

ANÁLISE DE REQUISITOS PARA UM SOFTWARE PARA REPRESENTANTES COMERCIAIS - CATÁLOGO VIRTUAL DE VESTUÁRIO

Gustavo Croisfelt Pereira
Graduando em Engenharia de Software – Uni-FACEF.
gustavo.croisfelt@gmail.com

Prof. Me. Carlos Alberto Lucas
Mestre em Educação
projeto@profcarloslucas.com.br

RESUMO

Este artigo trata de questões problemas relacionados aos processos, técnicos e profissionais, de um representante comercial de roupas. Tal representante comercial trabalha com uma pesada carga de peças de roupas utilizando alguns mostruários físicos, o que acaba por atrapalhar a eficiência dos atendimentos e o deslocamento do próprio representante. Um outro fator é a utilização de planilhas em Excel, para a formatação de pedidos muitas vezes realizados manualmente através da inserção de dados como código das roupas. E foi observando o fluxo de trabalho deste representante comercial de vestuário feminino, que surgiu a ideia e foi criado um catálogo digital para facilitar a apresentação dos produtos, controle e gerenciamento das transações comerciais. A solução desenvolvida em formato mobile para smartphones e tablets busca a digitalização de um catálogo com as peças de roupas de determinada coleção, onde no mesmo Software se dá também a criação da planilha conectada diretamente com a linha de pedidos da fábrica. Tendo em foco a análise de requisitos para o desenvolvimento de um Software eficiente e ágil foram usadas técnicas de Engenharia de Software para o levantamento de requisitos e a análise do processo de atendimento de um profissional da área de representação.

Palavras-chave: Representante Comercial. Vestuário. Aplicação Mobile. Pedidos.

ABSTRACT

The sales representative works with a high load of garments using physical displays, which ends up hindering the efficiency of the service and the displacement of the seller. Spreadsheets are used to format orders often made manually by entering data as a clothing code. Observing the workflow of a female clothing sales representative looking for a way to reduce the physical load represented in this case by the parts showcase and greater agility in the formatting of orders placed in factories, a solution in the form of mobile Software is relevant. for such sellers. The solution to be developed in mobile format for smartphones and tablets seeks to digitize a catalog with the clothing items of a certain collection where the same Software also creates the worksheet directly connected to the factory order line. With a focus on the analysis of requirements for the development of a efficient and agile Software, Software engineering techniques were used to raise requirements for analyzing the process of attending a professional in the representation area. The system will be easy to handle for the representative and the factory.

Keywords: Commercial Representative. Clothing. Mobile application. Requests.

1 INTRODUÇÃO

Analisando o dia a dia de um representante comercial, verificamos que duas das principais ferramentas que este profissional utiliza é o mostruário de peças das empresas as quais ele está vinculado e uma planilha de controle, criada no Excel. Com o mostruário, o representante apresenta os produtos ao cliente que após analisar as peças de roupas, formaliza o pedido.

Tal pedido é registrado na planilha do Excel, que possui uma quantidade relativamente alta de ‘campos/dados’ a serem preenchidos sobre o cliente e as referências das roupas que compõem o pedido. Constatamos também a necessidade e a importância do mostruário e da planilha ao representante comercial, pois são utilizadas em conjunto para que haja controle na formalização do pedido e a consequente transparência aos atores envolvidos: o cliente; o fabricante de roupas; e o próprio representante comercial.

Com este cenário e processos observados, com a maturidade adquirida como aluno de Engenharia de Software e principalmente, com as vantagens e funcionalidades proporcionadas pela Tecnologia da Informação, surgiram alguns questionamentos: Como podemos diminuir a carga de um mostruário físico de roupas? Como melhorar o processo de formalização de um pedido? Como facilitar tais processos sem comprometer a qualidade do atendimento do representante? Como facilitar a relação comercial entre os atores representante, cliente e fabricante?

Objetivando alcançar a resposta dos questionamentos é preciso manter a atenção e o foco, pois sabemos que a primeira regra para utilizar qualquer tecnologia nos negócios é que a automação aplicada a uma operação eficiente aumentará a eficiência. A segunda é que a automação aplicada a uma operação ineficiente aumentará a ineficiência. (GATES, 1995).

Foi baseado nos questionamentos, que apresentamos uma documentação consistente e detalhada sobre alguns artefatos da engenharia de Software, para que seja desenvolvido um aplicativo funcional que irá colaborar com os processos do representante comercial.

A justificativa é substituir os grandes e pesados mostruários de roupas, por um catálogo digital de alta definição de qualidade das peças do vestuário, bem como o gerenciamento de todos os dados referentes às peças de roupa, os trâmites de cada negociação e, consequentemente, um processo mais eficaz e seguro.

O objetivo deste artigo é apresentar algumas etapas dos requisitos de uma aplicação

móvel, capaz de incorporar um catálogo de produtos de vestuários de várias empresas fornecedoras/fabricantes. Em seguida, com o catálogo digital completo, o representante poderá cadastrar no aplicativo e incluir os seus clientes (lojas físicas).

Através dessa inclusão, o representante conseguirá iniciar o processo de efetivação do pedido do cliente e acompanhar a sua finalização, por exemplo, após informar o fornecedor sobre o pedido efetivado, o representante poderá verificar o status deste pedido durante o processamento de execução da fabricação das peças até o envio ao destino final, incluindo a geração da forma de pagamento optada pelo cliente.

A metodologia utilizada para execução deste artigo foi a etnografia como técnica para levantamento de requisitos, a aplicação de entrevistas com um representante comercial e pesquisa em referenciais bibliográficos para embasamento técnico e científico.

Para elaboração do artigo e aplicativo foram utilizadas várias ferramentas: Scrum, Heflo, Visual Paradigm e demais recursos para definir datas e organizar as etapas de entrega do projeto. Na documentação foram usadas ferramentas para gerar o diagrama de Casos de Uso e Matrizes de Rastreabilidade.

Para validação referencial, são apresentados conceitos teóricos sobre o uso da Engenharia de Software (ES) para o desenvolvimento de uma solução e a importância da eficiência nessa linha de trabalho de representação.

