

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
ESPECIALIZAÇÃO EM TECNOLOGIA JAVA E
DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS**

GEUCIMAR BRILHADOR

**REEXPLORER: UMA FERRAMENTA DE
SUPORTE À ENGENHARIA DE REQUISITOS**

MONOGRAFIA

CURITIBA

2013

GEUCIMAR BRILHADOR

**REEXPLORER: UMA FERRAMENTA DE
SUPORTE À ENGENHARIA DE REQUISITOS**

Monografia apresentada como requisito parcial à obtenção do título de Especialista em Tecnologia Java e Desenvolvimento para Dispositivos Móveis, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Adolfo Neto

CURITIBA

2013

TERMO DE APROVAÇÃO

REEXPLORER: UMA FERRAMENTA DE SUPORTE À ENGENHARIA DE REQUISITOS

Essa monografia foi apresentada às 11h da manhã do dia 19 de setembro de 2013 como requisito parcial para obtenção do título de Especialista em Tecnologia Java e Desenvolvimento para Dispositivos Móveis, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná. O candidato apresentou o trabalho para a Banca Examinadora composta pelos professores Dra. Maria Claudia F. Pereira Emer, Dr. Laudelino Cordeiro Bastos e Dr. Adolfo Gustavo Serra Seca Neto. Após a deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Adolfo Gustavo Serra Seca Neto
Orientador

RESUMO

BRILHADOR, Geucimar. Reexplorer: uma ferramenta de suporte à engenharia de requisitos. 2013. Monografia (Especialização em Tecnologia Java e Desenvolvimento para Dispositivos Móveis) – Departamento Acadêmico de Informática. Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Documentação é uma parte fundamental no desenvolvimento de *software*. No entanto, boa parte da documentação técnica ainda é desenvolvida em processadores de textos e planilhas. Isto se deve, além de outros fatores, ao fato de que estas ferramentas são populares e de fácil utilização. Mas, por não utilizarem ferramentas especializadas, atividades de documentação de *software* que poderiam ser simplificadas e automatizadas acabam sendo árduas. O armazenamento das informações desta forma atende as demandas da indústria, mas dificulta a manutenção e processamento posterior das informações. Este trabalho apresenta uma pesquisa sobre documentação de *software* e propõe uma ferramenta para facilitar as atividades de engenharia de requisitos e simplificar o processo de geração de documentos. Como resultado, foi desenvolvida a ferramenta *Reexplorer Community* para suportar as atividades de levantamento e gerenciamento de requisitos de acordo com a metodologia clássica de desenvolvimento, permitindo após a inclusão dos dados referentes ao projeto, a extração destas informações como documentos de visão, requisitos, casos de uso, casos de teste, impacto, rastreabilidade, regras de negócio, etc..

Palavras-chave

Requisitos, Geração Automática de Documentos, Engenharia de Requisitos.

ABSTRACT

BRILHADOR, Geucimar. Reexplorer: a tool to support requirements engineering. 2013. Monografia (Especialização em Tecnologia Java e Desenvolvimento para Dispositivos Móveis) – Departamento Acadêmico de Informática. Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Documentation is an essential part in software development. However, much of the technical documentation is still developed in word processors and spreadsheets. This is due, among other factors, to the fact that these tools are popular and easy to use. But, by not using specialized tools, software documentation activities that could be simplified and automated end up being tough. The storage of information in this way meets the demands of the industry, but is difficult to maintain and further processing of the information. This monograph presents a survey of software documentation and proposes a tool to facilitate the activities of requirements engineering and simplify the process of documents generation. As a result, the Reexplorer Community tool was developed to support the activities of requirements elicitation and management according to the classical development methodology, after allowing the inclusion of data relating to the project, the extraction of such information as vision documents, requirements, use cases, test cases, impact, traceability, business rules, etc..

Keywords

Requirements, Automatic Generation of Documents, Requirements Engineering.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - PIRÂMIDE DE REQUISITOS	11
FIGURA 2 - ESCOPO DO DOCUMENTO DE VISÃO.....	12

SUMÁRIO

1 INTRODUÇÃO	1
1.1 FORMULAÇÃO DO PROBLEMA E JUSTIFICATIVA.....	2
1.2 OBJETIVO GERAL.....	4
1.3 OBJETIVOS ESPECÍFICOS.....	4
2 REVISÃO BIBLIOGRÁFICA	5
2.1 A CONTROVÉRSIA.....	6
2.1.1 Processo Unificado: baseado em documentos.....	6
2.1.1.1 Iterações baseadas em requisitos.....	7
2.1.2 Modelo Ágil: documentação sob-demanda.....	8
2.1.2.1 Iterações baseadas em histórias do usuário.....	9
2.2 FUNDAMENTOS DA ENGENHARIA DE REQUISITOS.....	10
2.2.1 Para que servem os requisitos?.....	10
2.2.2 Pirâmide de Requisitos.....	10
2.2.3 Documento de visão: o passo inicial.....	11
2.2.4 Requisitos.....	12
2.2.4.1 Definição.....	12
2.2.5 Rastreabilidade e análise de impacto.....	13
2.3 CONSIDERAÇÕES.....	14
3 REEXPLORER: FERRAMENTA PARA ENGENHARIA DE REQUISITOS	16
3.1 MATERIAIS E MÉTODOS.....	17
4 DESENVOLVIMENTO	18
4.1 PROTÓTIPOS DE TELAS.....	18
4.1.1 Tela de login.....	18
4.1.2 Home.....	19
4.1.3 Menu principal.....	19
4.1.4 Tela de pesquisa.....	20
4.1.5 Tela de requisitos.....	20
4.1.6 Tela de reuniões.....	21
4.1.7 Tela de exportação de documentos.....	21
4.2 ARQUITETURA.....	22
4.3 REQUISITOS DO <i>SOFTWARE</i>	22
4.4 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.....	24

