquisa, contendo os botões para pesquisa de produtos, categorias e subcategorias.

3. O usuário clica no botão produtos.

4. O sistema então redireciona para tela de pesquisa .

5. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos produtos facilitando a seleção.

5.1 O sistema verifica o banco e retorna a lista resultante da consulta por nome.

5.2. O usuário seleciona o produto clicando na lista.

- 5.3. O sistema mostra um menu com as opções Editar e Excluir .
- 5.4. O usuário seleciona a opção editar.
- 5.5. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.
- 5.6. O usuário faz as alterações necessárias e clica em aceitar.
- 5.7. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Pesquisa de produtos.
- 6. O usuário clica em um produto da lista .
- 6.1. O sistema mostra um menu com as opções Editar e Excluir .
- 6.2. O usuário seleciona a opção editar.
- 6.3. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.
- 6.4. O usuário faz as alterações necessárias e clica em aceitar.
- 6.5. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Pesquisa de produtos .

- **Excluir um Produto**

Escopo: O sistema deverá fazer a exclusão de produtos de acordo com os dados informados pelo usuário e seleção do produto na lista demonstrada na pesquisa .

Pré-Condição: Ter o produto já cadastrado no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso o produto foi ou não excluído. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de produtos cadastrados atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de Produtos.

- 2. O sistema então redireciona para tela de opções de pesquisa .
- 3. O usuário seleciona das três opções, a de produtos.
- 4. O sistema então redireciona para tela de pesquisa de produtos .
- 5. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos produtos facilitando a seleção ou apenas.
- 5.1. O sistema verifica o banco e retorna a lista resultante da consulta por nome.
- 5.2. O usuário seleciona o produto clicando na lista.
- 5.3. O sistema mostra um menu com as opções Editar e Excluir .
- 5.4. O usuário seleciona a opção excluir.
- 5.5. O sistema informa o usuário da ação com dois botões com sim e não.
- 5.5.1. O usuário clica no botão sim.

5.5.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

5.6. O usuário clica no botão não.

5.6.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

6. O usuário clica em um produto da lista mostrada.

6.1. O sistema mostra um menu com as opções Editar e Excluir .

6.2. O usuário seleciona a opção excluir.

6.3. O sistema informa o usuário da ação com dois botões com sim e não.

6.3.1. O usuário clica no botão sim.

6.3.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

5.4. O usuário clica no botão não.

6.4.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

- **Cadastrar uma Categoria**

Escopo: O sistema deverá fazer o cadastro de categoria de acordo com os dados informados pelo usuário.

Pré-Condição: Não há.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a categoria foi ou não cadastrada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de categorias cadastradas.

Cenário de sucesso Principal: 1. O usuário clica no botão cadastrar da tela de produto.

2. O sistema então redireciona para tela de opções para cadastro, contendo os botões para cadastro de produto, categoria e subcategoria.

3. O usuário clica no botão categoria.

4. O sistema redireciona para a tela de cadastro de categoria com os campos Nome Categoria.

5. O usuário preenche o campo e clica no botão aceitar caso queira salvar o cadastro.

5.1. O sistema verifica os campos e o dado preenchido e insere no banco retornando a mensagem de sucesso para o usuário.

5.2. O sistema redireciona para a tela de opções de cadastro.

6. O usuário preenche os campo mas clica no botão cancelar.

6.1 O sistema retorna mensagem de cadastro cancelado e redireciona para a tela de opções de cadastro.

- **Pesquisar Categorias**

Escopo: O sistema deverá fazer a pesquisa de categorias, apresentando a lista dos categorias cadastradas e possibilitando a pesquisa por nome.

Pré-Condição: Ter categorias cadastradas.

Garantia de sucesso: Apresentar ao usuário a lista de categorias cadastradas ou as categorias resultantes caso o usuário fizer uma pesquisa.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de Produto.

2. O sistema então redireciona para tela de opções para pesquisa, contendo os botões para pesquisa de produtos, categorias e subcategorias.

3. O usuário clica no botão categoria.

4. O sistema então redireciona para tela de pesquisa já preenchida com a lista e um campo para preenchimento.

5. O usuário preenche os campo no botão pesquisar caso queira escolher um ou mais categorias com um nome específico.

5.1. O sistema faz uma consulta e retorna a lista de categorias achadas que correspondem ao nome inserido pelo usuário.

5.2. O sistema redireciona para a tela de opções de pesquisa.

6. O usuário visualiza a lista e clica em voltar.

6.1 O sistema retorna para a tela de opções de pesquisa.

- **Editar uma categoria**

Escopo: O sistema deverá fazer a edição de categorias de acordo com os dados informados pelo usuário.

Pré-Condição: Ter o categoria já cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a categoria foi ou não editada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de categorias cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de produto.

2. O sistema então redireciona para tela de opções para pesquisa, contendo os botões para pesquisa de produtos, categorias e subcategorias.
3. O usuário clica no botão categoria.
4. O sistema então redireciona para tela de pesquisa .
5. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos categorias facilitando a seleção.
 - 5.1 O sistema verifica o banco e retorna a lista resultante da consulta por nome.
 - 5.2. O usuário seleciona o produto clicando na lista.
 - 5.3. O sistema mostra um menu com as opções Editar e Excluir .
 - 5.4. O usuário seleciona a opção editar.
 - 5.5. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.
 - 5.6. O usuário faz as alterações necessárias e clica em aceitar.
 - 5.7. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Pesquisa de categorias.
6. O usuário clica em um produto da lista .
 - 6.1. O sistema mostra um menu com as opções Editar e Excluir .
 - 6.2. O usuário seleciona a opção editar.
 - 6.3. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.
 - 6.4. O usuário faz as alterações necessárias e clica em aceitar.
 - 6.5. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Pesquisa de categorias.

- **Excluir uma categoria**

Escopo: O sistema deverá fazer a exclusão de categorias de acordo com os dados informados pelo usuário e seleção da categoria na lista demonstrada na pesquisa .

Pré-Condição: Ter a categoria já cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a categoria foi ou não excluída. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de categorias cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de Produtos.

2. O sistema então redireciona para tela de opções de pesquisa .
3. O usuário seleciona das três opções, a de categorias.
4. O sistema então redireciona para tela de pesquisa de categorias .

5. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos produtos facilitando a seleção ou apenas.

5.1. O sistema verifica o banco e retorna a lista resultante da consulta por nome.

5.2. O usuário seleciona o produto clicando na lista.

5.3. O sistema mostra um menu com as opções Editar e Excluir .

5.4. O usuário seleciona a opção excluir.

5.5. O sistema informa o usuário da ação com dois botões com sim e não.

5.5.1. O usuário clica no botão sim.

5.5.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

5.6. O usuário clica no botão não.

5.6.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

6. O usuário clica em um produto da lista mostrada.

6.1. O sistema mostra um menu com as opções Editar e Excluir .

6.2. O usuário seleciona a opção excluir.

6.3. O sistema informa o usuário da ação com dois botões com sim e não.

6.3.1. O usuário clica no botão sim.

6.3.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

6.4. O usuário clica no botão não.

6.4.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

● **Cadastrar uma Subcategoria**

Escopo: O sistema deverá fazer o cadastro de subcategoria de acordo com os dados informados pelo usuário.

Pré-Condição: Ter categorias cadastradas.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a subcategoria foi ou não cadastrada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de subcategorias cadastradas.

Cenário de sucesso Principal: 1. O usuário clica no botão cadastrar da tela de produto.

2. O sistema então redireciona para tela de opções para cadastro, contendo os botões para cadastro de produto, categoria e subcategoria.
3. O usuário clica no botão subcategoria.
4. O sistema redireciona para a tela de cadastro de subcategoria com os campos Nome Categoria e Nome Subcategoria.
5. O usuário preenche o campo e clica no botão aceitar caso queira salvar o cadastro.
- 5.1. O sistema verifica os campos e o dado preenchido e insere no banco retornando a mensagem de sucesso para o usuário.
- 5.2. O sistema redireciona para a tela de opções de cadastro.
6. O usuário preenche os campos mas clica no botão cancelar.
- 6.1. O sistema retorna mensagem de cadastro cancelado e redireciona para a tela de opções de cadastro.

- **Pesquisar Subcategorias**

Escopo: O sistema deverá fazer a pesquisa de subcategorias, apresentando a lista dos subcategorias cadastradas e possibilitando a pesquisa por nome.

Pré-Condição: Ter subcategorias cadastradas.

Garantia de sucesso: Apresentar ao usuário a lista de subcategorias cadastradas ou as subcategorias resultantes caso o usuário fizer uma pesquisa.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de Produto.

2. O sistema então redireciona para tela de opções para pesquisa, contendo os botões para pesquisa de produtos, categorias e subcategorias.
3. O usuário clica no botão subcategoria.
4. O sistema então redireciona para tela de pesquisa já preenchida com a lista e um campo para preenchimento.
5. O usuário preenche os campos no botão pesquisar caso queira escolher um ou mais categorias com um nome específico.
- 5.1. O sistema faz uma consulta e retorna a lista de categorias achadas que correspondem ao nome inserido pelo usuário.
- 5.2. O sistema redireciona para a tela de opções de pesquisa.
6. O usuário visualiza a lista e clica em voltar.
- 6.1. O sistema retorna para a tela de opções de pesquisa.

- **Editar uma subcategoria**

Escopo: O sistema deverá fazer a edição de categorias de acordo com os dados informados pelo usuário.

Pré-Condição: Ter a subcategoria já cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a subcategoria foi ou não editada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de subcategorias cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de produto.

2. O sistema então redireciona para tela de opções para pesquisa, contendo os botões para pesquisa de produtos, categorias e subcategorias.

3. O usuário clica no botão subcategoria.

4. O sistema então redireciona para tela de pesquisa .

5. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos subcategorias facilitando a seleção.

5.1 O sistema verifica o banco e retorna a lista resultante da consulta por nome.

5.2. O usuário seleciona a subcategoria clicando na lista.

5.3. O sistema mostra um menu com as opções Editar e Excluir .

5.4. O usuário seleciona a opção editar.

5.5. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.

5.6. O usuário faz as alterações necessárias e clica em aceitar.

5.7. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Pesquisa de subcategorias.

6. O usuário clica em um produto da lista .

6.1. O sistema mostra um menu com as opções Editar e Excluir .

6.2. O usuário seleciona a opção editar.

6.3. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.

6.4. O usuário faz as alterações necessárias e clica em aceitar.

6.5. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Pesquisa de subcategorias.

- **Excluir uma subcategoria**

Escopo: O sistema deverá fazer a exclusão de subcategorias de acordo com os dados informados pelo usuário e seleção da subcategoria na lista demonstrada na pesquisa .

Pré-Condição: Ter a subcategoria já cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a subcategoria foi ou não excluída. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de subcategorias cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de Produtos.

2. O sistema então redireciona para tela de opções de pesquisa .

3. O usuário seleciona das três opções, a de subcategorias.

4. O sistema então redireciona para tela de pesquisa de subcategorias .

5. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de menos subcategorias facilitando a seleção ou apenas. 5.1. O sistema verifica o banco e retorna a lista resultante da consulta por nome.

5.2. O usuário seleciona a subcategoria clicando na lista.

5.3. O sistema mostra um menu com as opções Editar e Excluir .

5.4. O usuário seleciona a opção excluir.

5.5. O sistema informa o usuário da ação com dois botões com sim e não.

5.5.1. O usuário clica no botão sim.

5.5.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

5.6. O usuário clica no botão não.

5.6.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

6. O usuário clica em uma subcategoria da lista mostrada. 6.1. O sistema mostra um menu com as opções Editar e Excluir .

6.2. O usuário seleciona a opção excluir.

6.3. O sistema informa o usuário da ação com dois botões com sim e não.

6.3.1. O usuário clica no botão sim.

6.3.2. O sistema então exclui o cliente mostrando a mensagem de sucesso e atualiza a lista.

5.4. O usuário clica no botão não.

6.4.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

C.3 Módulo Agenda

- **Cadastrar uma Sessão**

Escopo: O sistema deverá fazer o cadastro de sessões de acordo com os dados informados pelo usuário.

Pré-Condição: Não há.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a sessão foi ou não cadastrada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de sessões cadastradas.

Cenário de sucesso Principal: 1. O usuário clica no botão cadastrar da tela de sessão.

2. O sistema então redireciona para tela de cadastro com os campos para preenchimento : Data, Hora, Cliente e Obs.

3. O usuário preenche os campos e clica no botão aceitar caso queira salvar o cadastro. 3.1. O sistema verifica os campos e os dados preenchidos e insere no banco retornando a mensagem de sucesso para o usuário.

3.2. O sistema redireciona para a tela de sessão.

4. O usuário preenche os campos mas clica no botão cancelar.

4.1 O sistema retorna mensagem de cadastro cancelado e redireciona para a tela de sessão.

- **Pesquisar Sessões**

Escopo: O sistema deverá fazer a pesquisa de sessões, apresentando a lista dos sessões cadastrados e possibilitando a pesquisa por data.

Pré-Condição: Ter sessões cadastrados.

Garantia de sucesso: Apresentar ao usuário a lista de sessões cadastradas e as sessões resultantes caso o usuário fizer uma pesquisa.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de sessão.

2. O sistema então redireciona para tela de pesquisa já preenchida com a lista e um campo para preenchimento.

3. O usuário preenche os campo no botão pesquisar caso queira escolher um ou mais sessões em um intervalo de datas específico. 3.1. O sistema faz uma consulta e retorna a lista de sessões achadas que correspondem ao intervalo de data inserido pelo usuário.

3.2. O sistema redireciona para a tela de sessão.

4. O usuário visualiza a lista e clica em voltar.

4.1 O sistema retorna para a tela de sessão.

- **Editar uma sessão**

Escopo: O sistema deverá fazer a edição de sessões de acordo com os dados informados pelo usuário e seleção da sessão na lista demonstrada na pesquisa.

Pré-Condição: Ter a sessão já cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a sessão foi ou não editada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de sessões cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de sessão.

2. O sistema então redireciona para tela de pesquisa .

3. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de sessões em uma data específica facilitando a seleção. 3.1 O sistema verifica o banco e retorna a lista resultante da consulta por data.

3.2. O usuário seleciona a sessão clicando na lista.

3.3. O sistema mostra um menu com as opções Editar e Excluir .

3.4. O usuário seleciona a opção editar.

3.5. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.

3.6. O usuário faz as alterações necessárias e clica em aceitar.

3.7. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de sessão.

4. O usuário clica em uma sessão da lista .

4.1. O sistema mostra um menu com as opções Editar e Excluir .

4.2. O usuário seleciona a opção editar.

4.3. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.

4.4. O usuário faz as alterações necessárias e clica em aceitar.

4.5. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de pesquisa.

- **Excluir uma sessão**

Escopo: O sistema deverá fazer a exclusão de sessões de acordo com os dados informados pelo usuário e seleção do sessão na lista demonstrada na pesquisa .