Na terceira etapa, são apresentados os processos para o desenvolvimento da solução proposta por meio de técnicas da Engenharia de Software, como alguns dados do levantamento de requisitos, interface de usuário, cuidado com os riscos e testes e os resultados obtidos. E por fim, na última etapa são apresentados alguns artefatos da solução implementada com imagens do sistema.

2 REFERENCIAL TEÓRICO

No tópico a seguir todos os referenciais teóricos serão abordados para o entendimento dos requisitos necessários para uma aplicação voltada a representantes comerciais.

2.1 Representação Comercial

O representante comercial é o Ator principal na negociação entre o fabricante e o comprador. O seu papel é de extrema relevância chegando ao protagonismo, pois é o responsável por fazer a diferença nas vendas. Cada negócio tem sua forma de vender e de relacionar-se com o cliente. Mas as empresas que contam com representantes comerciais sabem

que a relação precisa ser saudável e eficiente para quem vende e quem compra. (TURCATO, 2019).

Este profissional atua diretamente na venda de produtos ou serviços de uma empresa, podendo atuar de duas formas contratuais: como colaborador contratado ou autônomo. Na primeira forma, ele terá um vínculo fixo com a empresa através da Consolidação das Leis do Trabalho (CLT), já na segunda é através do registro junto ao Imposto Sobre Serviços (ISS) municipal podendo prestar serviços para mais de uma empresa.

Sua remuneração é composta, normalmente, em uma parte de salário fixo e uma parte em comissão de vendas, o que gera uma variação nos ganhos. Tendo a remuneração neste formato, o trabalhador dependerá de seu desempenho em alcançar as metas e objetivos propostos.

Neste segmento profissional existe uma confusão entre a profissão de representante e a de vendedor. As duas possuem suas similaridades, entretanto o vendedor sempre terá ligação a uma empresa e pode atuar tanto com vendas externas e internas.

O representante não precisa ter tal conexão e de maneira geral atua no campo conhecido como “*field sales* = vendas de campo”. De acordo com Gates (1995), as atividades que o representante comercial precisa desempenhar são:

- Vender produtos ou serviços para clientes novos e antigos;
- Entender como abordar clientes e gerar valor para eles desde o primeiro momento;
- Realizar atendimentos aos clientes antigos e manter um relacionamento saudável com eles;
- Participar de reuniões de vendas periódicas e prestar treinamento sobre o que é vendido;
- Planejar bem sua rotina comercial e prestar contas daquilo que é feito;
- Ter um grande poder de persuasão para contornar objeções e fechar negócios complicados;
- Ser didático e consultivo na hora de apresentar o produto ou serviço que é vendido;
- Ser o ponto de contato entre clientes e empresa de maneira rápida e eficiente;
- Resolver problemas que atrapalham a experiência do cliente com a marca.

Por tudo isto, é essencial que ele tenha um bom controle na gerência e eficiência de suas vendas, tendo o status de suas negociações e o cadastro de clientes, sempre atualizados.

2.2 A Engenharia de Software

Engenharia de Software é uma área da Engenharia de Sistemas totalmente dedicada para a Gestão de Projetos, como: projetar, implementar e modificar o Software, de forma que ele seja de alta qualidade, a um custo razoável, manutenível e rápido de construir. (LAPLANTE, 2007). Esta engenharia nos proporciona diversas ferramentas que facilitam e otimizam o desenvolvimento da solução de Software.

O engenheiro de Software define a especificação dos processos de planejamento, realizando o gerenciamento e a produção da solução em sua implementação. Ele acompanha e avalia o desenvolvimento de todos os projetos da empresa para verificar se o processo estabelecido está sendo executado de forma eficiente e efetiva. Caso sejam necessárias mudanças no processo estabelecido, ele as identifica e realiza, garantindo que a equipe adote tais mudanças. O engenheiro também reavalia o processo continuamente. (WAZLAWICK, 2013).

A engenharia de Software se destina no apoio ao desenvolvimento de Software profissional, em vez de atuar na programação individual. Ela inclui técnicas que apoiam a especificação, o projeto e a evolução do Software. (SOMMERVILLE, 2019).

Apresenta um conjunto de princípios que devem ser seguidos e aplicados na construção de um projeto de desenvolvimento de Software. Não são necessariamente regras, mas são orientações como uma forma de filosofia de desenvolvimento.

De acordo com Wazlawick (2013), estes são alguns princípios: Decomposição; Abstração; Generalização; Padronização; Flexibilização; Formalidade; Rastreabilidade; Desenvolvimento Interativo; Gerenciamento de Requisitos; Arquitetura Baseadas em Componentes; Modelagem Visual; Verificação Contínua de Qualidade; Controle de Mudanças; Gerenciamento de Riscos.

Neste projeto, consideramos tais princípios para que a solução proposta fosse adequada aos questionamentos propostos, como exemplo, a modelagem visual no formato de *Business Process Model and Notation* (BPMN) e *Use Case* e a citação dos requisitos do projeto.

O controle de mudanças é realizado através do método ágil Scrum, utilizado como forma de organização das tarefas a serem conduzidas. Todas as etapas foram planejadas visando um Software com funcionalidades e desempenho, independentemente da manutenibilidade e dependência.

A Dependabilidade é a capacidade do sistema computacional de prestar um serviço justificável e que engloba a confiabilidade e a segurança.

Em geral, os engenheiros de Software adotam uma abordagem sistemática e organizada para o trabalho, já que muitas vezes essa é a maneira mais eficaz de produzir Software de alta qualidade. Entretanto, a engenharia trata de escolher o método mais adequado para um conjunto de circunstâncias, de modo que uma abordagem mais criativa e menos formal de desenvolvimento, pode ser mais adequada para alguns tipos de Software. (SOMMERVILLE, 2019).