4.4.1	Requisitos Funcionais	25
4.4.2	Requisitos Não Funcionais	31
4.5	RELAÇÃO ENTRE REQUISITOS FUNCIONAIS E CASOS DE USO	35
4.6	RASTREAMENTO DE REQUISITOS	36
4.6.1	Matriz de rastreabilidade para trás de Requisitos Funcionais	37
4.6.2	Matriz de rastreabilidade para trás de Requisitos Não Funcionais	38
4.6.3	Matriz de rastreabilidade para frente de Requisitos Funcionais	39
4.6.4	Matriz de rastreabilidade para frente de Requisitos Não Funcionais.....	40
4.7	RASTREAMENTO DE DEPENDÊNCIAS	41
4.7.1	Matriz de custos	41
5	RESULTADOS	42
6	CONSIDERAÇÕES FINAIS.....	43
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	44
8	APÊNDICE.....	46
8.1	DIAGRAMA DE CASOS DE USO.....	46
8.2	ESPECIFICAÇÕES DE CASOS DE USO	46
8.3	DIAGRAMA DE CLASSES	81

1 INTRODUÇÃO

O desenvolvimento de *software*, de médio ou grande porte, visto de uma forma clássica, inicia-se com a definição da visão dos interessados, passando pelas fases de análise financeira, levantamento de requisitos, especificação, modelagem, desenvolvimento, teste e entrega do *software*. Em cada fase (IBM, 2007), vários documentos são produzidos para atender as diferentes necessidades das pessoas envolvidas no projeto. Alguns documentos são criados exclusivamente para atender os aspectos gerenciais do projeto, entre os quais podemos citar os aspectos financeiros e prazos. Além destes, vários documentos técnicos são desenvolvidos para guiar a arquitetura e as regras de domínio que o *software* se propõe a solucionar. Entre os documentos podemos citar documentos de visão, requisitos, especificações de casos de uso etc.

Na produção de documentos, as empresas de desenvolvimento de *software* fazem uso de ferramentas de escritório, como processadores de textos, planilhas e etc., para desenvolver parte dos documentos do *software*, criando tanto documentos gerenciais como documentos técnicos. Isto se deve, além de outros fatores, ao fato de que estas ferramentas são populares e de fácil utilização. Contudo, por não utilizarem ferramentas especializadas atividades de documentação de *software* que poderiam ser simplificadas e automatizadas acabam sendo árduas. Por exemplo, é muito comum na indústria de *software* a utilização de *templates* para a criação da documentação necessária ao processo de desenvolvimento de *software*. Esta abordagem padroniza a criação de documentos, mas entedia a atividade de documentação, pois a atividade se restringe a preencher *templates* conforme o padrão adotado.

Segundo Godowski (2005), uma pesquisa realizada na Universidade de Trento mostra que:

- 71,8% dos documentos são escritos em linguagem natural;
- 15,9% são escritos em linguagem estruturada;
- Somente 5% são escritos em linguagem formal.

Obviamente, o uso da linguagem natural para documentar *software* induz ao uso de processadores de texto para as atividades de documentação. No entanto, somente pesquisas sobre novas formas de documentar permitirão o uso de outras linguagens e ferramentas.

Sommerville escreve:

“A linguagem natural tem sido usada para escrever requisitos de *software* desde o início da engenharia de *software*. Ela é expressiva, intuitiva e universal. Ela é também potencialmente vaga, ambígua e isto significa que ela depende do conhecimento do leitor. Como resultado, existem várias formas alternativas propostas para escrever requisitos. Todavia, nenhuma destas tem sido largamente adotada e a linguagem natural continuará a ser a linguagem mais usada para especificar sistemas e requisitos de *software*.” (SOMMERVILLE, 2011, p. 97, tradução própria)

Na busca de novas alternativas para documentar *software*, este projeto propõe desenvolver uma ferramenta especializada no processo de documentação de *software*, mais especificamente na área de engenharia de requisitos, que ajude a identificar meios para documentar *software*. Espera-se que os documentos sejam resultado de um conjunto de informações consistentes e atualizadas, permitindo que a qualquer momento seja possível extrair, de forma dinâmica, os documentos para comunicação com os interessados, bem como os artefatos necessários para as atividades de desenvolvimento de *software*.

A versão inicial da ferramenta, denominada de *Reexplorer Community*, será desenvolvida para *tablets* com sistema operacional Android. Posteriormente, novas versões da ferramenta podem ser desenvolvidas para outros sistemas e dispositivos.