Pré-Condição: Ter a sessão já cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a sessão foi ou não excluída. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de sessões cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de sessão.

2. O sistema então redireciona para tela de pesquisa .

3. O usuário preenche os campos e clica no botão pesquisar caso queira o retorno de sessões em um intervalo de data específica facilitando a seleção. 3.1. O sistema verifica o banco e retorna a lista resultante da consulta por data.

3.2. O usuário seleciona a sessão clicando na lista.

3.3. O sistema mostra um menu com as opções Editar e Excluir .

3.4. O usuário seleciona a opção excluir.

3.5. O sistema informa o usuário da ação com dois botões com sim e não.

3.5.1. O usuário clica no botão sim.

3.5.2. O sistema então exclui a sessão mostrando a mensagem de sucesso e atualiza a lista.

3.6. O usuário clica no botão não. 3.6.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

4. O usuário seleciona a sessão clicando na lista mostrada.

4.1. O sistema mostra um menu com as opções Editar e Excluir .

4.2. O usuário seleciona a opção excluir.

4.3. O sistema informa o usuário da ação com dois botões com sim e não.

4.3.1. O usuário clica no botão sim.

4.3.2. O sistema então exclui a sessão mostrando a mensagem de sucesso e atualiza a lista.

4.4. O usuário clica no botão não.

4.4.1. O sistema então mostra mensagem de cancelamento e volta para a tela de pesquisa normal.

- **Visualizar Calendário**

Escopo: O sistema deverá mostrar um calendário e ao clicar em um dia preencher duas listas, caso houver itens cadastrados na data escolhida, uma para sessões e outra para compromissos .

Pré-Condição: Não há.

Garantia de sucesso: Apresentar ao usuário a lista de sessões relativas a data escolhida bem como a de compromissos.

Cenário de sucesso Principal: 1. O usuário clica no botão calendário da tela de Agenda.

2. O sistema então redireciona para tela do calendário com as listas já preenchidas com o dia marcado.

3. O usuário clica em uma data específica no calendário. 4. O sistema faz uma consulta e retorna a lista de sessões e compromisso que correspondem a data escolhida pelo usuário.

5. O usuário visualiza a lista e clica em voltar.

6. O sistema retorna para a tela de Agenda.

C.4 Módulo Financeiro

-

- **Editar pedido e itens**

Escopo: O sistema deverá editar o pedido e também os itens do pedido podendo ser editado separadamente em dois fluxos de atividades.

Pré-Condição: Pedido cadastrado no sistema.

Garantia de sucesso: Caso o pedido cadastrado já contenha itens, o sistema dará ao usuário a opção de somente editar o número de parcelas do pedido, caso contrário não há edição para pedido, mas poderá adicionar itens ao mesmo.

Cenário de sucesso Principal:

Fluxo principal : Edição do Pedido

1. O usuário clica no botão visualizar pedidos da tela financeira.

2. O sistema redireciona para a tela de pesquisa de pedido com a lista de pedidos e um campo para pesquisa por Nome do cliente.

3. O usuário escolhe o pedido da lista. 4. O sistema mostra as opções para os pedidos.

5. O usuário seleciona a opção editar pedido.

- 6.O sistema redireciona para a tela de alteração, caso tenha itens no pedido, o sistema disponibiliza a edição do campo numero de parcelas.
7. O usuário preenche o campo e se quiser ver o valor de cada parcela pode clicar em calcular e o sistema mostra o valor.
8. O usuário clica em salvar.
- 9.O sistema salva a alteração e mostra mensagem de sucesso, retornando a tela financeira.

Fluxo secundário : Edição de Itens do Pedido

- 1.O usuário clica em um pedido da lista de pesquisa.
- 2.O sistema mostra as opções para aquele pedido.
3. O usuário clica na opção conferir itens adicionados.
- 4.O sistema redireciona para a tela de pesquisa de itens com a lista de itens adicionados no pedido.
5. O usuário clica em um item e seleciona a opção editar.
6. O sistema redireciona para tela de edição habilitando apenas o campo de quantidade.
7. O usuário preenche e manda salvar.
8. O sistema salva e redireciona para a tela de pesquisa novamente, mostrando a mensagem de sucesso.

● Excluir pedido e itens

Escopo: O sistema deverá excluir o pedido e também os itens do pedido .

Pré-Condição:Pedido cadastrado no sistema.

Garantia de sucesso:Tanto o pedido quanto os itens poderão ser excluídos e o valor do item diminuído do pedido, bem como restaurado no estoque.

Cenário de sucesso Principal:

Fluxo principal : Exclusão do Pedido

1. O usuário clica no botão visualizar pedidos da tela financeira.
- 2.O sistema redireciona para a tela de pesquisa de pedido com a lista de pedidos e um campo para pesquisa por Nome do cliente.
3. O usuário escolhe o pedido da lista. 4. O sistema mostra as opções para os pedidos.
5. O usuário seleciona a opção excluir pedido.
- 6.O sistema mostra uma mensagem de confirmação e alerta para a exclusão dos itens e o retorno para o estoque.

7. O usuário confirma a exclusão.
8. O sistema exclui o pedido e atualiza a lista de pesquisa.

Fluxo secundário : Exclusão de Itens do Pedido

- 1.O usuário clica em um pedido da lista de pesquisa.
- 2.O sistema mostra as opções para aquele pedido.
3. O usuário clica na opção conferir itens adicionados.
- 4.O sistema redireciona para a tela de pesquisa de itens com a lista de itens adicionados no pedido.
5. O usuário clica em um item e seleciona a opção excluir.
- 6.O sistema mostra uma mensagem de confirmação e alerta para a exclusão do item e o retorno para o estoque.
7. O usuário confirma a exclusão.
8. O sistema exclui o item e atualiza a lista de itens, o valor do pedido e se houver, o valor da parcela.

● **Pesquisar Estoque**

Escopo: O sistema deverá fazer a pesquisa de produtos no estoque, apresentando a lista dos produtos no estoque e possibilitando a pesquisa por nome do produto.

Pré-Condição:Ter produtos cadastrados.

Garantia de sucesso:Apresentar ao usuário a lista de todo o estoque.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar estoque .

2. O sistema então redireciona para tela de pesquisa já preenchida com a lista e um campo para preenchimento.
3. O usuário preenche os campo no botão pesquisar caso queira escolher um ou mais produtos com um nome específico.
4. O usuário visualiza a lista e clica em voltar.
5. O sistema retorna para a tela de estoque.

● **Editar estoque**

Escopo: O sistema deverá fazer a edição dos produtos no estoque adicionando ou retirando produtos.

Pré-Condição:Ter produtos cadastrados no banco.

Garantia de sucesso:Informar ao usuário por meio de mensagem caso o produto foi ou não editado, se houver algum erro nos campos também tratar de informar ao

usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de produtos no estoque atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão adicionar ao estoque da tela financeira ou em algum item da lista de pesquisa.

2. O sistema então redireciona para tela de adição ao estoque.

3. O usuário faz as alterações necessárias, adicionando ou retirando itens inserindo o valor nos campos corretos, e clica em aceitar.

4. O sistema verifica os campos e altera o estoque no banco retornando a mensagem de alteração feita com sucesso, então redireciona para a tela de pesquisa caso o usuário tenha acessado por ela ou para a tela financeira.

Obs: O usuário poderá adicionar ou remover produtos do estoque acessando a tela de pesquisa e selecionando um item da lista, ou acessando a tela adicionar ao estoque. Ambas fazem a mesma função mas com fluxos diferente, isto foi especificado assim para ser mais fácil e ágil a mudança do estoque.

- **Pesquisar Estoque em falta**

Escopo: O sistema deverá fazer a pesquisa de produtos no estoque, apresentando a lista dos produtos no estoque que estão em falta e possibilitando a pesquisa por nome do produto.

Pré-Condição: Ter produtos cadastrados.

Garantia de sucesso: Apresentar ao usuário a lista do estoque em falta.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar estoque .

2. O sistema então redireciona para tela de pesquisa já preenchida com a lista e um campo para preenchimento.

3. O usuário preenche os campo no botão pesquisar caso queira escolher um ou mais produtos com um nome específico.

4. O usuário visualiza a lista e clica em voltar.

5. O sistema retorna para a tela de pesquisa de produtos em falta.

Obs: Ao clicar em algum item da lista deve-se apresentar a opção de editar o item, no caso fazendo a mesma função do Adicionar ao estoque, mas no final voltando a tela de produtos em falta atualizada.

- **Resumo financeiro**

Escopo: O sistema deverá apresentar um resumo das principais funções.

Pré-Condição: Não há.

Garantia de sucesso: Apresentar um resumo com informações sobre o estoque, produção diária, mensal e trimestral com relação a meta de acordo com o trimestre vigente, bem como o valor total no estoque.

Cenário de sucesso Principal: 1. O usuário clica no botão resumo na tela financeira .

2. O sistema então redireciona para tela de resumo já preenchida com as informações.

4. O usuário visualiza as informações e clica em voltar.

5. O sistema retorna para a tela financeira.

- **Cadastrar uma meta**

Escopo: O sistema deverá fazer o cadastro de metas de acordo com os dados informados pelo usuário.

Pré-Condição: Não há.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a meta foi ou não cadastrada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de metas cadastradas.

Cenário de sucesso Principal: 1. O usuário clica no botão meta da tela financeira.

2. O sistema redireciona para a tela de meta com os botões cadastrar meta e pesquisar metas.

3. O usuário clica em cadastrar meta.

4. O sistema então redireciona para tela de cadastro com os campos para preenchimento : Meta diária, Meta mensal, Meta trimestral, Data de cadastro e período.

5. O sistema também fornecerá um botão de calcular para cada valor de meta, para que preenchida uma das metas o sistema calcula as outras.

5.1. Caso o usuário preencha uma das metas e clique no botão calcular, o sistema calcula as metas restantes.

6. O usuário preenche os campos e clica no botão aceitar caso queira salvar o cadastro.

6.1. Caso o usuário preencha apenas uma meta, o sistema calcula as outras e salva.

6.2. Caso o usuário preencha duas metas, o sistema calcula a terceira e salva no banco.

6.3. Caso o usuário não preencha meta o sistema volta mensagem de erro com a mensagem: Deve se informar pelo menos uma meta.

7. O sistema retorna para a tela de meta.

- **Pesquisar metas**

Escopo: O sistema deverá fazer a pesquisa de metas, apresentando a lista de metas cadastradas em por ordem de cadastro, período e data de cadastro.

Pré-Condição: Ter metas cadastradas.

Garantia de sucesso: Apresentar ao usuário a lista de metas cadastradas ordenadas.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de meta

2. O sistema então redireciona para tela de pesquisa já preenchida com a lista em ordem.

3. O usuário visualiza a lista e clica em voltar.

4. O sistema retorna para a tela de meta.

- **Editar uma Meta**

Escopo: O sistema deverá fazer a edição de metas de acordo com os dados informados pelo usuário.

Pré-Condição: Ter a meta cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a meta foi ou não editada, se houver algum erro nos campos também tratar de informar ao usuário sobre o mesmo. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de metas cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de metas.

2. O sistema então redireciona para tela de pesquisa .

3. O usuário seleciona uma meta clicando na lista.

4. O sistema mostra um menu com as opções Editar e Excluir .

5. O usuário seleciona a opção editar.

6. O sistema redireciona para a tela de cadastro mas com os campos preenchidos.

7. O usuário faz as alterações necessárias e clica em aceitar.

8. O sistema verifica os campos e insere retornando a mensagem de cadastro feito com sucesso, então redireciona para a tela de Meta.

- **Excluir uma meta**

Escopo: O sistema deverá fazer a exclusão de metas de acordo com seleção da meta na lista demonstrada na pesquisa .

Pré-Condição: Ter a meta cadastrada no banco.

Garantia de sucesso: Informar ao usuário por meio de mensagem caso a meta foi ou não excluída. O usuário ao clicar no botão de pesquisa, o sistema mostrará a lista de metas cadastradas atualizada.

Cenário de sucesso Principal: 1. O usuário clica no botão pesquisar da tela de meta.

2. O sistema então redireciona para tela de pesquisa .

3. O usuário seleciona a meta clicando na lista.

4. O sistema mostra um menu com as opções Editar e Excluir .

5. O usuário seleciona a opção excluir.

6. O sistema informa o usuário da ação com dois botões com sim e não.

7. O usuário clica no botão sim.

8. O sistema então exclui a meta mostrando a mensagem de sucesso e atualiza a lista.

APÊNDICE D – Modelo do Projeto

A seguir temos alguns diagramas de atividades e de sequência que representam o modelo do projeto de acordo com os requisitos e os casos de uso encontrados na fase de elaboração e construção.

D.1 Diagramas de Atividades

Estes diagramas auxiliam na implementação das funções do sistema mostrando as atividades que são feitas e o fluxo que segue cada uma para os resultados esperados.

Para o cadastro de clientes, categorias, compromissos e metas aplica-se o diagrama mostrado na figura 18.

Para alteração e exclusão de clientes, categorias, compromissos e metas aplica-se o diagrama mostrado na figura 19

Para o cadastro de pedidos e itens o diagrama que representa-o é o da figura 9

Para alteração e exclusão de pedidos e itens tem-se o diagrama de atividades da figura 20

D.2 Diagramas de sequência

Estes diagramas mostram o fluxo de mensagens mostradas ao usuário, por isso como há mensagens no cadastro, edição e exclusão, só serão apresentados estas funções. Também os diagramas de edição são semelhantes ao de cadastro, mudando apenas as mensagens de cadastro para alteração logo não serão diagramados, podendo usar os de cadastro.