2.3 Métodos Ágeis

No desenvolvimento da solução proposta, para meios de organização e eficiência, foi utilizada a metodologia ágil Scrum. A filosofia desses métodos pode ser vista no ‘manifesto ágil’ (<http://agilemanifesto.org>): Estamos descobrindo maneiras melhores de desenvolver Softwares, fazendo-o nos mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

Indivíduos e iterações mais que processos e ferramentas;
Software em funcionamento mais que uma documentação abrangente;
Colaboração com o cliente mais que negociação de contratos;
Responder a mudanças mais que seguir um plano;
Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

O Scrum é um método que segue as diretrizes do manifesto ágil. Sua terminologia consiste em 7 itens que são:

- O Time de Desenvolvimento responsável pelo desenvolvimento do Software e documentos indispensáveis para o projeto;
- Incremento de produto que é entregue a partir de uma *sprint*, nesse item o produto em questão deve ser ‘potencialmente entregável’, ou seja, não é necessário trabalhos adicionais para incluí-lo no produto final;
- Backlog que é uma lista de tarefas que o time responsável deve realizar que abrange várias etapas do projeto;
- O Product Owner ou dono do produto como o próprio nome diz também pode ser o cliente, o gerente ou o representante de um *stakeholder*;

- A reunião desse método que leva o nome do próprio Scrum é realizada diariamente e todo o time deve estar presente e tem como prioridade as atividades a serem realizadas naquele dia;
- O Scrum Master que é o responsável pelo time e deve seguir o processo do Scrum, fazendo com que a equipe não perca o foco, e forneça a interação com o restante da empresa;
- A velocidade que determina como as *sprints* podem ser divididas e gerenciadas para um melhor desempenho da equipe.

Este projeto apresentou como Scrum Master o prof. Flávio Henrique de Oliveira Costa, sendo as reuniões realizadas uma vez por semana. Como o ciclo do *sprint* do Scrum se inicia pelo backlog do projeto, a lista inicial do backlog deste projeto foi derivada do Termo de Abertura do Projeto (TAP) com as ideias iniciais e a conclusão de cada tarefa mensal, sendo que outras etapas foram adicionadas ao backlog pelo Scrum Master durante o processo.

2.4 Verificação, Validação e Testes

Segundo a Norma NBR ISSO/IEC 9126-1:2003 é definido qualidade de Software como a “[...] totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas”.

Assegurar sua qualidade se torna essencial para a entrega de um produto polido e satisfatório ao cliente e que com isso, seja possível a melhor experiência e eficiência possível no produto.

Tendo isto em vista, se tem o uso de três etapas distintas que podem ser realizadas separadamente ou independentes são elas: Verificação; Validação; e Testes.

Segundo Wazlawick (2011) existe uma distinção que convém ser feita entre as três:

- **Verificação:** consiste em analisar o Software para ver se ele está sendo construído de acordo com o que foi especificado;
- **Validação:** consiste em analisar o Software construído para ver se ele atende as verdadeiras necessidades dos interessados;
- **Teste:** é uma atividade que permite realizar a verificação do Software.

Dessa forma, a validação e a verificação podem ser resumidas em perguntas, “Estamos fazendo a coisa certa?”; “Estamos fazendo a coisa do jeito certo?”, respectivamente.

A validação se incumbe de saber se os requisitos do Software refletem efetivamente ao

que foi pedido pelo cliente/usuário. Já a verificação atende tais requisitos da forma mais perfeita possível, que esteja livre de defeitos e apresente características de qualidade definidas anteriormente.

Os níveis de teste são usados para estabelecer se a função está diferente dos objetivos para as atividades de testes. Tais objetivos em sua grande maioria está diretamente relacionada a verificação. No caso da validação, apenas o teste de aceitação é feito para tal fim.

Mesmo tendo tudo isto em vista, realizar o teste de um Software não é simples, em muitos casos é mais difícil fazer bons casos de testes do que produzir o próprio Software. Dessa forma, ter muito controle e cuidados são necessários para que o teste de Software deixe de ser uma atividade ingênua, para que se torne uma atividade com resultados efetivos e previsíveis.

De acordo com Wazlawick (2011) é preciso utilizar a verificação, a validação e o teste como uma forma de evitar outros 03 (três) problemas que são:

- Defeito: É uma deficiência mecânica que pode ser uma linha de código ou conjunto de dados que vão provocar um erro;
- Erro: É uma execução diferente do resultado esperado ou correto;
- Falha: É quando o Software não funciona, provavelmente causada por um defeito, mas com outras possíveis causas.

Além destes três problemas, há também o “*mistake* = engano” que seria o erro humano.

Assim, compilando tudo isto, se tem os níveis de testes de funcionalidade. Como os objetivos de testes tendem para uma verificação se as funções e suas implementações são validadas, perante os requisitos apresentados pelo cliente. Esta classificação para tais testes é: funcionalidade; integração; ciclo de negócio; aceitação; e operação. E os testes suplementares como: performance; segurança; e tolerância a falhas.

Nas técnicas funcionais temos o teste de Caixa Preta que se baseia nas especificações do Software para derivar requisitos, com a função de enxergar o Software de um ponto de vista macro. Este processo envolve duas formas principais, que são: identificar as funções do Software; e criar casos de teste para checar se as funções estão sendo realizadas corretamente.

Existe também o teste de Caixa Branca que faz parte da técnica estrutural que vai se basear na estrutura interna do problema, ou seja, sua implementação. Testa os detalhes do procedimento e a maioria dos critérios dessa técnica utiliza uma representação de programa conhecida como grafo de programa ou grafo de fluxo de controle.

3 ANALISANDO OS REQUISITOS DA SOLUÇÃO

Logo após a escolha do tema a ideia foi focar na análise de vários requisitos da engenharia de software para a construção da solução proposta.

3.1 Colocando em prática a Verificação, Validação e Testes

Conforme mostrado na Tabela 1 é realizada a verificação e validação para os processos de documentação. Para exemplificar o processo as cores verde, amarelo e vermelho são usadas nas etapas dos processos sendo respectivamente concluído, em andamento, e ainda não checada respectivamente.

Esse exemplo se utiliza os próprios artefatos e processos da engenharia de software

Tabela 1 — Controle das tarefas de um TCC.