1.1 FORMULAÇÃO DO PROBLEMA E JUSTIFICATIVA

De acordo com Leffingwell (2003, p. 9), um veterano da indústria de *software*, criador da ferramenta *RequisitePro*, vice-presidente da *Rational Software* — empresa do grupo IBM — responsável pela divulgação do RUP e UML. Em um estudo realizado por Capers Jones em 1994 identificou que aproximadamente 30% dos defeitos encontrados nos *softwares* se devem aos erros cometidos durante o processo de documentação de *software*. Para Leffingwell é essencial ter requisitos bem especificados. Segundo ele, o documento de requisitos define os recursos que o sistema precisa ter e pode determinar o sucesso de um projeto. Ele afirma que:

“Uma vez que a maioria de nós não consegue memorizar mais do que poucas dezenas de informações em nossa memória, **documentar** os requisitos é necessário para manter uma comunicação efetiva entre vários *stakeholders*. Os requisitos precisam ser gravados num meio acessível: um

documento, um modelo, um banco de dados, ou numa lista em um quadro branco.” (LEFFINGWELL; WIDRIG, 2003, p. 16, tradução própria)

Por outro lado, Jack W. Reeves, outro veterano da indústria de *software* com mais de 30 anos de experiência — desenvolvedor de *softwares* para simuladores de ônibus espacial, controle de tráfego aéreo, geração de imagens médicas, distribuição de dados financeiros, *softwares* embarcados, *drivers* e utilitários —, escreveu no artigo “*What Is Software Design?*” publicado no *C++ Journal* de 1992:

“Eu li na revista *Datamation* há quase 10 anos atrás, um artigo que discutia porque os engenheiros precisavam ser bons escritores (eu acho que era este o objetivo principal do artigo — sobre o qual ele continuou falando), mas o ponto chave que eu identifiquei no artigo é que o resultado final do trabalho de um processo de engenharia é um documento. Em outras palavras, engenheiros produzem documentos, não coisas. Outras pessoas pegam estes documentos e produzem alguma coisa. Assim, eu me fiz a seguinte pergunta: ‘De toda a documentação que os projetos de *software* normalmente geram, há alguma coisa que poderia realmente ser considerado um documento de engenharia?’. A resposta que me veio foi, ‘Sim, há um documento, e apenas um — o código-fonte’”. (MARTIN, 2002, p. 517, tradução própria)

Estas duas visões antagônicas sobre documentação são resultado de uma discussão que atravessa décadas e uma série de perguntas ainda sem respostas.

É necessário documentar *softwares*? Como documentar um *software* com mais de um milhão de linhas de código? Quantos documentos são necessários para representar toda a solução e sua arquitetura? Como manter as informações em poucos documentos, sem excessos, e assegurar que todos eles estão atualizados? Quanto tempo e dinheiro é gasto para documentar estas informações e como garantir que os futuros leitores tenham acesso rápido a uma informação confiável?

Ao contrário de Leffingwell que acredita ser importante documentar *software*, as ideias de que a documentação tem impactos negativos, dificultam e aumentam os custos do projeto, como as defendidas por Reeves, vão influenciar, anos mais tarde, vários desenvolvedores de *software*.

Para encontrar respostas às perguntas acima é preciso pensar em novas formas de documentar um *software*, principalmente em relação aos aspectos técnicos. A indústria de *software* não pode apenas utilizar ferramentas para produzir

documentação. Ao contrário, precisa ter ferramentas que facilitem o armazenamento das informações e que permitam a geração da documentação necessária para a equipe técnica do projeto. Isto, de forma que pouco tempo seja gasto para manter a documentação atualizada e que a equipe fique concentrada nos aspectos essenciais de desenvolvimento do *software*.

1.2 OBJETIVO GERAL

Desenvolver uma ferramenta de suporte à engenharia de requisitos que permita armazenar as informações referentes ao *software* em desenvolvimento e produzir de forma dinâmica documentos atualizados de *software*.

1.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho incluem:

- O desenvolvimento de uma ferramenta para automatizar o processo de levantamento e gerenciamento de requisitos.
- A geração automática de documentos de visão, requisitos, testes, impactos e mudanças baseados em *templates*.
- Análise de vinculação de artefatos de *software* de alto nível de abstração com artefatos de mais baixo nível para análise e rastreamento de impacto de mudanças.

2 REVISÃO BIBLIOGRÁFICA

Desde os anos 70, em que Dijkstra (1972) cunhou o termo “crise de *software*” numa conferência realizada pela *Association for Computing Machinery* (ACM), o processo de desenvolvimento de *software* evoluiu significativamente. Nesta conferência, Dijkstra manifestava as dificuldades enfrentadas pela engenharia de *software* em relação aos orçamentos e prazos estourados, *software* de baixa qualidade, requisitos não atendidos e baixa manutenibilidade. Havia um consenso na época de que as possíveis soluções para o problema seria a evolução nas metodologias, técnicas e ferramentas, bem como o aumento de investimento na educação, visando mostrar o “que é” e “como fazer” o desenvolvimento de *software*.