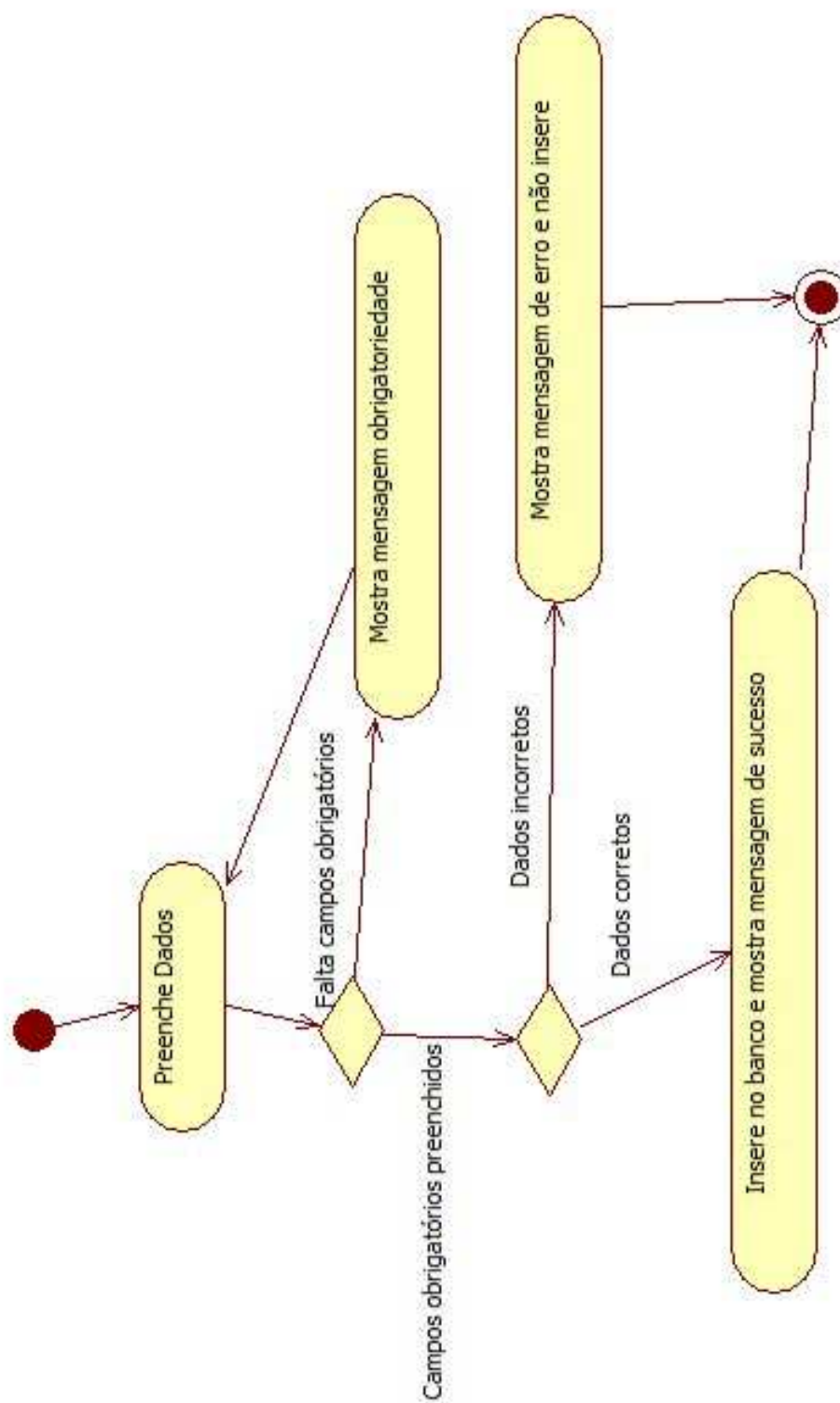


Figura 18 – Diagrama de Atividades para o caso de uso de cadastro de clientes, categorias, sessões e compromissos.

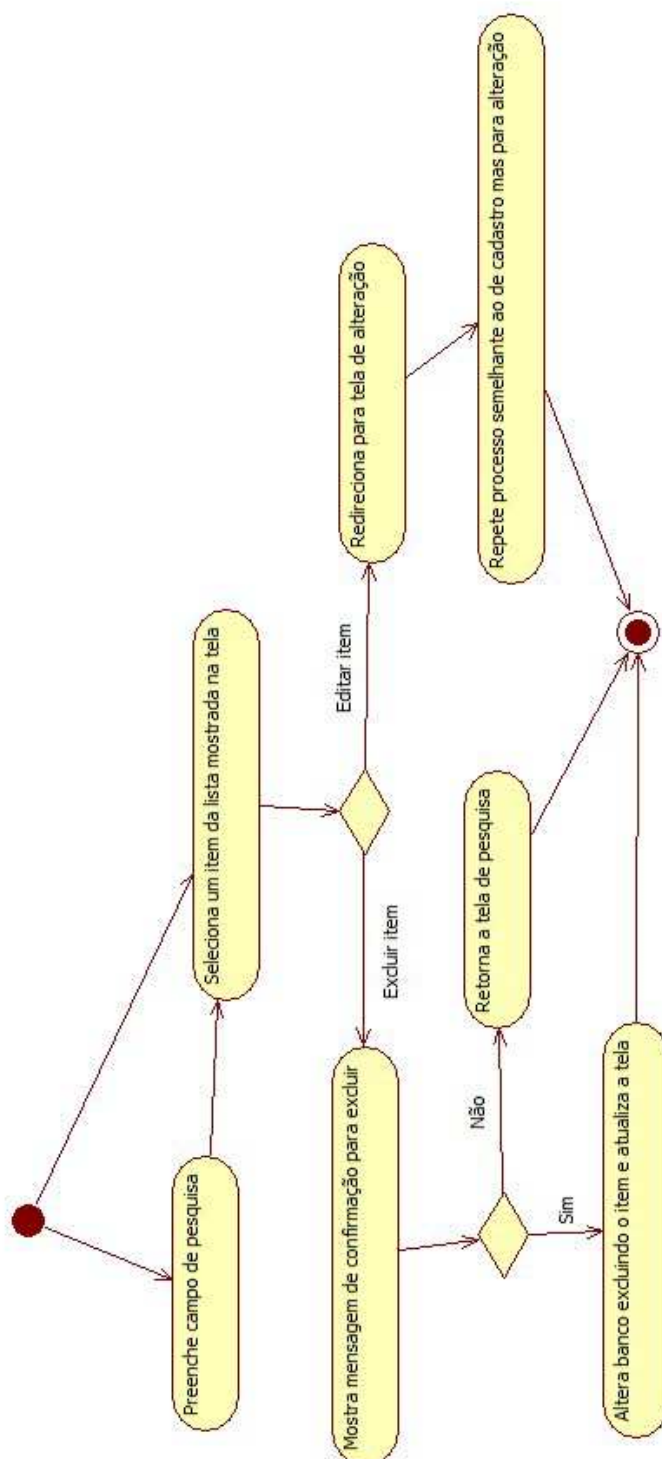


Figura 19 – Diagrama de Atividades para o caso de uso de alteração e exclusão de clientes, categorias, sessões e compromissos.

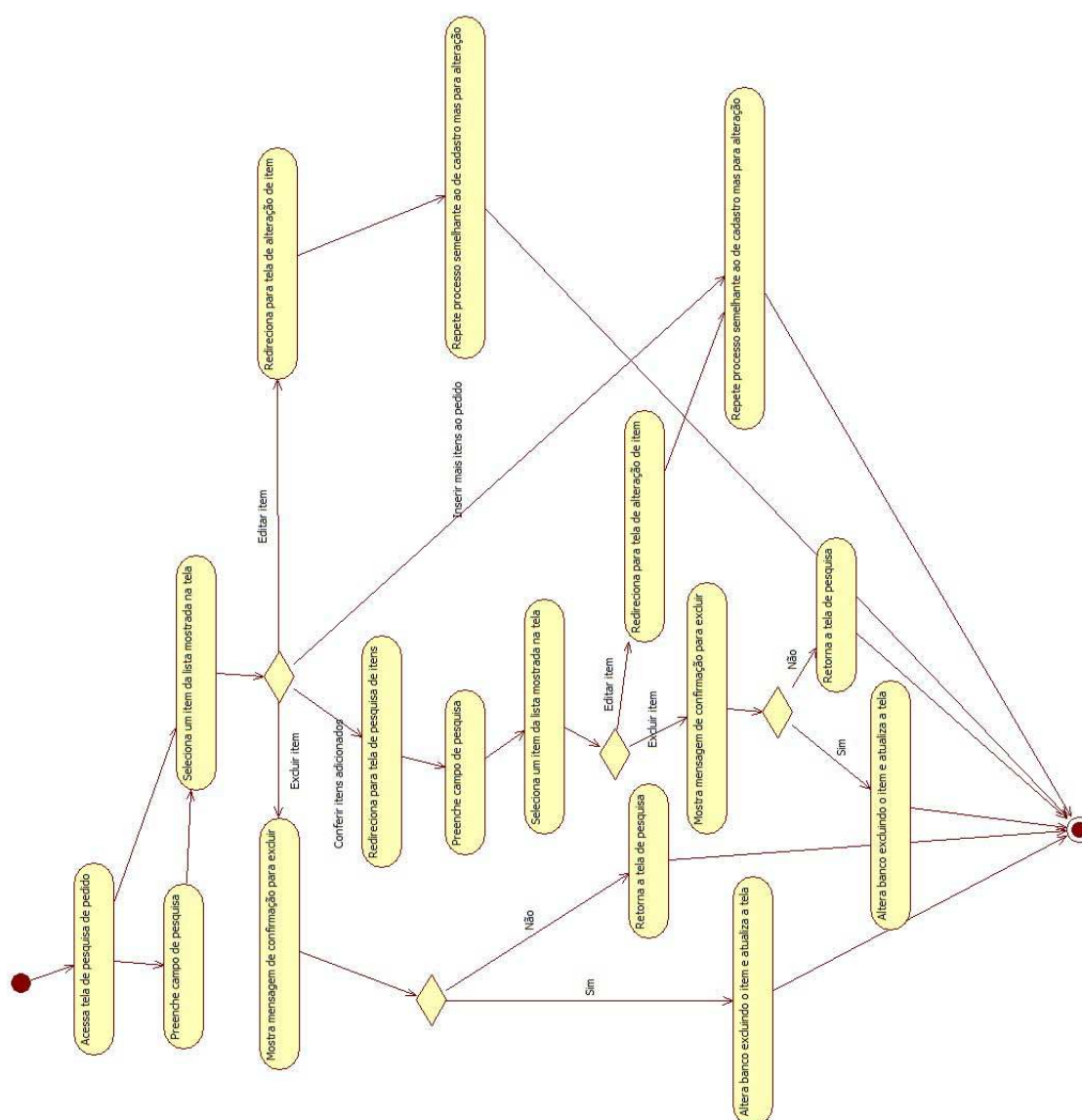


Figura 20 – Diagrama de Atividades para o caso de uso de alteração e exclusão de pedido e itens do pedido.

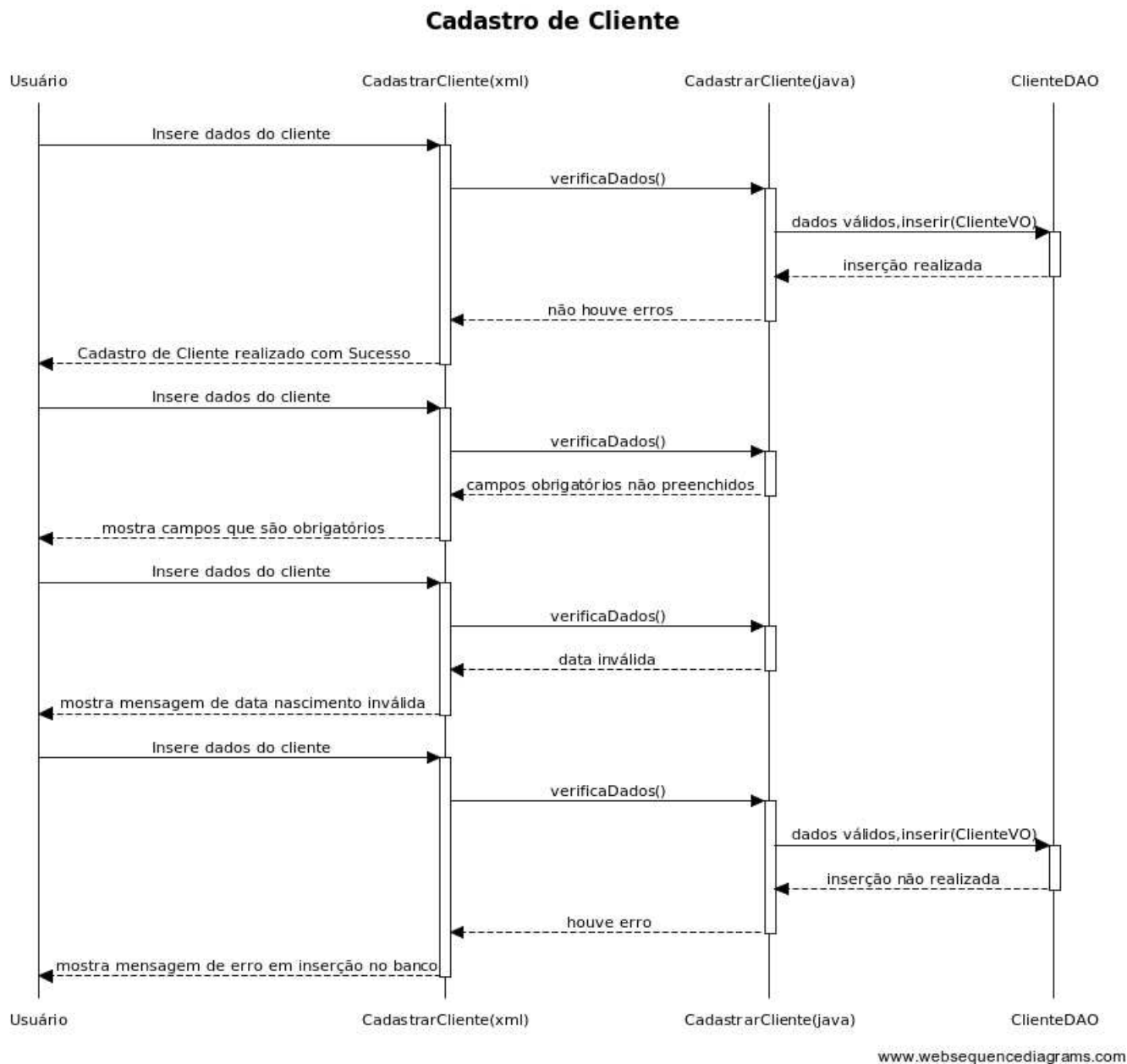


Figura 21 – Diagrama de Sequência para o caso de uso cadastro de clientes.