Artefato	Atividade	Responsável	Data Avaliação	Técnica e Estratégia
Discussão e Definição do Funcionamento do Software	Verificação e Validação	Carlos Alberto Lucas	Terminado	Brainstorming, definição das funções e estruturas e conversação
Definição das estruturas das telas	Verificação e Validação	Gustavo Croisfelt Pereira	Terminado	Brainstorming e Conversação
Levantamento de Requisitos	Verificação e Validação	Carlos Alberto Lucas	Terminado	Brainstorming e Conversação
Documento de BPMN	Verificação e Validação	Carlos Alberto Lucas	A ser avaliado pelo Orientador	Deve respeitar as normas de montagem de BPMN e mostrar claramente a estrutura de funcionamento do Software
Documento 5W1H	Verificação e Validação	Carlos Alberto Lucas	A ser avaliado pelo Orientador	Deve possuir as perguntas e respostas definidas para o desenvolvimento do Software
Documento de Caso de Uso	Verificação e Validação	Carlos Alberto Lucas	A ser avaliado pelo Orientador	Deve respeitar as normas de montagem de Casos de Uso e mostrar claramente a estrutura de funcionamento do Software
Telas do aplicativo	Verificação e Validação	Carlos Alberto Lucas	Quando o desenvolvimento de Telas Iniciar	Devem estar de acordo com a ideia do projeto
API do Projeto	Verificação e Validação	Carlos Alberto Luvas	Assim que todas as etapas do TCC estiveram concluídas	Deverá funcionar de forma conjunta e fluida com as funcionalidades das telas

Fonte: O autor.

Nas Tabelas 2 a 13 são apresentados os testes a serem realizados nas funcionalidades das telas deste projeto. Inicialmente as tabelas foram de testes básicos e em seguida, os testes mais complexos com INPUT e OUTPUT.

Tabela 2 — Testes Tela do Catálogo.

Contador:	001
Localização:	Tela Catálogo.
Criticidade:	Alta.
Objeto de Teste:	Verificar as mudanças de páginas do catálogo.
Caso de Teste:	Testar o funcionamento do deslizamento da tela para troca de páginas.
Pré – Condição	1. Catálogo devidamente localizado no app.
Procedimento	1. Clicar na opção catálogo da página inicial.

	<ol style="list-style-type: none"> 2. Deslizar as imagens do catálogo para a esquerda para avançar a imagem seguinte. 3. Deslizar as imagens do catálogo para direita para retornar a imagens anterior. 4. Deslizar até o fim das páginas do catálogo digital. 5. Depois da última página retorna ao início do catálogo.
Resultado Esperado:	<ol style="list-style-type: none"> 1. O aplicativo abre o catálogo de produtos 2. Ao deslizar a imagem para a esquerda uma outra imagem na ordem do catálogo irá aparecer. 3. Ao deslizar a imagem para a direita a imagem anterior do catálogo irá aparecer. 4. O aplicativo deve mostrar todas as imagens do catálogo sem repetições. 5. Chegando ao fim do catálogo a próxima imagem deverá ser a do início.

Fonte: O autor.

Tabela 3 — Testes Tela Cadastro do Cliente.

Contador:	002
Localização:	Tela Cadastro do Cliente.
Criticidade:	Alta.
Objeto de Teste:	Verificar o funcionamento do botão finalização de cadastro do cliente.
Caso de Teste:	Testar a inclusão do cadastro e sua finalização.
Pré-Condição	1. Dados do cliente devidamente preenchidos.
Procedimento	<ol style="list-style-type: none"> 1. Clicar na opção “Cadastro do Cliente” da página inicial. 2. Digitar os dados necessários para o cadastro. 3. Em todos os campos digite uma entrada invalida para tal campo. Exemplo: no CNPJ digitar uma letra. 4. Clicar no botão “Cadastrar” sem digitar uma entrada obrigatória. 5. Clicar no botão “Cadastrar” com todos os dados corretos.
Resultado Esperado:	<ol style="list-style-type: none"> 1. O aplicativo abre a página de cadastro do cliente. 2. O aplicativo deve indicar dados obrigatórios. 3. O aplicativo irá indicar onde está a entrada errada e não permitirá o cadastro. 4. O aplicativo irá indicar qual entrada de dados está faltando. 5. O aplicativo exibira uma mensagem de Cadastro realizado com sucesso.

Fonte: O autor.

Tabela 4 — Testes Tela Login.

Contador:	003
Localização:	Tela Login.
Criticidade:	Alta.
Objeto de Teste:	Verificar as funcionalidades da tela de login.

Caso de Teste:	Testar cada campo localizado na tela de login.
Pré-Condição	1. A tela de login ter sido efetivamente criada.
Procedimento	<ol style="list-style-type: none"> 1. Inserir o usuário de teste no campo “Usuário”. 2. Inserir a senha de teste no campo “Senha”. 3. Clicar no botão “Entrar” 4. Utilizar a função “Esqueceu o usuário?” 5. Utilizar a função “Esqueceu a senha?” 6. Checar o botão “Cadastrar”.
Resultado Esperado:	<ol style="list-style-type: none"> 1. O usuário teste e visto escrito no campo “Usuário. 2. A senha e vista de forma protegida demonstrando somente os números de caracteres. 3. Ao clicar em “Entrar” abre a tela do menu principal. 4. Muda para a tela de recuperação de usuário/senha 5. Muda para a tela de recuperação de usuário/senha 6. Abre a tela de Cadastro do Representante Comercial.

Fonte: O autor.

Tabela 5 — Testes Tela Menu Inicial.

Contador:	004
Localização:	Tela Menu Inicial.
Criticidade:	Alta.
Objeto de Teste:	Botões do menu inicial.
Caso de Teste:	Testar a mudança de tela do menu inicial através dos botões.
Pré-Condição	1. O menu e as telas que os botões encaminham devem estar minimamente implantadas.
Procedimento	<ol style="list-style-type: none"> 1. Clicar no primeiro botão do Menu intitulado de “Catálogo de Roupas”. 2. Clicar no segundo botão do Menu intitulado de “Área do Cliente”.
Resultado Esperado:	<ol style="list-style-type: none"> 1. O botão abre a tela do Catálogo. 2. O botão abre a tela dos dados dos clientes do representante.

Fonte: O autor.

Tabela 6 — Testes Tela Área do Cliente.