Nesta época, a engenharia de *software* trabalhava de forma sequencial, cada fase do processo de desenvolvimento era realizada completamente para só então iniciar os trabalhos da próxima fase. Esta abordagem ficou conhecida como modelo em cascata.

No final da década de 80, Boehm (1988) apresentou um modelo de engenharia que ficou conhecido como espiral. Neste modelo o desenvolvimento era realizado de forma iterativa e incremental. As atividades de análise e engenharia eram aplicadas em cada iteração do projeto, permitindo assim maior foco no objetivo do projeto.

Anos mais tarde, profissionais como Ivar Jacobson, Grady Booch e James Rumbaugh, desenvolvedores de grandes projetos de *software*, se uniram para criar um processo utilizando o modelo iterativo (JACOBSON, et. al, 1999), com grande influência da orientação a objetos e da linguagem unificada de modelagem (*Unified Modeling Language, UML*). O modelo ficou conhecido como o Processo Unificado.

Apesar de já se passarem mais de 40 anos desde o evento promovido pela ACM e, após a criação do RUP, o processo de desenvolvimento de *software* ter evoluído significativamente, em 2001 vários desenvolvedores, entre eles Kent Beck, Robert C. Martin, Ward Cunningham, Martin Fowler e outros, se uniram para criar o Manifesto para Desenvolvimento Ágil de *Software* (BECK, et al, 2001). De acordo com o manifesto, eles acreditavam que o processo de desenvolvimento deve ser informal, voltado às necessidades do cliente, com menor interferência dos processos de engenharia e com maior flexibilidade para mudanças.

2.1 A CONTROVÉRSIA

A discussão sobre a aplicação de um modelo formal como o proposto pelo RUP, ou informal como o apresentado pelos autores do modelo ágil, sobre a melhor abordagem para desenvolvimento de *software*, se com um processo rígido de engenharia ou não, teve como origem as necessidades de mudança que surgem durante as fases de desenvolvimento. A pilha de documentação produzida pela engenharia clássica de *software*, diagramas etc. e suas interdependências, além de aumentar o trabalho, burocratiza significativamente as solicitações de mudança apresentadas pelos interessados. Esta abordagem também representa um aumento significativo no custo de desenvolvimento. De acordo com Pressman (2010, p. 68), é consideravelmente baixo o custo de mudanças nas fases iniciais do projeto. Porém, à medida que o projeto avança, aumenta o custo para alteração da documentação e do código existente.

Responder rapidamente às solicitações de mudança é um dos pilares do Manifesto para Desenvolvimento Ágil de *Software*. Os *softwares* buscam solucionar problemas da sociedade, e, como ela está em constante mudança (SOMMERVILLE, 2011), é inevitável que tais mudanças reflitam no desenvolvimento. Por isso, é importante estar preparado para atender as mudanças a qualquer momento, mesmo em fases avançadas do projeto.

2.1.1 Processo Unificado: baseado em documentos.

Embora o RUP tenha se consolidado apenas a partir da segunda metade da década de 90, tendo, portanto, pouco mais de 15 anos desde sua formalização, ele incorporou várias práticas e técnicas que já eram utilizadas nos modelos tradicionais como cascata e espiral. O grande diferencial do RUP, em relação aos modelos tradicionais, foi a utilização do modelo iterativo e unificação de várias técnicas e ferramentas em um único processo (IBM, 2007).

Nesta última década, muitas empresas adotaram o RUP ou processos similares para desenvolvimento de *software*. Isto trouxe para a indústria de *software* um padrão que enriqueceu muito o processo de desenvolvimento e que sistematizou bastante as atividades de documentação.

2.1.1.1 Iterações baseadas em requisitos.

Em 1997 foram incluídos os documentos de requisitos e testes como parte do processo de documentação de *software* adotado pelo RUP. A partir de então, o documento de requisitos tornou-se a base para os demais documentos e artefatos de *software*. Assim, os documentos de um sistema, como documento de visão, requisitos, diagramas de classe, sequência e outros, definidos na UML, passaram a fazer parte dos itens que deveriam ser desenvolvidos juntamente com o *software*. Sommerville, ao falar sobre esta forma de desenvolvimento, escreve:

“... quando estamos falando sobre engenharia de *software*, o *software* não é apenas o programa em si, mas também toda a **documentação** e dados de configuração que é necessário para que o programa funcione corretamente. O sistema normalmente consiste de um número de programas e arquivos de configuração que são usados para configurar estes programas. Isto pode incluir **documentação do sistema**, que descreve a estrutura do sistema; **documentação do usuário**, que explica como utilizar o sistema, e websites para o usuário baixar informações atualizadas do produto. Esta é uma das mais importantes diferenças entre o desenvolvimento profissional de *software* e o amador. Se você está escrevendo um programa para você mesmo, ninguém além de você irá utilizá-lo, você não precisa ficar perturbado em escrever guia de usuário, documentar o design do sistema, etc. Porém, se você está escrevendo um *software* que outras pessoas utilizarão e outros engenheiros irão alterá-lo então você tem que fornecer informação adicional assim como o código do *software*.” (SOMMERVILLE, 2011, p. 5, tradução e grifo próprio)

De acordo com ele a,

“Especificação de *software* ou engenharia de requisitos é o processo de compreender e definir quais serviços são necessários no sistema e identificar as regras na operação e no desenvolvimento do sistema. A engenharia de requisitos é um estágio particularmente crítico do processo de *software* sendo que um erro neste estágio inevitavelmente trará problemas no design e na implementação do sistema.