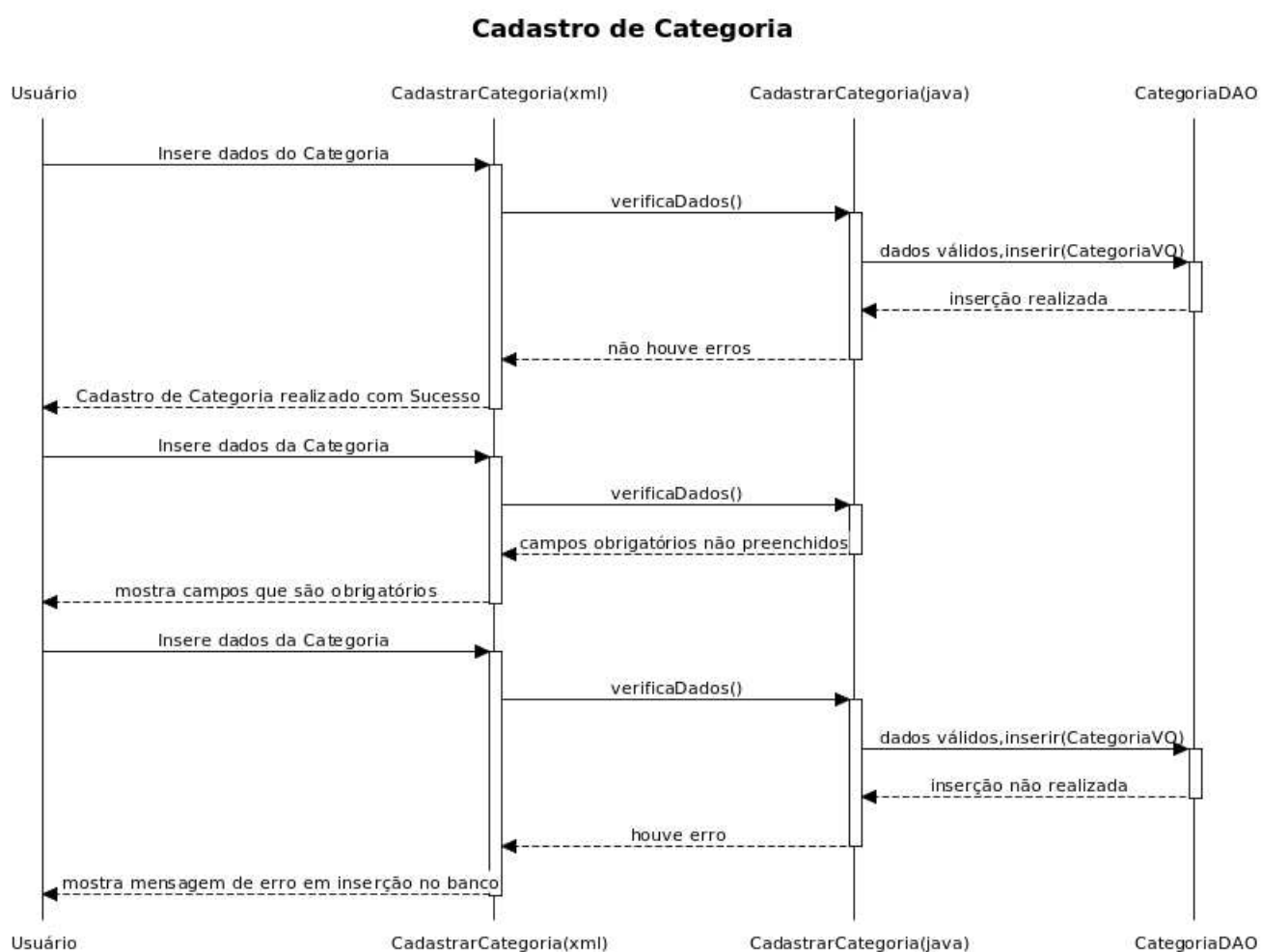


Figura 22 – Diagrama de Sequência para o caso de uso cadastro de categorias.

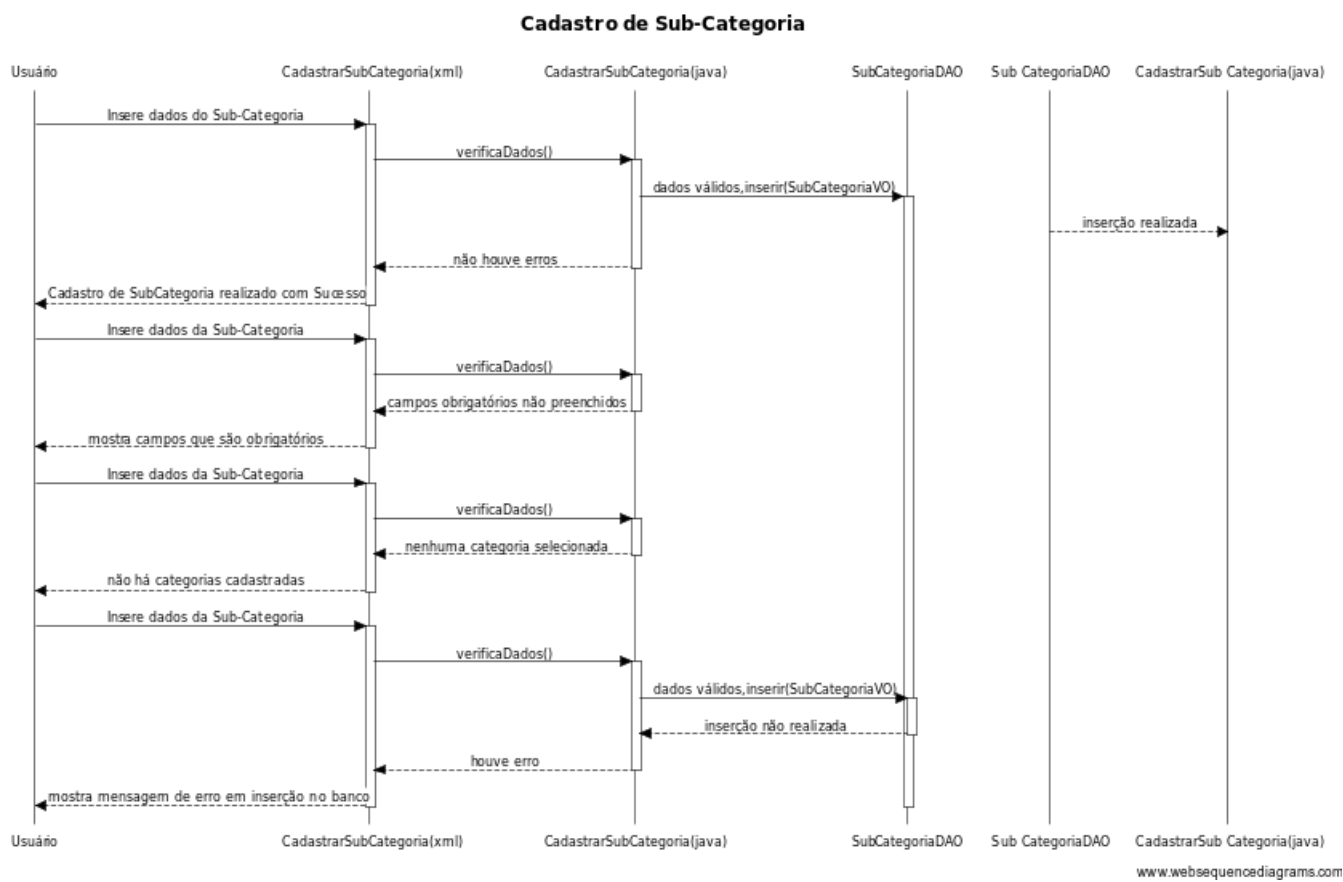


Figura 23 – Diagrama de Sequência para o caso de uso cadastro de subcategorias

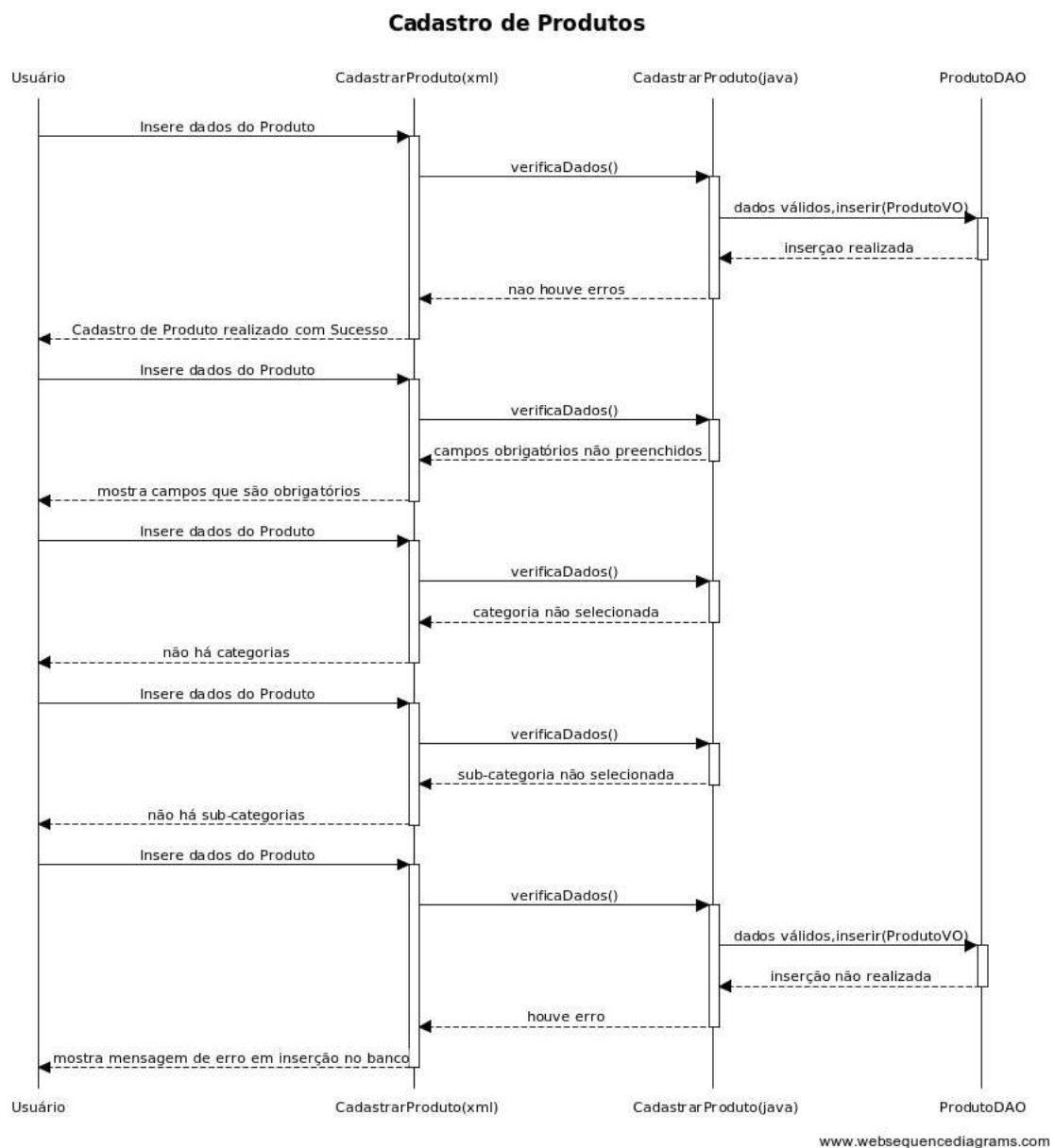


Figura 24 – Diagrama de Sequência para o caso de uso cadastro de produtos.

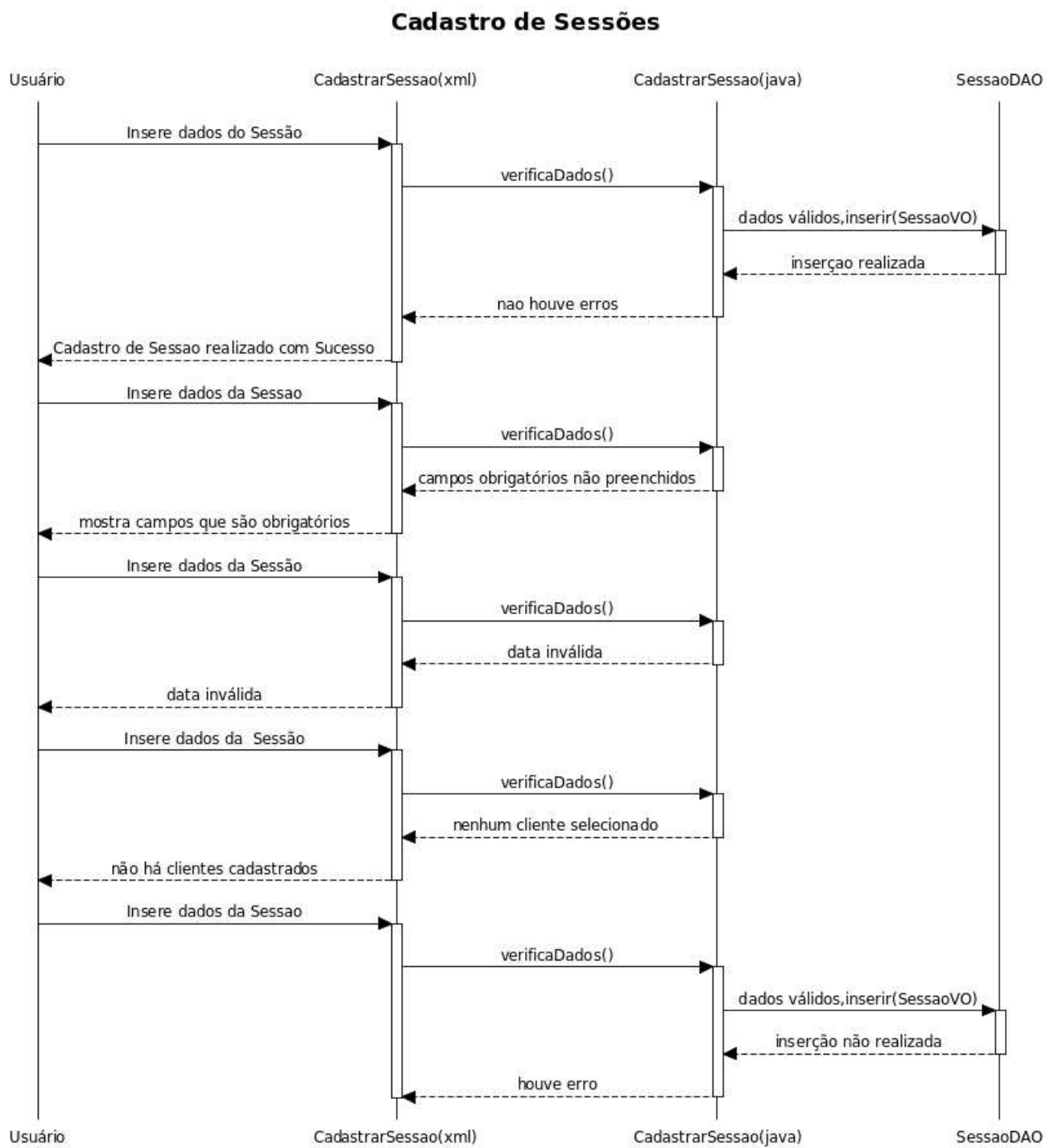


Figura 25 – Diagrama de Sequência para o caso de uso cadastro de sessões.