Contador:	005
Localização:	Tela Área do Cliente.
Criticidade:	Alta.
Objeto de Teste:	Tela dos dados do cliente do representante.
Caso de Teste:	Checar e testar se todas as funcionalidades dessa tela estão funcionando e presentes.
Pré-Condição	1. Tela devidamente implantada.
Procedimento	<ol style="list-style-type: none"> 1. Checar botão de “Cadastro do Cliente”. 2. Checar botão de “Pedidos”. 3. Checar botão “Clientes”.
Resultado Esperado:	1. Abre a tela de cadastro do cliente.

	<ol style="list-style-type: none"> 2. Abre a tabela com os pedidos realizados pelo cliente. 3. Abre a tela de clientes já cadastrados e seus dados de cadastro.
--	---

Fonte: O autor.

Tabela 7 — Razão Social.

INPUT	OUTPUT
NULL	Campo Obrigatório.
1234	Alerta1 = Entrada Inválida.
Loja Teste	Entrada Válida.
Loja Teste	Alerta2 = Entrada já existe deseja substituir ou criar outra entrada com mesmo nome?

Fonte: O autor.

Tabela 8 — CNPJ.

INPUT	OUTPUT
NULL	Campo Obrigatório.
1234	Alerta3 = Número de caracteres incorreto.
Loja Teste	Alerta1 = Entrada Inválida.
11111111111111	Entrada Válida.
11111111111111	Alerta4 = CNPJ já existente.
111111B1111111	Erro = Entrada Inválida caractere Incorreto.

Fonte: O autor.

Tabela 9 — Endereço.

INPUT	OUTPUT
NULL	Campo Obrigatório.
1234	Alerta1 = Entrada Inválida.
Rua Teste 1234	Entrada Válida.
11111111111111	Entrada Inválida.
Teste	Alerta5 = Necessita especificar Rua/Avenida e número.
Rua Te1ste 1234	Erro = Entrada Inválida caractere Incorreto.

Fonte: O autor.

Tabela 10 — CEP.

INPUT	OUTPUT
NULL	Campo Obrigatório.
1234	Alerta3 = Número de caracteres Incorreto.
Rua Teste 1234	Alerta1 = Entrada Inválida.
11111111	Entrada válida.
111h11f11	Erro = Entrada Inválida caractere Incorreto.

Fonte: O autor.

Tabela 11 — Pessoa Responsável.

INPUT	OUTPUT
NULL	Campo Obrigatório.
1234	Alerta1 = Entrada Inválida.
Rua Teste 1234	Alerta1 = Entrada Inválida.
11111111	Alerta1 = Entrada Inválida.
Gustavo Croisfelt Pereira	Entrada Válida.
Gustavo Cr0isfelt Pereira	Erro = caractere Incorreto.

Fonte: O autor.

Tabela 12 — Tela de Catálogo.

INPUT	OUTPUT
Deslizar a tela para esquerda	Abre a próxima imagem.
Deslizar a tela para a direita	Abre a imagem anterior.
2 toques na imagem	Abre Informação do Código do Produto e a opção de adicionar ao pedido.
Clicar Botão Adicionar	A roupa e adicionada a tabela de pedidos do cliente
Clicar Botão Adicionar em uma imagem inválida	Erro = Imagem Inválida.
Deslizar o catálogo de imagens até o fim	Imagens sem repetição e retorna à primeira imagem do catálogo.

Fonte: O autor.

Tabela 13 — Tela de Login.

INPUT	OUTPUT
Deslizar a tela para esquerda	Abre a próxima imagem.
Deslizar a tela para a direita	Abre a imagem anterior.
2 toques na imagem	Abre Informação do Código do Produto e a opção de adicionar ao pedido.
Clicar Botão Adicionar	A roupa e adicionada a tabela de pedidos do cliente.
Clicar Botão Adicionar em uma imagem inválida	Erro = Imagem Inválida.
Deslizar o catálogo de imagens até o fim	Imagens sem repetição e retorna à primeira imagem do catálogo.

Fonte: O autor.

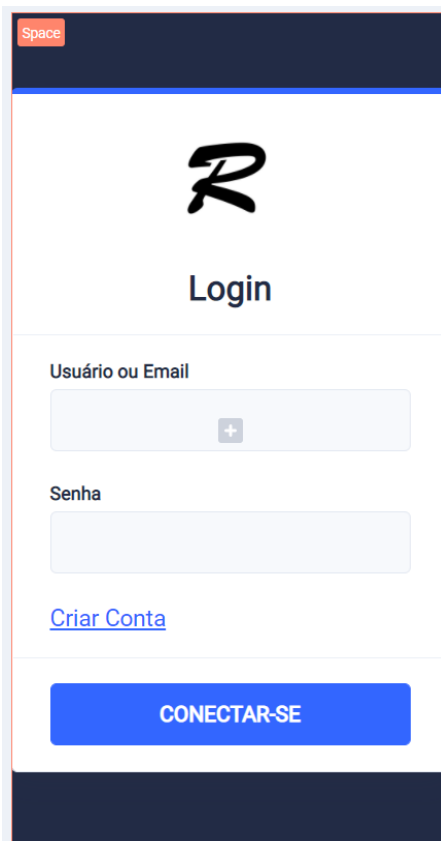
Para exemplificar os testes seguem a seguir exemplos simples das telas a serem testadas. A figura 1 representa uma tela de login para o usuário com duas áreas para input do nome do usuário ou do email utilizado no cadastro, outra área para a senha cadastrada. Logo abaixo um botão para direcionar para a criação da conta no aplicativo e em seguida o botão para entrar no aplicativo.

A figura 2 demonstra a área do perfil do cliente destinada ao cadastro dos dados e a

utilização de uma imagem do cliente pessoa ou da empresa cliente.

A terceira tela representada seria a tela do catálogo na figura 3 que seria uma imagem do produto que cobre todo o espaço do aplicativo e funciona com um sistema de paginas por deslizamento “touch”.

Figura 1 — Login.