O processo da engenharia de requisitos tem como objetivo produzir um **documento de requisitos** que especifique os requisitos do sistema de forma que satisfaça e atenda todas as necessidades dos interessados. Os requisitos normalmente são apresentados em dois níveis de detalhes. Requisitos de alto nível para usuários finais e clientes; desenvolvedores do

sistema precisam de uma especificação mais detalhada, de mais baixo nível.” (SOMMERVILLE, 2011, p. 36-37, tradução e grifo próprio)

Os documentos citados por Sommerville eram e são utilizados na maioria dos projetos que adotam o RUP como processo de desenvolvimento. Mudar as informações nos documentos de mais alto nível, como o documento de requisitos, significa que todos os documentos dependentes, códigos fontes e mesmo a arquitetura do sistema pode precisar de alteração para atender as solicitações de mudanças. Além disso, no modelo iterativo os requisitos são definidos de forma incremental, o que significa que um requisito pode sofrer mudanças até a conclusão do projeto. Assim, nenhum requisito é/está totalmente fechado no início do projeto. Para Leffingwell (2003), a transição para o modelo iterativo mostrou que a ideia de que os requisitos poderiam ser completamente especificados antes do desenvolvimento do *software* era infundada e deveria ser abandonada.

Embora, o modelo iterativo adotado pelo RUP consiga se adaptar mais facilmente às solicitações de mudança do que os modelos tradicionais, ele ainda é muito criticado pelos desenvolvedores que defendem um processo informal.

2.1.2 Modelo Ágil: documentação sob-demanda.

No final da década de 90, quando o RUP ainda dava seus primeiros passos, muitos desenvolvedores já manifestavam seu descontentamento com os modelos existentes e com o excesso de processos, ferramentas, documentações, planejamentos e formalismos adotados por estes modelos. Martin (2002), um dos defensores do manifesto ágil, questionava se ao desenvolver um *software* deveríamos gastar tempo escrevendo pilhas de documentação ou ir diretamente para a codificação da solução e produzir códigos autoexplicativos.

Ao propor mudanças nos modelos existentes, os autores do manifesto adaptaram as técnicas de desenvolvimento em pequenas iterações com foco nas necessidades dos clientes, programação em par, refatoração e desenvolvimento dirigido por testes. Segundo Leffingwell (2011), o movimento ágil se firmou como um processo baseado em equipes que, no pior dos casos, tende a excluir as práticas de gestão de processos. Obviamente, embora não se tenha informação suficiente para avaliar a eficácia desta abordagem em projetos de médio e grande porte, sabe-se que muitos desenvolvedores têm utilizado esta metodologia com sucesso em

pequenos *softwares* e várias empresas estão utilizando uma abordagem mista em projetos de maior porte.

2.1.2.1 Iterações baseadas em histórias do usuário.

Diferente das metodologias clássicas, que partem da fase de levantamento de requisitos para as fases de especificação e modelagem para só então iniciar a fase de desenvolvimento do *software*, uma equipe que adota uma metodologia ágil de desenvolvimento parte de pequenas anotações conhecidas como “histórias do usuário” diretamente para o desenvolvimento do código. Cada iteração incrementa uma nova funcionalidade completamente testada e aprovada pelo cliente.

“Histórias dos usuários são pequenas descrições das funcionalidades fornecidas pelo cliente que ajudam a entender o que o *software* deve fazer. **Diferente de especificação de *software* em metodologias clássicas**, a história do usuário visa fornecer apenas **informações essenciais, indispensáveis** para o entendimento dos envolvidos no processo de desenvolvimento, com intuito de gerar uma visão compartilhada de negócio.” (SOARES; FERREIRA, 2010, p. 68)

Apesar de haver certa semelhança entre as histórias do usuário com as especificações de requisitos feitas na abordagem clássica é importante observar que existem diferenças essenciais entre elas. Ambler (2012), afirma que a história do usuário é apenas um lembrete para o desenvolvedor falar com o cliente antes de iniciar o desenvolvimento. Soares e Ferreira (2010), afirmam que as histórias do usuário contêm apenas as informações necessárias para o desenvolvimento e são diferentes dos casos de uso. A diferença essencial entre a história do usuário e a especificação de caso de uso é que a história do usuário é uma narrativa feita do ponto de vista do usuário, enquanto que o caso de uso é uma narrativa impessoal. Outra diferença importante nesta abordagem, é que o *product owner* (PO), uma pessoa que conhece do negócio, capaz de sanar as dúvidas dos desenvolvedores, convive diariamente com a equipe passando as informações, que na engenharia clássica iria para um documento, diretamente ao programador que vai desenvolver a funcionalidade, substituindo o conceito de documentação. Além disso, o PO prioriza as atividades do projeto, indicando quais funcionalidades devem ser desenvolvidas.