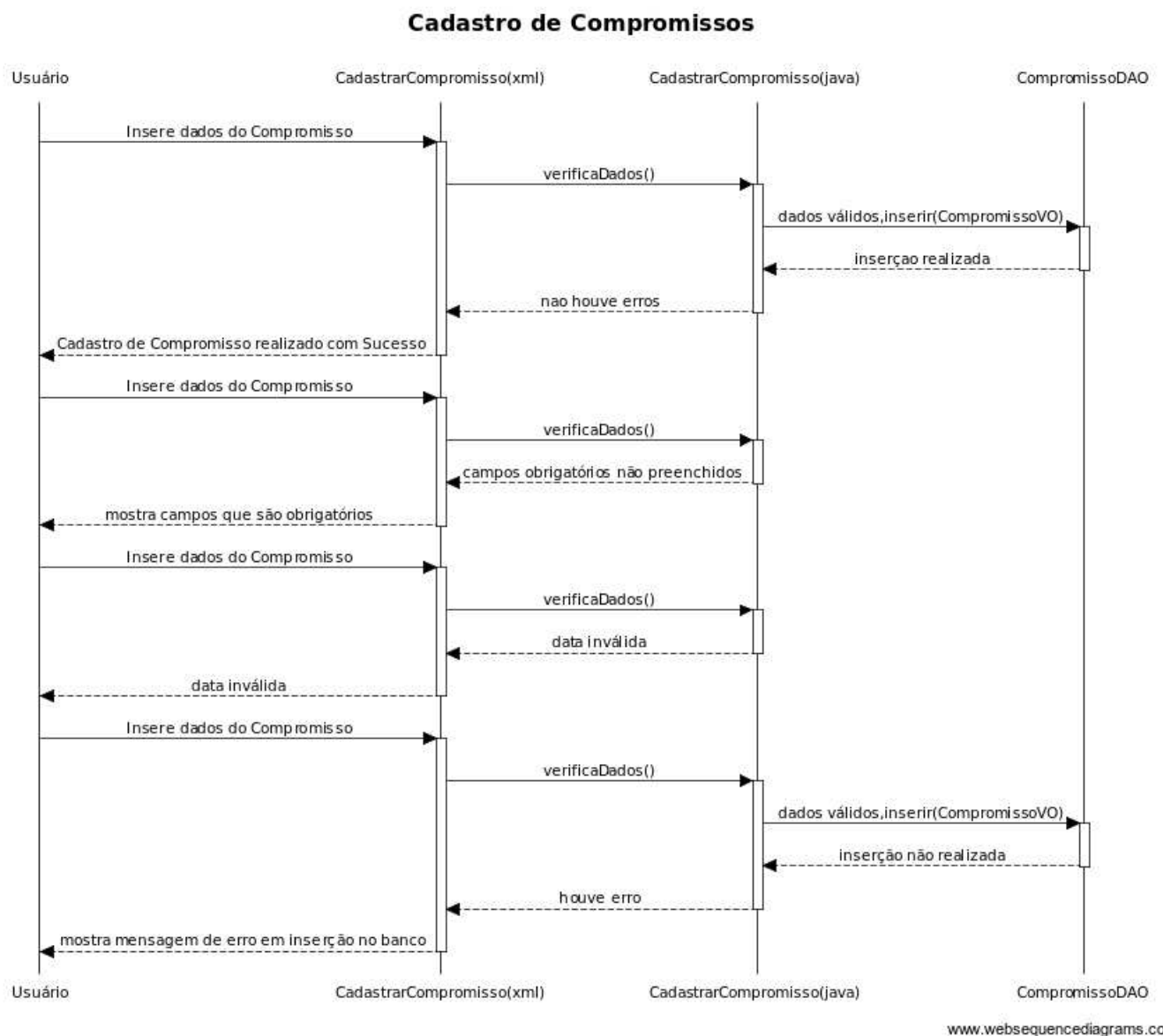


Figura 26 – Diagrama de Sequência para o caso de uso cadastro de compromissos.

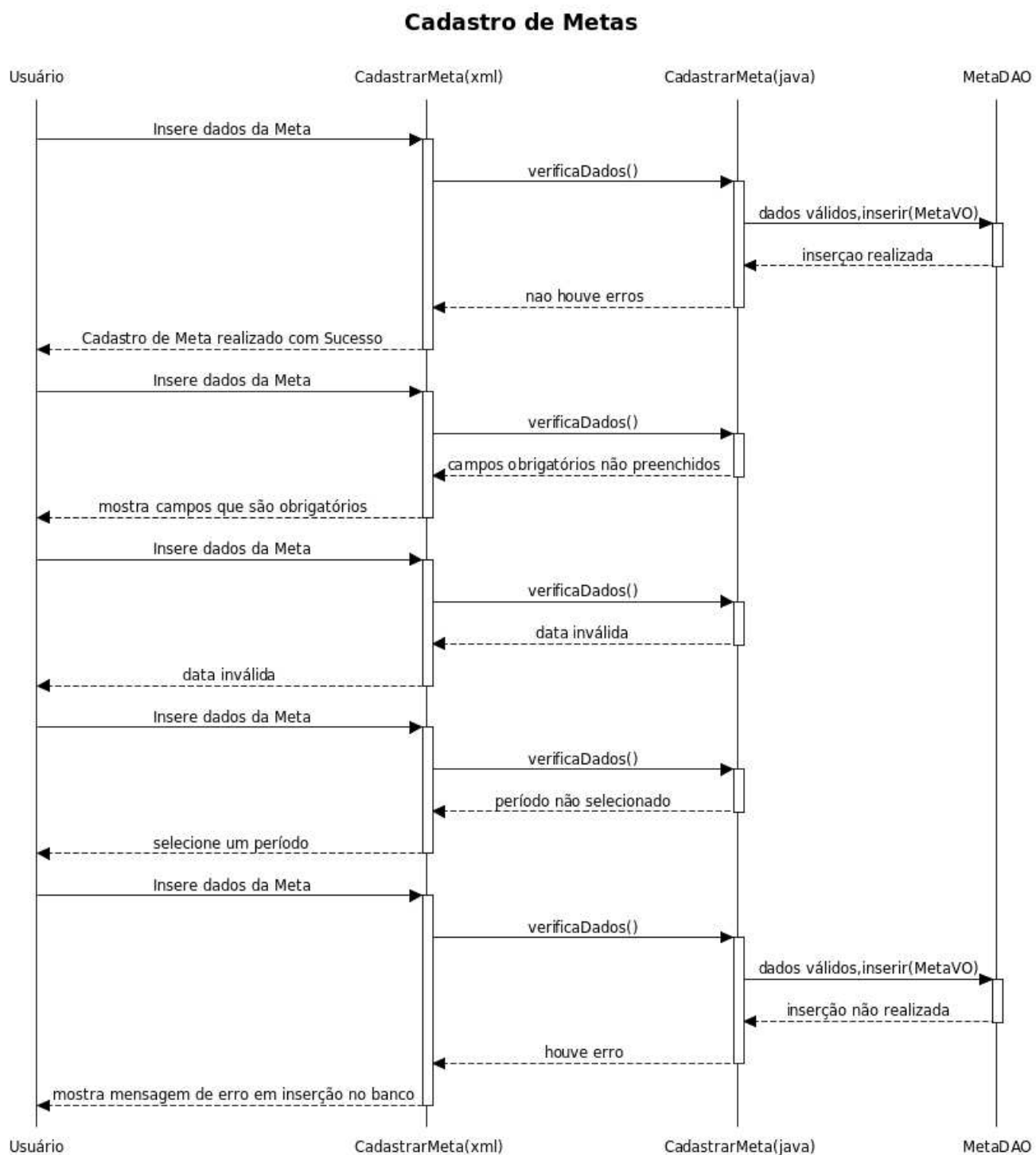


Figura 27 – Diagrama de Sequência para o caso de uso cadastro de metas.

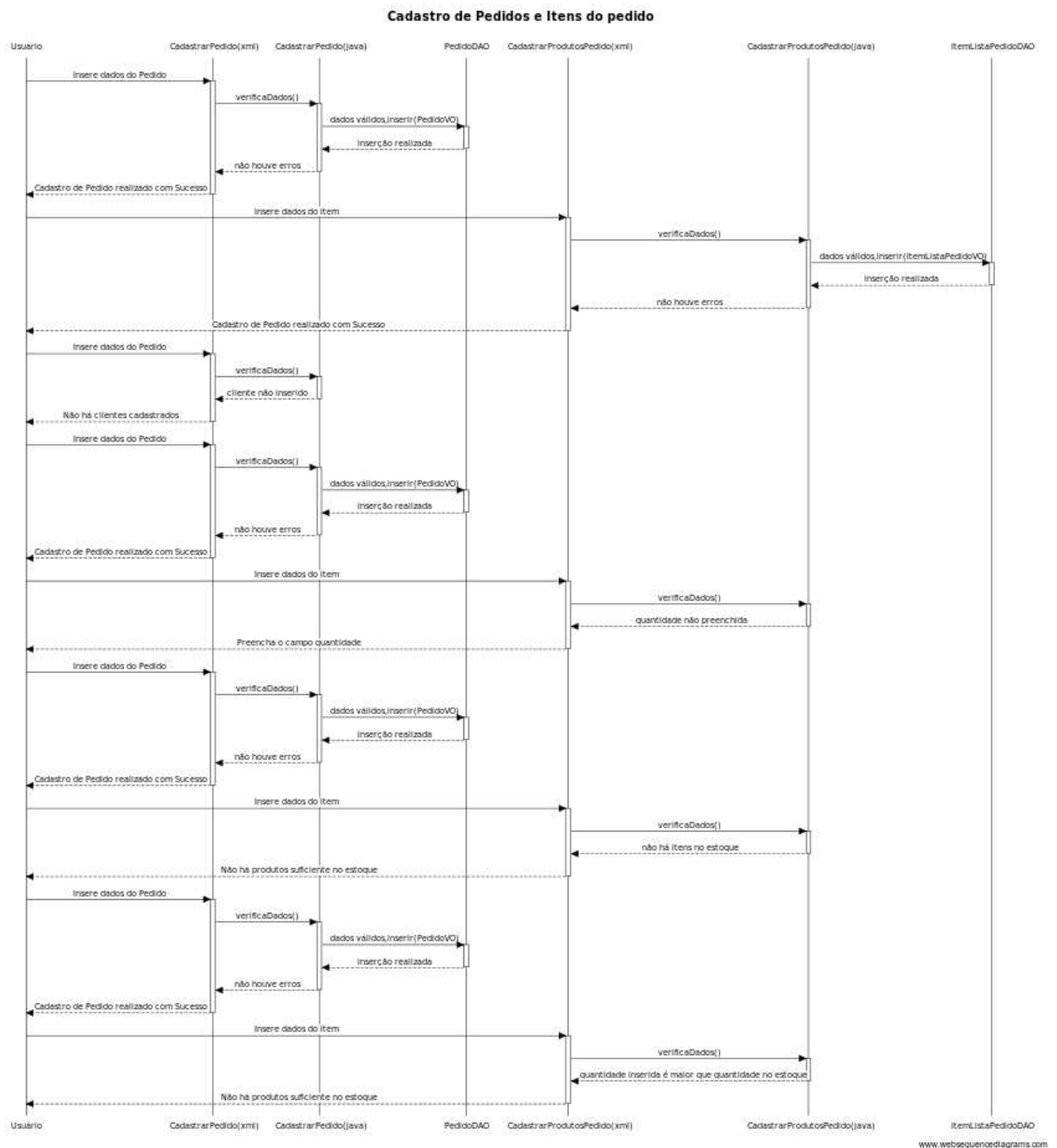


Figura 28 – Diagrama de Sequência para o caso de uso cadastro de pedidos e itens do pedido.

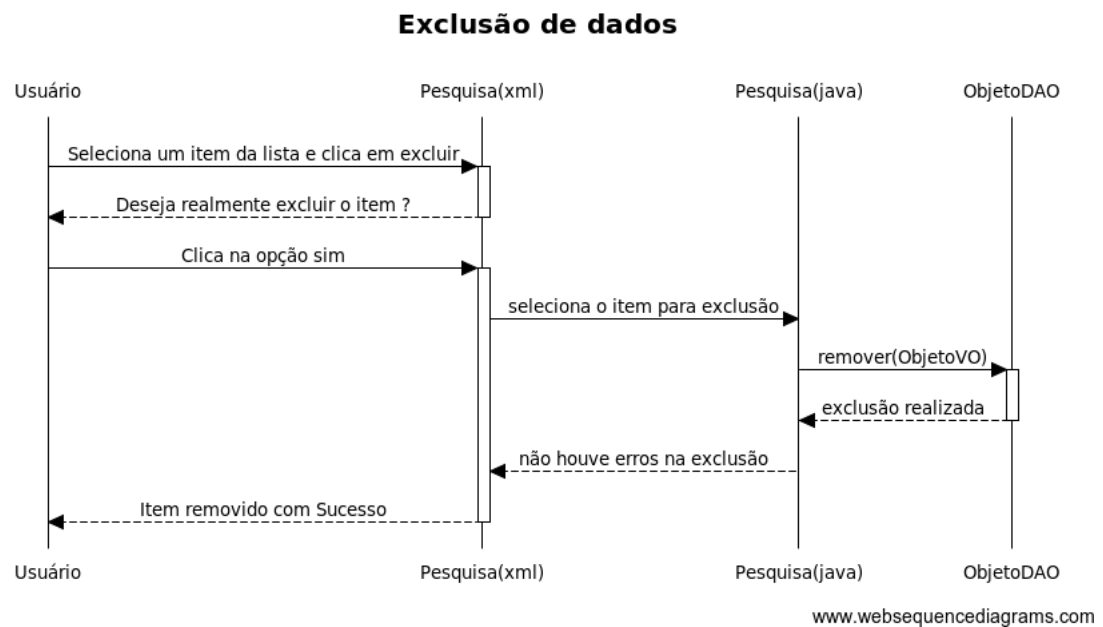


Figura 29 – Diagrama de Sequência para o caso de uso exclusão de itens(Obs: o único caso em que não a exclusão é o de estoque , uma vez que o estoque é excluído junto com o produto).

APÊNDICE E – Protótipo Hi-Fi

Telas referentes ao protótipo Hi-Fi do sistema Meus Estoque Rosa.

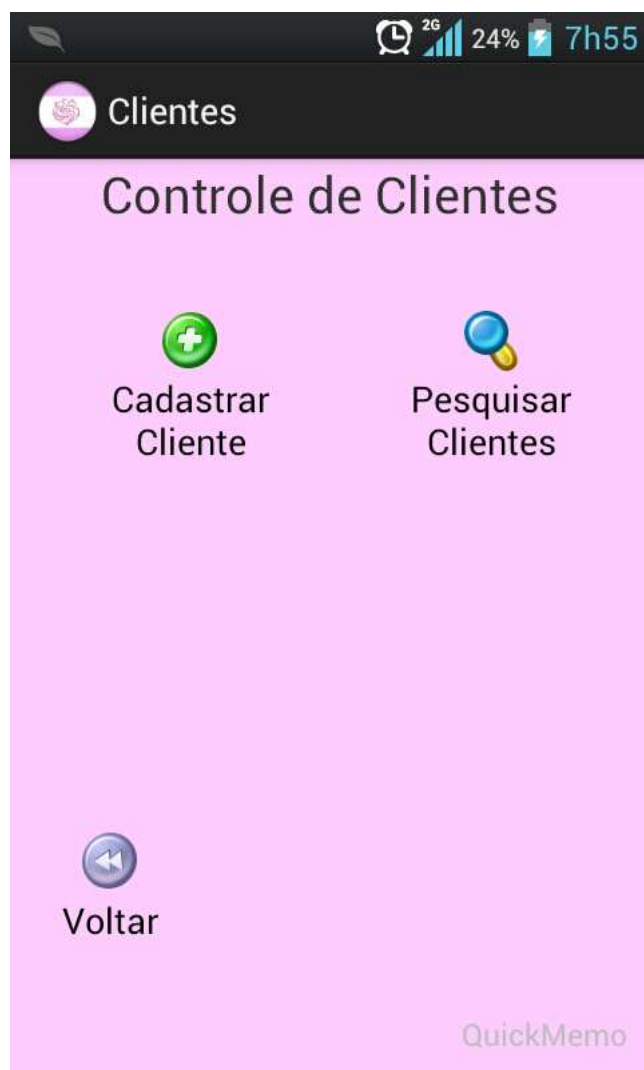


Figura 30 – Protótipo Hi-Fi : Tela de cliente.