The image shows a mobile application login screen. At the top, there is a dark blue header with the word "Space" in a small red box. Below the header, the screen is white. In the center, there is a large, stylized black letter "R" logo. Below the logo, the word "Login" is written in a bold, black font. Underneath, there are two input fields: the first is labeled "Usuário ou Email" and has a small grey plus icon on the right; the second is labeled "Senha". Below the password field, there is a blue link that says "Criar Conta". At the bottom of the form area, there is a large blue button with the text "CONECTAR-SE" in white capital letters. The entire screen is framed by a dark blue border at the top and bottom.

Fonte: O autor.

Figura 2 — Perfil do Cliente.

The image shows a web form titled "Perfil do Cliente". On the left, there is a circular profile picture placeholder with a blue background and a white icon of a person. Below it is a "Choose File" button. The form contains several input fields: "Pessoa Responsável", "Endereço", "CEP", "Nome/Loja", "Email", "Razão Social", "Telefone", and "CNPJ". A blue "CADASTRAR" button is located at the bottom right of the form.

Fonte: O autor.

Figura 3 — Catálogo.



Fonte: O autor.

3.2 Riscos

Um risco normalmente é composto de causa e problemas. A causa vem na forma de uma

incerteza e os problemas que possam ocorrer será em função da causa, assim provocando um impacto em uma ou mais partes do projeto.

Os riscos de gerenciamento do projeto envolvem a capacidade da equipe de planejar e seguir o plano dentro do cronograma e do custo previstos. (WAZLAWICK, 2011).

As tabelas 14, 15 e 16 abaixo demonstram um plano de mitigação de riscos para os problemas que podem surgir no desenvolvimento da aplicação. Cada um dos possíveis cenários e identificado pelas siglas A1, A2, A3, A4 e A5

Tabela 14 — Tabela de Riscos.

Id	Causa
A1	Diferentes Fabricas podem usar diferentes dados de clientes. A1
A2	Necessidade de entendimento da ferramenta pelas Fabricas e pelos Representantes. A2
A3	O projeto dependera de duas “fontes” Fabrica e Representantes. A3
A4	Possivel nao integração entre o catalogo de imagens e a tabela de pedidos. A4
A5	Funcionalidade depedende de dados gerados pela Fábrica(Produtos, Catálogos). A5
Risco	
	Pode haver mudanças no desenvolvimento do projeto dependendo das diferentes fabricas. A1
	Pode haver problemas no ensinamento da ferramenta em ambas as partes. A2
	Pode haver falha de comunicação entre as duas partes. A3
	Talvez as duas partes tenha de ser separadas dentro da aplicação. A4
	Atrasos nos produtos da fabrica ou nao entendimento da ferrementa. A5
Efeito	
	Maior tempo de desenvolvimento para abranger maior numero de clientes. A1
	Maior tempo para implementar a ferramenta efetivamente. A2
	Incoerência dos pedidos feitos. A3
	Ferramenta nao vai ser tão eficiente quanto previsto. A4
	Torna a ferramenta inutilizavel pelo representante comercial. A5

Fonte: O autor.

Tabela 15 — Tabela de Riscos.

Plano de redução de probabilidade
Desenvolver uma forma de facil mudança de dados para abranger todas as fabricas. A1
Desenvolver um ensino simples que esclarece todas as funcionalidades da ferramenta. A2
Gerar uma ferramente de comunicação entre as duas partes. A3
Planejar o desenvolvimento para ambas as possibilidades. A4
Gerenciar datas limites onde pelo menos o catalogo e divulgado. A5
Plano de redução de impacto
Elaborar um interface que permita a adição e subtração de dados para formatação da tabela. A1
Verificar se existe uma forma de melhorar a comunicação entre as partes para o ensino. A2
Projetar uma interface que evite isso. A3
Caso sejam separadas desenvolver de uma forma que seja tão quanto eficiente. A4
Prever datas a serem cumpridas pelo utilizador da ferramenta. A5

Fonte: O autor.

Tabela 16 — Tabela de Riscos.

P	I	E	
A	A	A	A1
M	A	A	A2
M	A	A	A3
B	M	B	A4
B	A	A	A5

Fonte: O autor.

3.3 5W2H

Normalmente, conhecido por essa sigla se trata de uma checklist de atividades com foco na solução proposta que devem ser seguidas para o desenvolvimento e clareza do projeto. As siglas vêm do Inglês 5W que se referem: What (o que será feito?); Why (por que será feito?); Where (onde será feito?); When (quando?); Who (por quem será feito?). E o 2H se referem: How (como será feito?); e How Much (quanto vai custar?). No caso deste projeto por não ter implicações financeiras, adotamos uma forma modificada, sendo ela o 5W1H, excluindo-se a How Much.

Planilha 5w1h							
Nº	Descrição inicial	What O que	Why Porque	Where Onde	Who Quem	When Quando	How Como
2	Demonstração dos Produtos da Coleção	Disponibilização do Catálogo da Coleção no Software	Facilitar a demonstração dos produtos para o cliente do representante	No Menu do Software sobre a aba de nome Catálogo	Gustavo	30/jun.	Pela Inserção das Imagens do Catálogo
		Facilidade no manuseio do catálogo digital	Tornar o aplicativo fácil de manusear em uma das suas principais funcionalidades fazendo o mesmo mais atrativo a ser utilizado	Na interface de manuseio do catálogo	Gustavo	15/jul.	Fazendo um design simples que seja eficiente e rápido
		Fácil busca através do código do produto	O Aplicativo deve possuir uma forma de busca pois os catálogos tendem a ser extensos	Na interface de manuseio do catálogo	Gustavo	20/jun.	Incorporando um sistema de buscas na UI
2	Diferentes Fábricas podem usar diferentes dados de clientes	Padronizar a Interface onde sejam possíveis alterações pontuais	A padronização facilita uma maior abrangência de fabricas e algumas alterações pontuais o torna flexível	Em todas as interfaces do Software	Gustavo	30/jun.	Fazendo um a interface simples e de fácil manuseio
		Facilitar a incorporação dos diferentes dados no Software	Com essa incorporação se facilita um atendimento mais amplo	No banco de dados do sistema	Gustavo	15/jul.	Com INSERTs no Banco de Dados
		Tentar padronizar os dados das fabricas dentro do Software	Facilita o desenvolvimento do aplicativo	No aplicativo em geral e em comum acordo com as fábricas	Gustavo	20/jun.	Padronizando os dados do BD de acordo com as fábricas
2	Necessidade de entendimento da ferramenta pelas Fábricas e pelos Representantes	Ensinar o funcionamento da ferramenta	Entender uma ferramenta de trabalho e de extrema importância ainda mais uma que vai facilitar o trabalho	Na própria fábrica	Gustavo	30/jun.	Demonstrando através de apresentações e a utilizando
		Cria Incentivos para o uso	Tendo uma forma de "recompensa" e visto como uma influência para usar a ferramenta	Na propria fábrica	Fábrica	15/jul.	Através de bonificações extras