2.2 FUNDAMENTOS DA ENGENHARIA DE REQUISITOS

Segundo Leffingwell (2011), os erros na especificação de requisitos são apontados como as principais causas de falhas nos projetos. Para ele, garantir a participação do usuário no processo de desenvolvimento, cuidar com requisitos e especificações incompletas, mudanças e atender para as reais necessidades dos usuários são os grandes desafios da engenharia de requisitos.

2.2.1 Para que servem os requisitos?

A engenharia de requisitos, conforme aplicada nas metodologias clássicas de desenvolvimento de *software*, tem por fim desenvolver ferramentas, técnicas, modelos e documentos de entrada para as fases de modelagem e desenvolvimento de *software*. O documento de requisitos, produto final das atividades de engenharia de requisitos, pode ser utilizado para vários propósitos (SOMMERVILLE, 2011, p. 92), entre eles:

- Definição de escopo e planejamento de projeto;
- Desenho e teste do *software*;
- Documentação e material de treinamento.

Porém, o gerenciamento de requisitos não é muito comum em projetos pequenos, sendo uma necessidade evidente em projetos de médio ou grande porte, com centenas ou milhares de requisitos.

Um gerenciamento eficiente de requisitos deve estar atento às mudanças que ocorrem nos requisitos devidos aos erros no levantamento inicial, mudança nas regras do negócio e a novos requisitos que surgem com o avanço no projeto do *software*. Deve possuir um modelo de rastreamento que permita aos interessados no projeto, avaliar o impacto e risco de determinadas mudanças nos requisitos.

2.2.2 Pirâmide de Requisitos

A motivação para a criação de um *software*, normalmente, está relacionada às necessidades que algum interessado ou um grupo de interessados têm de determinado *software*. Para Leffingwell (2003), as necessidades são originárias do **problema** que o interessado enfrenta em determinado domínio. Estas necessidades por sua vez apontam para as funcionalidades e que o interessado julga necessário para atender suas expectativas.

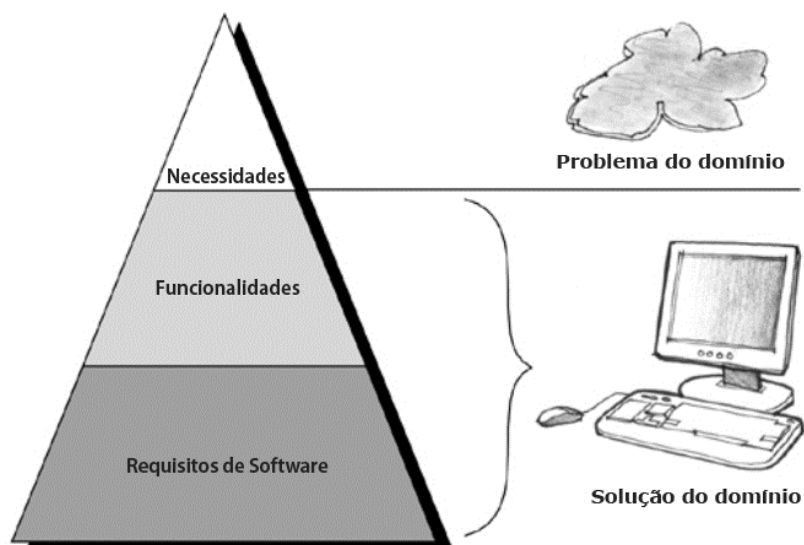


Figura 1 - Pirâmide de requisitos. (LEFFINGWELL; WIDRIG, 2003, p. 21, tradução própria)

Segundo ele, na pirâmide há três níveis de informações. No topo dela estão as necessidades do usuário, algo entre cinco e cinquenta itens de alto nível. Logo abaixo estão as funcionalidades, possíveis soluções preferencialmente narradas pelos interessados. As funcionalidades são vistas com um conjunto de requisitos em alto nível que ajudam o interessado a realizar seu objetivo. Na base da pirâmide os requisitos que servem como entrada para a equipe técnica de projeto envolvida no desenvolvimento do *software*.

2.2.3 Documento de visão: o passo inicial

Entender o que o interessado deseja e definir o escopo que o projeto terá é o primeiro passo para a construção de um *software*. Partindo de um texto escrito em linguagem natural, em alto nível de abstração, o documento de visão registra as necessidades e as funcionalidades que o interessado deseja. As duas divisões no topo da pirâmide apresentam o escopo do documento de visão (LEFFINGWELL; WIDRIG, 2003, p. 174).