2G 25% 7h55

Cadastro Clientes

Preencha os Campos

Nome:

Telefone: Campo Obrigatório

Endereço:

Data de Nascimento:

Observações:

Aceitar Cancelar Voltar

QuickMemo

Figura 31 – Protótipo Hi-Fi : Tela de cadastro de clientes mostrando os campos obrigatórios.

The image shows a mobile application interface for 'Cadastro Clientes' (Client Registration). The status bar at the top indicates 2G network, 26% battery, and 7h56. The app title 'Cadastro Clientes' is displayed in a dark header. Below the title, the instruction 'Preencha os Campos' (Fill in the fields) is shown. The form contains four input fields: 'Nome:' (Name) with the value 'Luana', 'Telefone:' (Phone) with the value '32236333', 'Endereço:' (Address) with the value 'rua alguma coisa 111', and 'Data de Nascimento:' (Date of Birth) which is currently empty. A numeric keypad is overlaid at the bottom of the screen, featuring digits 1-9, 0, a backspace key, and a 'Próx.' (Next) button. The keypad also includes letter suggestions for digits 2-9 (e.g., ABC for 2, DEF for 3).

Nome:	Telefone:	Endereço:	Data de Nascimento:
Luana	32236333	rua alguma coisa 111	

Figura 32 – Protótipo Hi-Fi : Tela de cadastro de clientes mostrando os campos preenchidos sem a data.

Cadastro Clientes

Preencha os Campos

Nome:

Telefone:

Endereço:

Data de Nascimento:

Observações:

Campo Obrigatório

Aceitar Cancelar Voltar

QuickMemo

Figura 33 – Protótipo Hi-Fi : Tela de cadastro de clientes mostrando os campos preenchidos com a data no formato adequado.

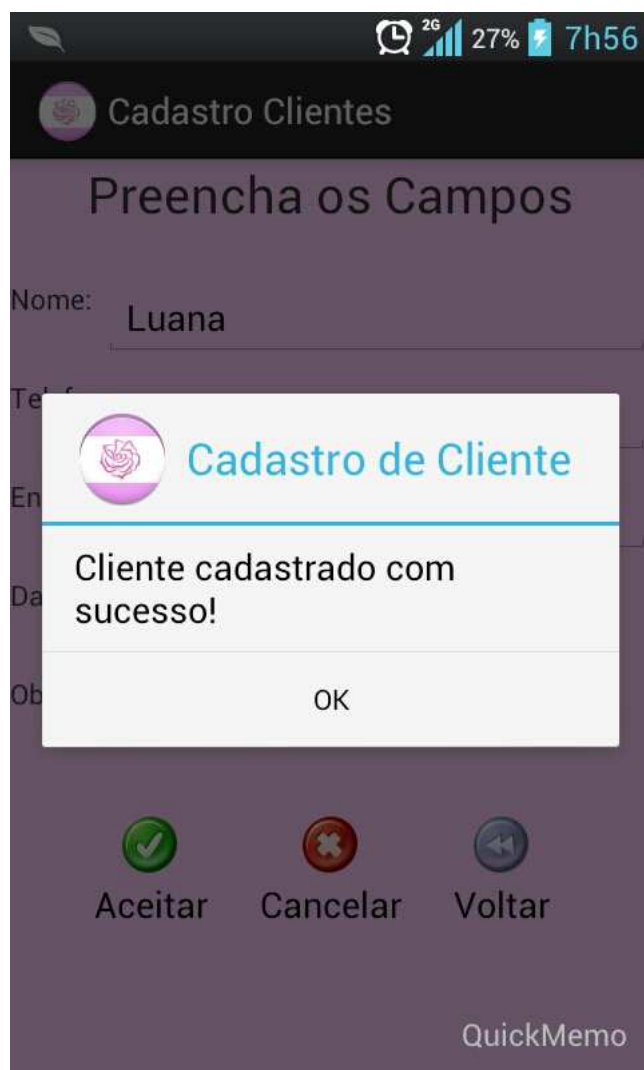


Figura 34 – Protótipo Hi-Fi : Mensagem de sucesso para o cadastro de clientes.

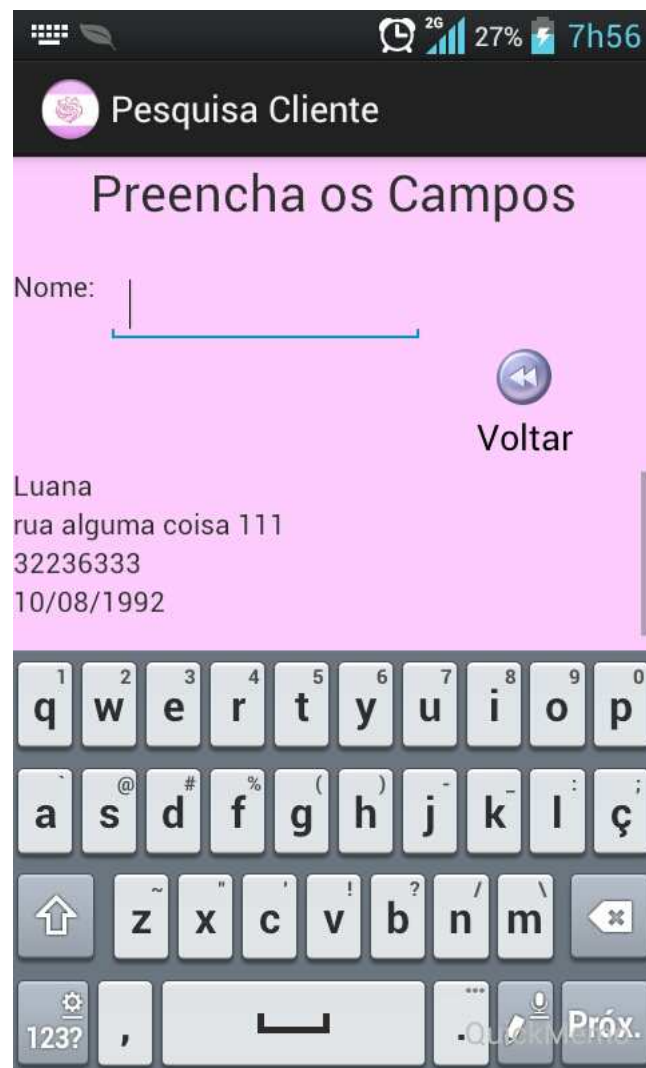


Figura 35 – Protótipo Hi-Fi : Tela de pesquisa de clientes.

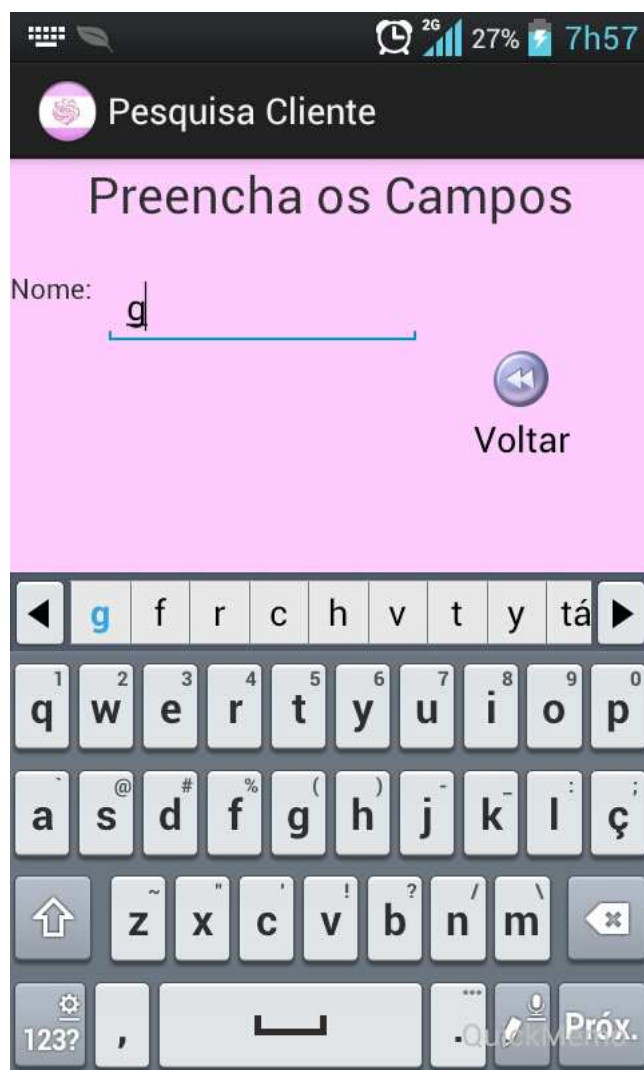


Figura 36 – Protótipo Hi-Fi : Tela de pesquisa de clientes, filtro de pesquisa.



Figura 37 – Protótipo Hi-Fi : Tela de calendário do módulo agenda, seleção de um dia aleatório .

APÊNDICE F – Exemplos de Códigos-Fonte em Java

Neste trecho de código estão descritas todas as variáveis necessárias para a criação das tabelas do módulo de produtos, bem como a instrução para criação das tabelas usando as variáveis.

```
1 //variaveis Cliente
2 public static final String tblCliente = "cliente";
3 public static final String idCliente = "_idCliente";
4 public static final String nomeCliente = "nomeCliente";
5 public static final String enderecoCliente = "enderecoCliente";
6 public static final String telefoneCliente = "telefoneCliente";
7 public static final String dataNascCliente = "dataNascCliente";
8 public static final String obsCliente = "obsCliente";
9 //variaveis SubCategoria
10 public static final String tblSubCategoria = "subCategoria";
11 public static final String idSubCategoria = "_idSubCategoria";
12 public static final String nomeSubCategoria = "nomeSubCategoria";
13 public static final String idCategoriaFKS = "idCategoria";
14 public static final String nomeCategoriaFS = "nomeCategoria";
15 //variaveis Categoria
16 public static final String tblCategoria = "categoria";
17 public static final String idCategoria = "_idCategoria";
18 public static final String nomeCategoria = "nomeCategoria";
19
20 //variaveis Produtos
21 public static final String tblProdutos = "produtos";
22 public static final String idProduto = "_idProdutos";
23 public static final String idCategoriaFKP = "idCategoria";
24 public static final String idSubCategoriaFKP = "idSubCategoria";
25 public static final String descricaoProduto = "descricaoProduto";
26 public static final String nomeProduto = "nomeProduto";
27 public static final String precoProduto = "precoProduto";
28 public static final String nomeCategoriaFP = "nomeCategoria";
29 public static final String nomeSubCategoriaFP = "nomeSubCategoria
    ";
30
31
```

```
32 private static final String DATABASE_NAME = "projeto.db";
33 private static final int DATABASE_VERSION = 1;
34 //Estrutura da tabela Cliente (sql statement)
35 private static final String createCliente = "create table " +
36 tblCliente + "( " + idCliente + " integer primary key
37     autoincrement, " +
38 nomeCliente + " text not null, " +
39 enderecoCliente + " text not null, " +
40 telefoneCliente + " text not null, " +
41 obsCliente + " text , " +
42 dataNascCliente + " text not null));";
43 //Estrutura da tabela SubCategoria
44 private static final String createSubCategoria = "create table "
45 +
46 tblSubCategoria + "( " + idSubCategoria + " integer primary key
47     autoincrement, " +
48 nomeSubCategoria + " text not null, " +
49 idCategoriaFKS + " integer not null, " +
50 "FOREIGN KEY(" + idCategoriaFKS + ") REFERENCES " + tblCategoria+"("
51 + idCategoria+"));";
52 //Estrutura da tabela Categoria
53 private static final String createCategoria = "create table " +
54 tblCategoria + "( " + idCategoria + " integer primary key
55     autoincrement, " +
56 nomeCategoria + " text not null));";
57 //Estrutura da tabela Produtos
58 private static final String createProdutos = "create table " +
59 tblProdutos + "( " + idProduto + " integer primary key
60     autoincrement, " +
61 descricaoProduto + " text not null, " +
62 nomeProduto + " text not null, " +
63 idCategoriaFKP + " integer not null, " +
64 idSubCategoriaFKP + " integer null, " +
65 precoProduto + " real not null," +
66 "FOREIGN KEY(" + idCategoriaFKP + ") REFERENCES " + tblCategoria+"("
67 + idCategoria+"),"
68 + "FOREIGN KEY(" + idSubCategoriaFKP + ") REFERENCES " +
69 tblSubCategoria+"(" + idSubCategoria+"));";
```

Neste trecho de código estão todas os scripts de criação para o banco de dados equivalente aos módulos cliente e produtos, cada instrução *create* cria uma tabela no banco e para que essas instruções sejam executadas a fim de criar o banco e as relações,

o seguinte trecho de código é chamado:

```
1 public void onCreate(SQLiteDatabase database) {
2     //Criacao da tabela
3
4     database.execSQL(createCliente);
5     database.execSQL(createCategoria);
6     database.execSQL(createSubCategoria);
7     database.execSQL(createProdutos);
8 }
```