Planilha 5w1h							
Nº	Descrição inicial	What O que	Why Porque	Where Onde	Who Quem	When Quando	How Como
1		Compartilhar a eficiência e a facilidade da ferramenta	Tendo acesso a esses dados o representante tem noção do que a ferramenta e capaz	A fábrica irá compartilhar esses dados	Gustavo	20/jun.	Demonstrando através do canal de comunicação correspondente da Fábrica
2	O projeto dependerá de duas “fontes” Fábrica e Representantes	Facilitar o upload do catálogo de produtos	Quanto mais rápido e cedo o catálogo ficar disponível mais cedo o representante pode iniciar suas vendas	No Software	Gustavo/Fábrica	30/jun.	Facilitar para a fábrica como fazer o upload do catálogo
		Gerar o pedido dentro do aplicativo	Isso irá agilizar o processo de atendimento	No Software	Gustavo	15/jul.	Criar a funcionalidade de gerar o pedido dentro do aplicativo
		Agilizar o processo e entrega dos pedidos	Agilizando o processo de entregas através do aplicativo abre espaço para adquirir novos clientes	Fábrica/Representante	Fábrica	20/jun.	Tendo acesso ao pedido quase que no mesmo momento que o atendimento a Fábrica pode agilizar o processamento do pedido

3.4 Levantamento de Requisitos:

Para o conhecimento do que a solução vai precisar foi feito um levantamento de requisitos com o cliente representante.

Requisitos Funcionais:

- RF01: A página de inicialização deve pedir o login e a senha do representante.
- RF02: Se o representante não tiver se cadastrado no Software deve ter a opção de fazer cadastro.
- RF03: O menu deve ter o catálogo da coleção mais atual.
- RF04: A lista de clientes onde pode ser adicionado ou excluído tal cliente.
- RF05: A opção de formatar o pedido do cliente dentro do aplicativo através do catálogo e a checagem dos itens.
- RF06: Toda página que uma das opções do menu direcionar terá um botão de voltar.
- RF07: Na página que se encontra o catálogo haverá um sistema de buscas que utilizara o código ou o nome dos produtos para facilitar a busca dentro do catálogo.
- RF08: O sistema deve permitir o upload pela fábrica do catálogo e suas informações.
- RF11: O sistema deve ser capaz de enviar o pedido para a fábrica.

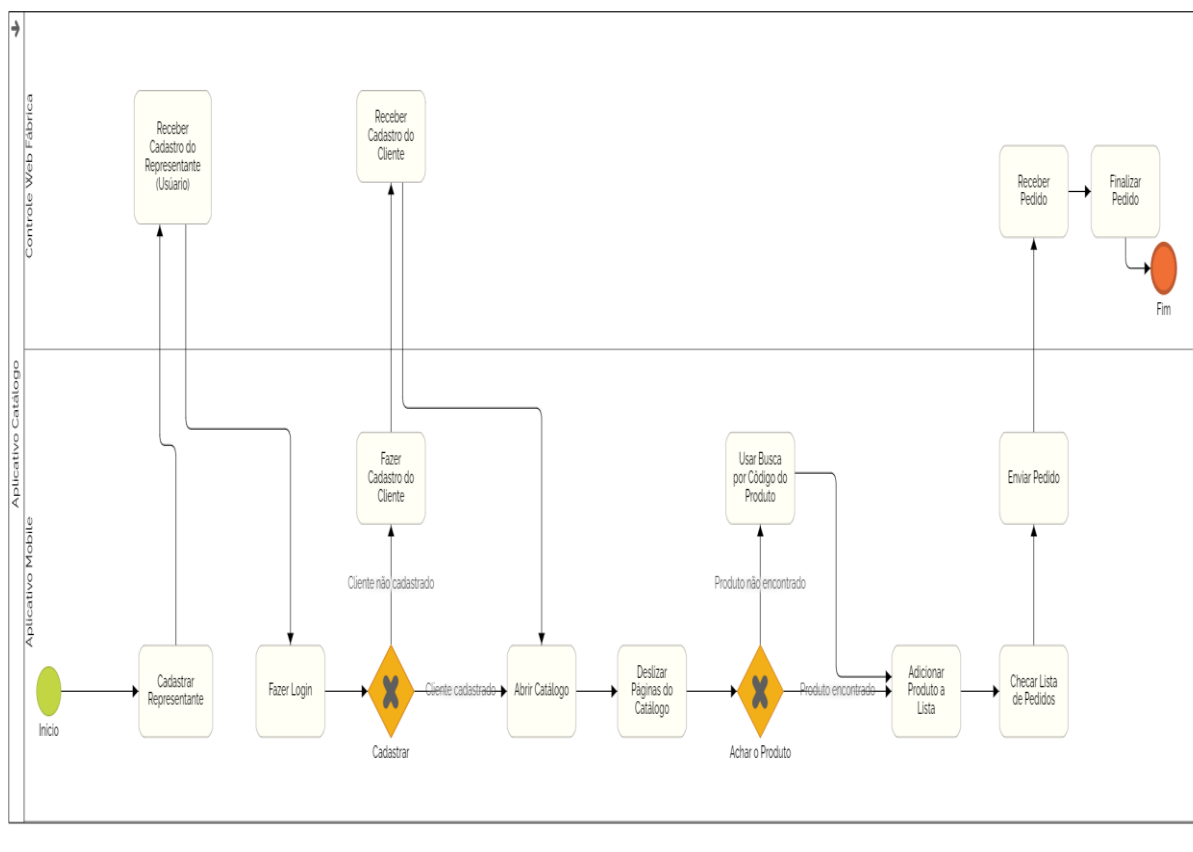
Requisitos Não Funcionais:

- RNF01: Na tela pós-login será incluído o logo da fábrica que o representante trabalha.
- RNF02: Deve possuir um menu na lateral com as opções de funcionalidade.
- RNF03: Toda página que uma das opções do menu direcionar terá um botão de voltar.
- RNF04: O sistema deve apresentar agilidade para que seja possível o aumento na eficiência de processamento de pedidos
- RNF05: O sistema deve estar “on-line” a todo momento para que não haja interrupções nos atendimentos.
- RNF06: O sistema deve ser seguro, pois além de informações de trabalho do representante também irá carregar consigo informações de pedidos e clientes.
- RNF07: O sistema deverá ser desenvolvido em React Native.
- RNF08: O acesso ao sistema não pode demorar mais do que 10 segundos.
- RNF09: Um representante não poderá ter acesso aos dados de outro representante.
- RNF10: A Fábrica terá acesso aos dados do pedido e dos representantes.

3.5 BPMN

O “*Business Process Model and Notation*” é uma série de ícones padronizados para o desenho de processos. É utilizado para um fácil entendimento do usuário e gerenciamento dos processos de negócio.

Figura 4 — BPMN.

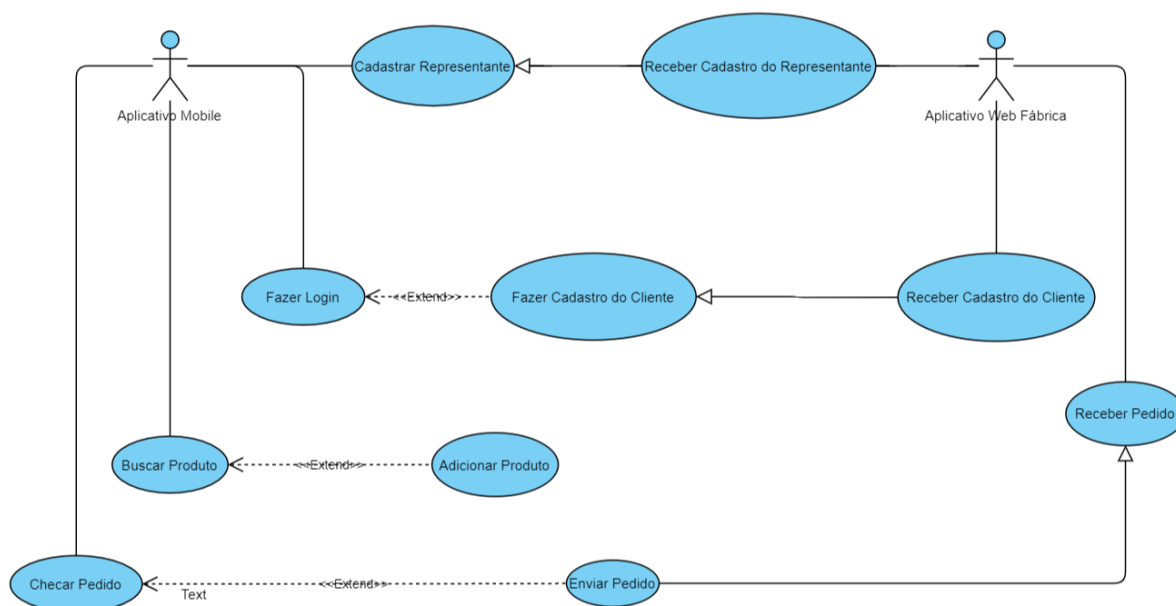


Fonte: O autor.

3.6 Caso de Uso

O diagrama de Caso de Uso do inglês *Use Case* é uma representação gráfica do relacionamento de casos e atores e são amplamente utilizados para a representação das funcionalidades do sistema.

Figura 5 — Caso de Uso.



4 RESULTADOS

O aplicativo proposto proporcionará um menor desgaste físico do representante comercial, pois não mais necessitará carregar os mostruários, que foram substituídos por um catálogo digital. Outro fator essencial é a facilitação com os processos de gerenciamentos de pedidos, controle de clientes e comunicação com os fabricantes. Protótipos das interfaces foram feitos para a ideia geral do produto e a formulação dos testes. Os artefatos criados para a realização da solução, demonstraram o quanto a engenharia de Software é essencial.

5 CONCLUSÃO

De acordo com os resultados alcançados, as questões citadas na introdução foram plenamente satisfeitas e implementadas, proporcionando uma considerável colaboração nos processos do representante comercial. É importante ressaltar o quanto uma pesquisa em referenciais teóricos contribui com a aplicação da etnografia e entrevista para levantamento de requisitos.

Baseando-se nas etapas da engenharia de Software para a criação da aplicação, se observa a criação de uma solução para facilitar o trabalho de um profissional, no caso o representante comercial de roupas. A melhoria contínua do processo no atendimento favorece os vários processos em uma única aplicação.

Finalizando, é importante ressaltar a necessidade da aplicação do processo de ambientação para uso do aplicativo para o uso do representante e o como a solução vai ser aproveitada quando desenvolvida.

REFERÊNCIAS

AGILE MANIFESTO. Manifesto for Agile software development. 2020. **Agile Manifesto**. Disponível em: < <http://agilemanifesto.org/>>. Acesso em: 5 mar. de março de 2020.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 9126-1**: descreve um modelo de qualidade do produto de software. Rio de Janeiro: ABNT, 2003. :2003

GATES, B. A. **Estrada do futuro**. São Paulo: Companhia das Letras, 1995.

LAPLANTE, John. **Processos de software**, 2007.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução Luiz Cláudio Queiroz. 10. ed. São Paulo: Pearson, 2019.

WAZLAWICK, Raul Sidnei. **Engenharia de software: conceitos e práticas**. 2. ed. Rio de Janeiro: Elsevier, 2013.

TURCATO, Augusto. Representante comercial: 6 características deste profissional de vendas. 2019. Disponível em: < <https://crmpiperun.com/blog/representante-comercial/>>. Acesso em: 5 mar. De março de 2020.