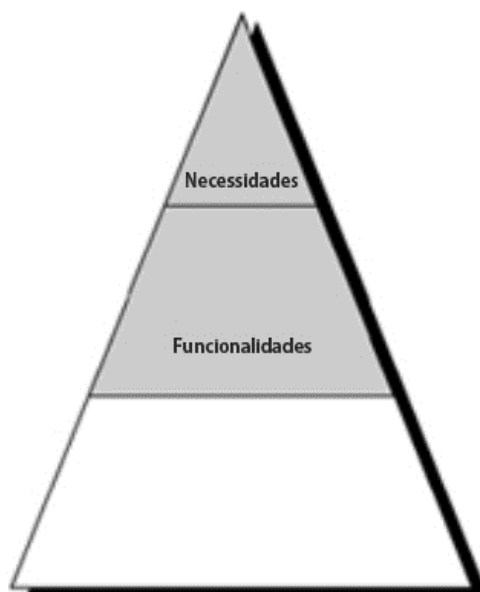


Figura 2 - Escopo do documento de visão. (LEFFINGWELL; WIDRIG, 2003, p. 174, tradução própria)

O documento de visão não possui informações técnicas detalhadas e naturalmente não é o documento utilizado para a codificação do *software*. No entanto, ele responde questões como: Qual é o problema que o *software* soluciona? Quais funcionalidades ele fornece? É um documento que serve para comunicar num nível muito elevado de abstração o objetivo do *software* e serve também como base para elaboração de contratos (SOMMERVILLE, 2011, p. 91) entre cliente e desenvolvedores de *softwares*.

2.2.4 Requisitos

2.2.4.1 Definição

Entre os especialistas, é comum o entendimento de que um requisito é uma condição que o *software* em desenvolvimento deve ter para atender a necessidade do cliente. O requisito tende a ser mais objetivo que a descrição da necessidade e pode ser escrito com a ideia de implementação na mente. Para Zielczynski (2006), um requisito é uma condição que o sistema deve atender.

Sommerville (2011, p. 83) entende que existe uma separação no nível de abstração entre os requisitos. Para ele, as necessidades do usuário são escritas em linguagem natural, em alto nível, tornando possível ao usuário definir regras e o que se espera com sua utilização. Já os requisitos são descrições mais detalhadas que precisam definir exatamente o que precisa ser desenvolvido.

As necessidades são direcionadas para usuários, compradores, gerentes de contrato etc. Os requisitos servem para engenheiros, arquitetos e desenvolvedores.

Para Bogazzi (2008), independente do processo de desenvolvimento adotado, capturar corretamente os requisitos e mantê-los atualizados por todas as fases do projeto é uma tarefa muito importante para garantir a satisfação do cliente. De acordo com a *Hewlett-Packard* (2012), uma boa prática na definição de requisitos é iniciar a especificação em alto-nível e, com submissões sucessivas aos desenvolvedores, ir aprofundando nos detalhes técnicos até que a informação seja suficiente para os desenvolvedores iniciarem a codificação.

É importante, logo após a conclusão da especificação do documento de requisitos, fazer a validação. Como os requisitos estão sujeitos à inconsistência, ambiguidade e incompletude é importante que eles passem por revisões rigorosas. Problemas identificados devem ser corrigidos preferencialmente na fase inicial do projeto, pois, quanto antes os problemas forem identificados menores são os custos para suas correções (PRESSMAN, 2010, p.67).

2.2.5 Rastreabilidade e análise de impacto

Através da engenharia de requisitos também é possível estabelecer as diretrizes que conduzirão todo o processo de desenvolvimento e que servirão de entradas para as demais atividades relacionadas ao projeto. Contudo, mesmo durante a fase de desenvolvimento do *software* é comum que os requisitos passem por alterações. De acordo com Sommerville (2011), as mudanças no projeto de *software* são inevitáveis, logo, o processo de desenvolvimento adotado deve estar preparado para atender as mudanças.

Documentar e manter as documentações atualizadas são atividades que devem ser realizadas durante todo o ciclo de desenvolvimento de um *software*. Em face da dificuldade de manutenção e do risco na inclusão de novas funcionalidades, *softwares* desenvolvidos sem documentação tendem a serem descontinuados muito cedo. Ter ferramentas e documentos que ajudem na rastreabilidade das mudanças nos requisitos e que possibilitem a análise de impacto das mudanças é vital para assegurar que o *software* seja usado por vários anos. Para facilitar a análise, é importante criar matrizes de rastreabilidade, que mapeiam os requisitos funcionais e não funcionais para trás identificando o solicitante da funcionalidade e para frente identificando os artefatos que foram criados com base nos requisitos.

2.3 CONSIDERAÇÕES

Na 20th IEEE International Requirements Engineering Conference realizada em agosto e setembro de 2011 em Trento na Itália, Rupp (2011) discursando sobre o tema **Está a indústria tentando se livrar dos requisitos?**, diz: “*Se alguém olha a abordagem das metodologias ágeis, assim como Scrum, e do mínimo de Engenharia de Requisitos (ER) contida nestes modelos de processo, pode ter a impressão de que a ER em breve deixará de existir.*” A diferença fundamental entre o modelo clássico e o modelo ágil de desenvolvimento de *software*, afirma Sommerville (2011, p. 67), é que no modelo clássico sabe-se que o *software* sofrerá mudanças, por isso, é gasto um tempo adicional tentando prever futuras mudanças. No desenvolvimento ágil, sabe-se que as mudanças virão, mas nenhuma mudança é motivo de preocupação até que ela venha. Segundo ele, algumas empresas têm adotado metodologias mistas, incluindo programação em par, etc., mas utilizando o documento de requisitos do modelo tradicional.