O trecho a seguir é umas das classes de controle das funções de persistência do banco de dados, a classe apresentada é a de Cliente e está no arquivo ClinteDAO.java .

```
1 public class ClienteDAO {
2     private SQLiteDatabase database;
3     private DatabaseDAO dbHelper;
4     //Campos da tabela Cliente
5     private String[] colunas = {DatabaseDAO.idCliente,
6 DatabaseDAO.nomeCliente,
7 DatabaseDAO.telefoneCliente,
8 DatabaseDAO.enderecoCliente,
9 DatabaseDAO.dataNascCliente,
10 DatabaseDAO.obsCliente};
11 public ClienteDAO(Context context) {
12     dbHelper = new DatabaseDAO(context);
13 }
14 public void open() throws SQLException {
15     database = dbHelper.getWritableDatabase();
16 }
17 public void close() {
18     dbHelper.close();
19 }
20 public long Inserir(ClienteVO pValue) {
21     ContentValues
22     values = new ContentValues();
23     //Carregar os valores nos campos do Cliente que sera incluido
24     values.put(DatabaseDAO.nomeCliente, pValue.getNomeCliente());
25     values.put(DatabaseDAO.enderecoCliente, pValue.getEnderecoCliente
26         ());
27     values.put(DatabaseDAO.telefoneCliente, pValue.getTelefoneCliente
28         ());
29     values.put(DatabaseDAO.dataNascCliente, pValue.getDataNascCliente
30         ());
```

```
28 values.put(DatabaseDAO.obsCliente, pValue.getObsCliente());
29 return database.insert(DatabaseDAO.tblCliente, null, values);
30 }
31 public int Alterar(ClienteVO pValue) {
32     long id = pValue.getIdCliente();
33     ContentValues values = new ContentValues();
34     //Carregar os novos valores nos campos que serao alterados
35     values.put(DatabaseDAO.nomeCliente, pValue.getNomeCliente());
36     values.put(DatabaseDAO.enderecoCliente, pValue.getEnderecoCliente
37         ());
38     values.put(DatabaseDAO.telefoneCliente, pValue.getTelefoneCliente
39         ());
40     values.put(DatabaseDAO.dataNascCliente, pValue.getDataNascCliente
41         ());
42     values.put(DatabaseDAO.obsCliente, pValue.getObsCliente());
43     //Alterar o registro com base no ID
44     return database.update(DatabaseDAO.tblCliente, values,
45         DatabaseDAO.idCliente + " = " + id, null);
46 }
47 public void Excluir(ClienteVO pValue) {
48     long id = pValue.getIdCliente();
49     //Exclui o registro com base no ID
50     database.delete(DatabaseDAO.tblCliente, DatabaseDAO.idCliente + "
51         = " + id, null);
52 }
53 public List<ClienteVO> Consultar() {
54     List<ClienteVO> lstCliente = new ArrayList<ClienteVO>();
55     //Consulta para trazer todos os dados da tabela Cliente ordenados
56     //pela coluna Nome
57     Cursor cursor = database.query(DatabaseDAO.tblCliente, colunas,
58         null, null, null, null, DatabaseDAO.nomeCliente);
59     cursor.moveToFirst();
60     while (!cursor.isAfterLast()) {
61         ClienteVO lClienteVO = cursorToCliente(cursor);
62         lstCliente.add(lClienteVO);
63         cursor.moveToNext();
64     }
65     //Tenha certeza que voce fechou o cursor
66     cursor.close();
67     return lstCliente;
68 }
69 public List<String> ConsultarNomes(){
```

```
64 List<String> nomes = new ArrayList<String>();
65 String consulta = "select "+DatabaseDAO.nomeCliente+" from " +
    DatabaseDAO.tblCliente;
66 Cursor cursor = database.rawQuery(consulta,null);
67 int accountnameIndex = cursor.getColumnIndexOrThrow(DatabaseDAO.
    nomeCliente);
68 if(cursor.moveToFirst()){
69 do{
70 nomes.add(cursor.getString(accountnameIndex));
71 } while(cursor.moveToNext());
72 }
73 cursor.close();
74 return nomes;
75 }
76 public String RetornaNome(long id){
77 String[] args = {Long.toString(id)};
78 String consulta = "select "+DatabaseDAO.nomeCliente+" from " +
    DatabaseDAO.tblCliente+ " where " + DatabaseDAO.idCliente + " =
    ? ";
79 Cursor cursor = (database.rawQuery(consulta, args));
80 int accountnameIndex = cursor.getColumnIndexOrThrow(DatabaseDAO.
    nomeCliente);
81 String nome = "";
82 if(cursor.moveToFirst()){
83 do{
84 nome = cursor.getString(accountnameIndex);
85 } while(cursor.moveToNext());
86 }
87 return nome;
88 }
89 public long RetornaId(String nomeCliente){
90 String[] args = {nomeCliente};
91 String consulta = "select "+DatabaseDAO.idCliente+" from " +
    DatabaseDAO.tblCliente+ " where " + DatabaseDAO.nomeCliente + "
    = ? ";
92 Cursor cursor = (database.rawQuery(consulta, args));
93 int accountnameIndex = cursor.getColumnIndexOrThrow(DatabaseDAO.
    idCliente);
94 long id = 0;
95 if(cursor.moveToFirst()){
96 do{
97 id = Long.parseLong(cursor.getString(accountnameIndex));
```

```
98 } while(cursor.moveToNext());
99 }
100 return id;
101 }
102 //Converter o Cursor de dados no objeto POJO ClienteVO
103 private ClienteVO cursorToCliente(Cursor cursor) {
104     ClienteVO lClienteVO = new ClienteVO();
105     lClienteVO.setIdCliente(cursor.getLong(0));
106     lClienteVO.setNomeCliente(cursor.getString(1));
107     lClienteVO.setTelefoneCliente(cursor.getString(2));
108     lClienteVO.setEnderecoCliente(cursor.getString(3));
109     lClienteVO.setDataNascCliente(cursor.getString(4));
110     lClienteVO.setObsCliente(cursor.getString(5));
111     return lClienteVO;
112 }
113
114 }
```

A seguir tem-se um exemplo de classe POJO

```
1 //responsavel pelo transporte de dados na aplicacao e com o banco
2 public class ClienteVO implements Serializable{
3
4     private static final long serialVersionUID = 1L;
5     private long idCliente;
6     private String nomeCliente;
7     private String enderecoCliente;
8     private String telefoneCliente;
9     private String dataNascCliente;
10    private String obsCliente;
11    public long getIdCliente() {
12        return idCliente;
13    }
14    public void setIdCliente(long idCliente) {
15        this.idCliente = idCliente;
16    }
17    public String getNomeCliente() {
18        return nomeCliente;
19    }
20    public void setNomeCliente(String nomeCliente) {
21        this.nomeCliente = nomeCliente;
22    }
23    public String getEnderecoCliente() {
```

```
24 return enderecoCliente;
25 }
26 public void setEnderecoCliente(String enderecoCliente) {
27     this.enderecoCliente = enderecoCliente;
28 }
29 public String getTelefoneCliente() {
30     return telefoneCliente;
31 }
32 public void setTelefoneCliente(String telefoneCliente) {
33     this.telefoneCliente = telefoneCliente;
34 }
35 public String getDataNascCliente() {
36     return dataNascCliente;
37 }
38 public void setDataNascCliente(String dataNascCliente) {
39     this.dataNascCliente = dataNascCliente;
40 }
41 public String toString() {
42     return nomeCliente;
43 }
44 public String getObsCliente() {
45     return obsCliente;
46 }
47 public void setObsCliente(String obsCliente) {
48     this.obsCliente = obsCliente;
49 }
50
51 }
```

Um exemplo de pesquisa ao banco pode ser visto no trecho de código a seguir:

```
1 public ProdutoVO Consultar(long idproduto) {
2     ProdutoVO produto = new ProdutoVO();
3     String[] args = {String.valueOf(idproduto)};
4     String consulta = "select * from "+DatabaseDAO.tblProdutos+ " as
        a inner join "+DatabaseDAO.tblCategoria+" as b on a."+
        DatabaseDAO.idCategoriaFKP+" = b."+DatabaseDAO.idCategoria+"
5 + " inner join "+DatabaseDAO.tblSubCategoria+" as c on a." +
        DatabaseDAO.idSubCategoriaFKP+" = c."+DatabaseDAO.
        idSubCategoria+" where "+DatabaseDAO.idProduto+" = ?";
6     Cursor cursor = database.rawQuery(consulta,args);
7     cursor.moveToFirst();
8     while (!cursor.isAfterLast()) {
```

```
9 produto = cursorToProdutoVO(cursor);
10 cursor.moveToNext();
11 }
12
13 cursor.close();
14 return produto;
15 }
```

A estrutura responsável pelas tabelas do módulo financeiro pode ser vista a seguir, e pode-se observar como são chamadas as funções de criação das tabelas, que devem respeitar a ordem de relacionamento das classes para que o banco funcione corretamente.

```
1
2 //variaveis Estoque
3 public static final String tblEstoque = "estoque";
4 public static final String idEstoque= "_idEstoque";
5 public static final String idProdutoPKFKE = "idProdutoEstoque";
6 public static final String quantidadeEstoque = "quantidadeEstoque";
7 // variaveis Pedido
8 public static final String tblPedido = "pedido";
9 public static final String idPedido = "_idPedido";
10 public static final String valorTotal= "valorTotal";
11 public static final String idClienteFKP = "idCliente";
12 public static final String dataPedido = "dataPedido";
13 public static final String numParcelas = "numParcelas";
14 public static final String valorParcela = "valorParcela";
15 //variaveis itemListaPedido
16 public static final String tblItemListaPedido = "itemListaPedido";
17
18 public static final String idItem = "_idItem";
19 public static final String quantidadeItens = "quantidadeItens";
20 public static final String idProdutoFKI = "idProduto";
21 public static final String idListaPedidoFK= "idListaPedido";
22 //variaveis listaPedido
23 public static final String tblListaPedido = "listaPedido";
24 public static final String idListaPedido = "_idListaPedido";
25 public static final String idPedidoFK= "idPedido";
26 //variaveis Meta
27 public static final String tblMeta = "meta";
28 public static final String idMeta = "_idMeta";
29 public static final String metaDiaria = "metaDiaria";
30 public static final String metaMensal = "metaMensal";
```

```
30 public static final String metaTrimestre = "metaTrimestre";
31 public static final String periodoAno = "periodoAno";
32 public static final String dataCadastroMeta = "dataCadastroMeta";
33 //Estrutura da tabela Estoque
34 private static final String createEstoque = "create table " +
35 tblEstoque + "( " + idEstoque + " integer primary key
36     autoincrement, " +
37 idProdutoPKFKE + " integer not null, " +
38 quantidadeEstoque + " integer not null ,"+
39 "FOREIGN KEY("+ idProdutoPKFKE +") REFERENCES "+ tblProdutos+"(" +
40     idProduto+"))";
41 //Estrutura da tabela Pedido
42 private static final String createPedido = "create table " +
43 tblPedido + "( " + idPedido + " integer primary key autoincrement
44     , " +
45 valorTotal + " real not null, " +
46 idClienteFKP + " long not null ,"+
47 dataPedido + " text not null ,"+
48 numParcelas + " integer not null ,"+
49 valorParcela + " real not null ,"+
50 "FOREIGN KEY("+ idClienteFKP +") REFERENCES "+ tblCliente+"(" +
51     idCliente+"))";
52
53 private static final String createListaPedido = "create table " +
54 tblListaPedido + "( " + idListaPedido + " integer primary key
55     autoincrement, " +
56 idPedidoFK + " long not null, " +
57 "FOREIGN KEY("+ idPedidoFK +") REFERENCES "+ tblPedido+"(" +
58     idPedido+"))";
59
60 private static final String createItemListaPedido = "create table
61     " +
62 tblItemListaPedido + "( " + idItem + " integer primary key
63     autoincrement, " +
64 idListaPedidoFK + " long not null, " +
65 idProdutoFKI + " long not null, " +
66 quantidadeItens + " integer not null, " +
67 "FOREIGN KEY("+ idListaPedidoFK +") REFERENCES "+ tblListaPedido+
68     "(" + idListaPedido+"), " +
69 "FOREIGN KEY("+ idProdutoFKI +") REFERENCES "+ tblProdutos+"(" +
70     idProduto+"))";
```

```
62 private static final String createMeta = "create table " +
63 tblMeta + "( " + idMeta + " integer primary key autoincrement, "
64     +
65 metaDiaria + " real not null, " +
66 metaMensal + " real not null, " +
67 metaTrimestre + " real not null, "+
68 periodoAno + " integer not null, " +
69 dataCadastroMeta + " text not null );";
70
71 public void onCreate(SQLiteDatabase database) {
72     //Criacao da tabela
73     //...
74     database.execSQL(createEstoque);
75     database.execSQL(createPedido);
76     database.execSQL(createListaPedido);
77     database.execSQL(createItemListaPedido);
78     database.execSQL(createMeta);
79 }
```

APÊNDICE G – Descrição dos casos de Teste Unitários

Caso de uso: Cadastrar Cliente	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Cliente cadastrado com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de clientes e clicar no botão Cadastrar Cliente, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 9 – Caso de Teste: Cadastro de Clientes

Caso de uso: Cadastrar Categoria	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco. A mensagem de sucesso deve ser a seguinte: Categoria cadastrada com sucesso! A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de produtos e clicar no botão Cadastro e na opção Cadastrar Categoria, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 10 – Caso de Teste: Cadastro de Categorias

Caso de uso: Cadastrar Subcategoria	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco. A mensagem de sucesso deve ser a seguinte: Subcategoria cadastrada com sucesso! A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de produtos e clicar no botão Cadastro e na opção Cadastrar Subcategoria, inserir os dados e confirmar as saídas
Pré Requisito	Ter categorias cadastradas
Observação	Não há

Tabela 11 – Caso de Teste: Cadastro de Subcategorias

Caso de uso: Cadastrar Produto	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Produto cadastrado com sucesso! A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de produtos e clicar no botão Cadastro e na opção Cadastrar Produto, inserir os dados e confirmar as saídas
Pré Requisito	Ter categorias e subcategorias cadastradas.
Observação	Não há

Tabela 12 – Caso de Teste: Cadastro de Produtos

Caso de uso: Cadastrar Sessão	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Sessão cadastrada com sucesso! A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de agenda e clicar no botão Sessão e na opção Cadastrar Sessão, inserir os dados e confirmar as saídas
Pré Requisito	Ter clientes cadastrados.
Observação	Deve-se garantir que não haja sessões cadastradas no mesmo horário

Tabela 13 – Caso de Teste: Cadastro de Sessões

Caso de uso: Cadastrar Compromisso	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Compromisso cadastrado com sucesso! A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de agenda e clicar no botão Compromisso e na opção Cadastrar Compromisso, inserir os dados e confirmar as saídas
Pré Requisito	Não há.
Observação	Deve-se garantir que não haja compromissos cadastradas no mesmo horário

Tabela 14 – Caso de Teste: Cadastro de Compromissos

Caso de uso: Cadastrar Meta	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem, verificar se o calculo das metas que não foram inseridas foi feito. A mensagem de sucesso deve ser a seguinte: Meta cadastrada com sucesso! A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Meta e na opção Cadastrar Meta, inserir os dados e confirmar as saídas
Pré Requisito	Não há.
Observação	Deve-se garantir que ao clicar no botão calcular o sistema calcule as metas de forma correta e aproximada.

Tabela 15 – Caso de Teste: Cadastro de Metas

Caso de uso: Cadastrar Pedido e Itens do Pedido	
Objetivo	Testar o caso de uso verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Pedido cadastrado com sucesso! A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique. Neste mesmo teste deve se testar se o cadastro de itens está de acordo, assim clicando em adicionar produtos após o preenchimento dos campos em um novo cadastro de pedidos, no caso deve seguir a mesma ideia do cadastro de pedidos.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Cadastrar Pedido , inserir os dados e confirmar as saídas, bem como para a adição de itens ao pedido.
Pré Requisito	Ter clientes cadastrados e produtos no estoque.
Observação	Não há.

Tabela 16 – Caso de Teste: Cadastro de Pedidos e Itens do pedido

Caso de uso: Adicionar ao estoque	
Objetivo	Testar se ao acessar a opção da tela financeira, Adicionar ao Estoque, o sistema redireciona para a tela de alteração do estoque, verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Estoque alterado com sucesso! , esta mensagem deve aparecer no momento que o usuário clicar em Adicionar mas possibilitar a inserção de mais itens. A mensagem caso o campo esteja faltando deve ser mostrada por meio de um ícone de exclamação no campo obrigatório, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique..
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Adicionar ao Estoque , inserir os dados e confirmar as saídas
Pré Requisito	Não há.
Observação	Deve-se garantir que ao clicar no botão pesquisar estoque e selecionar um item, logo clicar em editar o sistema também redirecione para a mesma tela.

Tabela 17 – Caso de Teste: Adicionar ao Estoque

Caso de uso: Pesquisar Clientes	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Clientes o sistema redireciona para a tela de pesquisa com a lista de clientes cadastrados, caso o usuário preencha o campo de Nome Cliente o sistema filtra a lista pelo nome inserido.
Descrição	Para este teste é preciso acessar a tela de clientes e clicar no botão Pesquisar Clientes. Para testar o filtro basta preencher o campo Nome Cliente.
Pré Requisito	Ter clientes cadastrados.
Observação	Deve-se filtrar os clientes a partir da primeira letra inserida.

Tabela 18 – Caso de Teste: Pesquisar Clientes

Caso de uso: Pesquisar Categorias	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Categorias o sistema redireciona para a tela de pesquisa com a lista de Categorias cadastrados, caso o usuário preencha o campo de Nome Categoria o sistema filtra a lista pelo nome inserido.
Descrição	Para este teste é preciso acessar a tela de categorias e clicar no botão Pesquisar Categorias. Para testar o filtro basta preencher o campo Nome Categoria.
Pré Requisito	Ter categorias cadastrados.
Observação	Deve-se filtrar os Categorias a partir da primeira letra inserida.

Tabela 19 – Caso de Teste: Pesquisar Categorias

Caso de uso: Pesquisar Subcategorias	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Subcategorias o sistema redireciona para a tela de pesquisa com a lista de Subcategorias cadastrados, caso o usuário preencha o campo de Nome Subcategoria o sistema filtra a lista pelo nome inserido.
Descrição	Para este teste é preciso acessar a tela de subcategorias e clicar no botão Pesquisar Subcategorias. Para testar o filtro basta preencher o campo Nome Subcategoria.
Pré Requisito	Ter subcategorias cadastrados.
Observação	Deve-se filtrar os Subcategorias a partir da primeira letra inserida.

Tabela 20 – Caso de Teste: Pesquisar Subcategorias

Caso de uso: Pesquisar Produtos	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Produtos o sistema redireciona para a tela de pesquisa com a lista de produtos cadastrados, caso o usuário preencha o campo de Nome Produto o sistema filtra a lista pelo nome inserido.
Descrição	Para este teste é preciso acessar a tela de produtos e clicar no botão Pesquisar Produtos. Para testar o filtro basta preencher o campo Nome Produto.
Pré Requisito	Ter produtos cadastrados.
Observação	Deve-se filtrar os produtos a partir da primeira letra inserida.

Tabela 21 – Caso de Teste: Pesquisar Produtos

Caso de uso: Visualizar Calendário	
Objetivo	Testar verificando se ao clicar no botão Calendário o sistema redireciona para a tela de pesquisa com a lista de sessões e compromissos do dia vigente e o calendário marcando o dia atual, caso o usuário clique em qualquer dia mostrado no calendário, o sistema deve retornar a lista de sessões e compromissos do dia selecionado.
Descrição	Para este teste é preciso acessar a tela de agenda e clicar no botão Calendário. Para testar o filtro basta clicar em um dia aleatório no calendário exibido.
Pré Requisito	Ter sessões e compromissos cadastrados.
Observação	Não há.

Tabela 22 – Caso de Teste: Visualizar Calendário

Caso de uso: Pesquisar Compromissos	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Compromissos o sistema redireciona para a tela de pesquisa com a lista de compromissos cadastrados, caso o usuário preencha os campos de Data Início e Data Fim o sistema filtra a lista pelo intervalo de data, a pesquisa também deve funcionar para apenas um campo preenchido ou nenhum.
Descrição	Para este teste é preciso acessar a tela de compromisso e clicar no botão Pesquisar Compromissos. Para testar o filtro basta preencher os campos de Data Início e Data Fim.
Pré Requisito	Ter compromissos cadastrados.
Observação	Não há.

Tabela 23 – Caso de Teste: Pesquisar Compromissos

Caso de uso: Pesquisar Metas	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Metas o sistema redireciona para a tela de pesquisa com a lista de metas cadastradas ordenadas por data de cadastro, período e id da meta.
Descrição	Para este teste é preciso acessar a tela de Meta e clicar no botão Pesquisar Metas.
Pré Requisito	Ter metas cadastradas.
Observação	Não há.

Tabela 24 – Caso de Teste: Pesquisar Metas

Caso de uso: Resumo Financeiro	
Objetivo	Testar verificando se ao clicar no botão Resumo o sistema redireciona para a tela com os dados do ganho diário, mensal e trimestral mostrando quanto falta para atingir a meta ou quanto passou levando em consideração a data atual, verificar se a busca esta retornando os dados corretos de acordo com a data atual. Deve verificar também se esta mostrando o número de itens no estoque e o valor em reais, bem como o número de compromissos e sessões do mês e o número de clientes inseridos até o momento.
Descrição	Para este teste é preciso acessar a tela Financeira e clicar no botão Resumo.
Pré Requisito	Não há.
Observação	Não há.

Tabela 25 – Caso de Teste: Resumo Financeiro

Caso de uso: Pesquisar Estoque	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Estoque o sistema redireciona para a tela de pesquisa com a lista de produtos no estoque, caso o usuário preencha o campo de Nome Produto o sistema filtra a lista pelo nome inserido.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Pesquisar Estoque. Para testar o filtro basta preencher o campo Nome Produto.
Pré Requisito	Ter produtos cadastrados, logo ter o cadastro no estoque.
Observação	Ao cadastrar um produto, o sistema também cadastra o estoque.

Tabela 26 – Caso de Teste: Pesquisar Estoque

Caso de uso: Pesquisar Produtos em falta	
Objetivo	Testar verificando se ao clicar no botão Produtos em Falta o sistema redireciona para a tela de pesquisa com a lista de produtos no estoque que estão com quantidade igual a zero, caso o usuário preencha o campo de Nome Produto o sistema filtra a lista pelo nome inserido.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Produtos em Falta. Para testar o filtro basta preencher o campo Nome Produto.
Pré Requisito	Ter produtos cadastrados, logo ter o cadastro no estoque.
Observação	Ao cadastrar um produto, o sistema também cadastra o estoque. Caso o usuário altere o estoque a lista de produtos em falta deve ser alterada.

Tabela 27 – Caso de Teste: Pesquisar Produtos em Falta

Caso de uso: Pesquisar Sessões	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Sessões o sistema redireciona para a tela de pesquisa com a lista de sessões , caso o usuário preencha os campos de Data Início e Data Fim o sistema filtra a lista pelo intervalo de data, a pesquisa também deve funcionar para apenas um campo preenchido ou nenhum.
Descrição	Para este teste é preciso acessar a tela de agenda e clicar no botão Sessões e em Pesquisar Sessões. Para testar o filtro basta preencher os campos de Data Início e Data Fim.
Pré Requisito	Ter sessões cadastradas.
Observação	Não há

Tabela 28 – Caso de Teste: Pesquisar Sessões

Caso de uso: Pesquisar Pedidos	
Objetivo	Testar verificando se ao clicar no botão Pesquisar Pedidos o sistema redireciona para a tela de pesquisa com a lista de pedidos cadastrados, caso o usuário preencha o campo Nome o sistema deve retornar apenas os pedido daquele cliente.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Pesquisar Pedidos. Para testar o filtro basta preencher o campo nome.
Pré Requisito	Ter pedidos cadastrados.
Observação	Não há.

Tabela 29 – Caso de Teste: Pesquisar Pedidos

Caso de uso: Pesquisar Item do Pedido	
Objetivo	Acessar a tela de pesquisa de Itens verificando as informações se estão corretas, ao preencher o campo Nome Produto o sistema deve filtrar os itens pelo nome do produto.
Descrição	Para este teste é preciso acessar a tela de financeira e clicar no botão Pesquisar Pedidos, selecionar um pedido e clicar em conferir itens .
Pré Requisito	Não há
Observação	Não há

Tabela 30 – Caso de Teste: Pesquisar Itens do Pedido

Caso de uso: Editar Cliente	
Objetivo	Acessar a tela de pesquisa de clientes, selecionar um cliente da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Cliente alterado com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de clientes e clicar no botão Pesquisar Cliente, selecionar um cliente e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 31 – Caso de Teste: Alteração de Clientes

Caso de uso: Editar Categoria	
Objetivo	Acessar a tela de pesquisa de categorias, selecionar uma categoria da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Categoria alterada com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de categorias e clicar no botão Pesquisar Categoria, selecionar uma Categoria e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 32 – Caso de Teste: Alteração de Categorias

Caso de uso: Editar Subcategoria	
Objetivo	Acessar a tela de pesquisa de subcategorias, selecionar uma subcategoria da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Subcategoria alterada com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de subcategorias e clicar no botão Pesquisar Subcategoria, selecionar uma Subcategoria e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 33 – Caso de Teste: Alteração de Subcategorias

Caso de uso: Editar Produto	
Objetivo	Acessar a tela de pesquisa de produtos, selecionar um produto da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Produto alterado com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de produtos e clicar no botão Pesquisar Produto, selecionar um Produto e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 34 – Caso de Teste: Alteração de Produtos

Caso de uso: Editar Sessão	
Objetivo	Acessar a tela de pesquisa de sessões, selecionar uma Sessão da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Sessão alterada com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de sessões e clicar no botão Pesquisar Sessão, selecionar uma Sessão e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 35 – Caso de Teste: Alteração de Sessões

Caso de uso: Editar Compromisso	
Objetivo	Acessar a tela de pesquisa de compromissos, selecionar um Compromisso da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Compromisso alterado com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de compromissos e clicar no botão Pesquisar Compromisso, selecionar um Compromisso e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 36 – Caso de Teste: Alteração de Compromissos

Caso de uso: Editar Pedido	
Objetivo	Acessar a tela de pesquisa de pedidos, selecionar um Pedido da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Pedido alterado com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Pesquisar Pedido, selecionar um Pedido e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 37 – Caso de Teste: Alteração de Pedidos

Caso de uso: Editar Item do Pedido	
Objetivo	Acessar a tela de pesquisa de Itens, selecionar um Item da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Item alterado com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de financeira e clicar no botão Pesquisar Pedidos, selecionar um pedido e clicar em conferir itens e então clicar em um Item da lista e editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 38 – Caso de Teste: Alteração de Itens do Pedido

Caso de uso: Editar Meta	
Objetivo	Acessar a tela de pesquisa de metas, selecionar uma meta da lista e escolher a opção editar, ao redirecionar para a tela de cadastro verificar se os dados estão preenchidos corretamente e alterá-los, mandar salvar verificando se os dados inseridos estão de acordo com os dados no banco, verificar se ao inserir dados inválidos o sistema informa ao usuário o erro e também se caso ocorra algum erro na inserção o sistema também mostre ao usuário por meio de mensagem. A mensagem de sucesso deve ser a seguinte: Meta alterada com sucesso! A mensagem caso algum campo esteja faltando deve ser mostrada por meio de um ícone de exclamação nos campos obrigatórios, caso seja um erro deve ser mostrada como um popup na tela que desaparece sem algum clique.
Descrição	Para este teste é preciso acessar a tela de metas e clicar no botão Pesquisar Metas, selecionar uma meta e clicar em editar, inserir os dados e confirmar as saídas
Pré Requisito	Não há
Observação	Não há

Tabela 39 – Caso de Teste: Alteração de Metas

Caso de uso: Excluir Metas	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de metas e clicar no botão Pesquisar Metas, clicar em um item da lista e excluir.
Pré Requisito	Ter metas cadastrados.
Observação	Não há.

Tabela 40 – Caso de Teste:Excluir Metas

Caso de uso: Excluir Clientes	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de clientes e clicar no botão Pesquisar Clientes, clicar em um item da lista e excluir.
Pré Requisito	Ter clientes cadastrados.
Observação	Não há.

Tabela 41 – Caso de Teste:Excluir Clientes

Caso de uso: Excluir Categorias	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de categorias e clicar no botão Pesquisar Categorias, clicar em um item da lista e excluir.
Pré Requisito	Ter categorias cadastradas.
Observação	Não há.

Tabela 42 – Caso de Teste:Excluir Categorias

Caso de uso: Excluir Subcategorias	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de subcategorias e clicar no botão Pesquisar subcategorias, clicar em um item da lista e excluir.
Pré Requisito	Ter subcategorias cadastradas.
Observação	Não há.

Tabela 43 – Caso de Teste:Excluir Subcategorias

Caso de uso: Excluir Produtos	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de produtos e clicar no botão Pesquisar Produtos, clicar em um item da lista e excluir.
Pré Requisito	Ter produtos cadastrados.
Observação	Não há.

Tabela 44 – Caso de Teste:Excluir Produtos

Caso de uso: Excluir Sessões	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de sessões e clicar no botão Pesquisar Sessões, clicar em um item da lista e excluir.
Pré Requisito	Ter Sessões cadastrados.
Observação	Não há.

Tabela 45 – Caso de Teste:Excluir Sessões

Caso de uso: Excluir Compromissos	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de compromissos e clicar no botão Pesquisar Compromissos, clicar em um item da lista e excluir.
Pré Requisito	Ter compromissos cadastrados.
Observação	Não há.

Tabela 46 – Caso de Teste:Excluir Compromissos

Caso de uso: Excluir Pedidos	
Objetivo	Na tela de pesquisa, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela financeira e clicar no botão Pesquisar Pedidos, clicar em um item da lista e excluir.
Pré Requisito	Ter pedidos cadastrados.
Observação	Não há.

Tabela 47 – Caso de Teste:Excluir Pedidos

Caso de uso: Excluir Itens do Pedido	
Objetivo	Na tela de pesquisa dos itens do pedido selecionado, clicar em um item e na opção excluir item, verificando se o sistema mostra a mensagem de confirmação para o usuário, após a exclusão, verificar se a lista foi atualizada.
Descrição	Para este teste é preciso acessar a tela de Pesquisar Pedido e conferir itens, clicar em um item da lista e excluir.
Pré Requisito	Ter itens cadastrados no pedido.
Observação	Não há.

Tabela 48 – Caso de Teste:Excluir Itens do pedido