O modelo unificado de desenvolvimento de *software* é bastante consolidado. Todavia, é importante buscar novas abordagens que possam evoluir os processos de documentação de *software*, bem como identificar meios de simplificar a geração de documentos de forma que qualquer erro de especificação possa ser rapidamente aplicado. De acordo com Leffingwell (2011), a explosão de metodologias ágeis no final da década de 90 se deve, além de outros fatores, à ideia de que escrever código e submetê-lo de imediato para avaliação do cliente pode ser mais econômico do que tentar antecipar todas as informações no início do projeto.

A discussão sobre a melhor forma de se desenvolver *software*, as práticas e metodologias que devem ser aplicadas, não está encerrada. Naturalmente, a abordagem de desenvolvimento a ser utilizada depende do tipo, tamanho do *software* que está sendo desenvolvido, equipe e outros fatores. O uso de determinado processo em diferentes contextos trará melhores ou piores resultados.

Como os documentos são escritos em linguagem natural para atender diversas pessoas que não estão envolvidas diretamente no desenvolvimento do *software*, eles não possuem informações suficientes para os níveis mais técnicos de projeto que demandam por diagramas, notações, fórmulas e abstrações especiais de acordo com a complexidade do *software*. Manter os documentos, nos mais variados níveis de abstração, atualizados não é uma tarefa fácil. As discussões

mostram que ter documentação suficiente, atualizada e de fácil manutenção, ainda é um dos grandes desafios do desenvolvimento de *software*.

Assim, considerando que o modelo ágil não prioriza a documentação de *software*, este trabalho se baseia na metodologia clássica de desenvolvimento e nos fundamentos de engenharia de requisitos apresentados por Leffingwell e outros autores; buscando formas dinâmicas de gerar documentação de *software*.

3 REEXPLORER: FERRAMENTA PARA ENGENHARIA DE REQUISITOS

Como pode se observar, um dos grandes desafios no desenvolvimento de *software* é compreender as necessidades dos interessados, documentar e manter as informações relativas ao *software* atualizadas.

A ferramenta *Reexplorer Community*, proposta neste trabalho, foi idealizada e desenvolvida com base nos princípios de documentação utilizados no RUP, mais especificamente nas atividades de engenharia clássica de requisitos, para ajudar na fase de levantamento, especificação e gerenciamento de requisitos. Com ela, os analistas podem coletar dados referentes às necessidades dos interessados, funcionalidades esperadas, requisitos funcionais e não funcionais do *software*, elaborar questionários para entrevistas, agendar eventos, definir *templates* de casos de uso, elicitar transações e classes, validar requisitos e incluir solicitação de mudança etc.

A versão foi desenvolvida para *tablets*, com sistema operacional Android, e tem como foco principal analistas de sistema e negócio. A opção pela criação da ferramenta para este dispositivo se deve ao fato de que a atividade de levantamento de requisitos é uma atividade intensa. Exige atenção, concentração e habilidade do analista para interagir com os *stakeholders* e ao mesmo tempo registrar as informações essenciais para o desenvolvimento do projeto. Assim, é importante ter em mãos uma ferramenta que possa ajudar na atividade. Além disso, os *tablets* oferecem recursos de captura de áudio, vídeo, imagem que podem ampliar os recursos da ferramenta.

Existem no mercado ferramentas comerciais, entre elas *RequisitePro* da *Rational*, *CaliberRM* da *Borland*, etc., de código fechado que impossibilitam sua utilização em pesquisas. Ter uma ferramenta que permite a manipulação do código fonte, possibilita não apenas a automatização do processo de gerenciamento de requisitos, mas também a avaliação de diferentes técnicas na geração de documentos e a vinculação — de forma não intrusiva — de requisitos aos demais artefatos de *software*. Assim, é possível estudar mecanismos capazes de manter a consistência de documentos de *software* em diferentes níveis de abstração além da utilização de linguagem formal, UML e outras linguagens para evoluir o processo de documentação. A ferramenta valoriza as técnicas e linguagens desenvolvidas pelos processos clássicos, mas busca promover a geração automática de documentos e

parte do princípio de que a documentação seja gerada de forma dinâmica de acordo com as necessidades dos interessados.

3.1 MATERIAIS E MÉTODOS

O trabalho foi iniciado com uma pesquisa e coleta de dados referente à documentação de *software*. Através desta atividade levantaram-se as referências bibliográficas citadas no final deste documento. Além disso, as informações absorvidas na disciplina de Engenharia de Requisitos do Mestrado em Computação Aplicada da UTFPR ajudaram nas definições dos requisitos da ferramenta.

Para produção da documentação foi utilizado o processador de texto *Word* v. 2010. Para prototipagem de telas foi utilizado o *software Pencil* v. 2.0. No desenvolvimento foram utilizados os *softwares Enterprise Architect* v. 7.5, para modelagem da ferramenta utilizando a UML, e *Eclipse Indigo* v. 3.7 com Android SDK v. 22.1, para codificação da aplicação na linguagem Java.

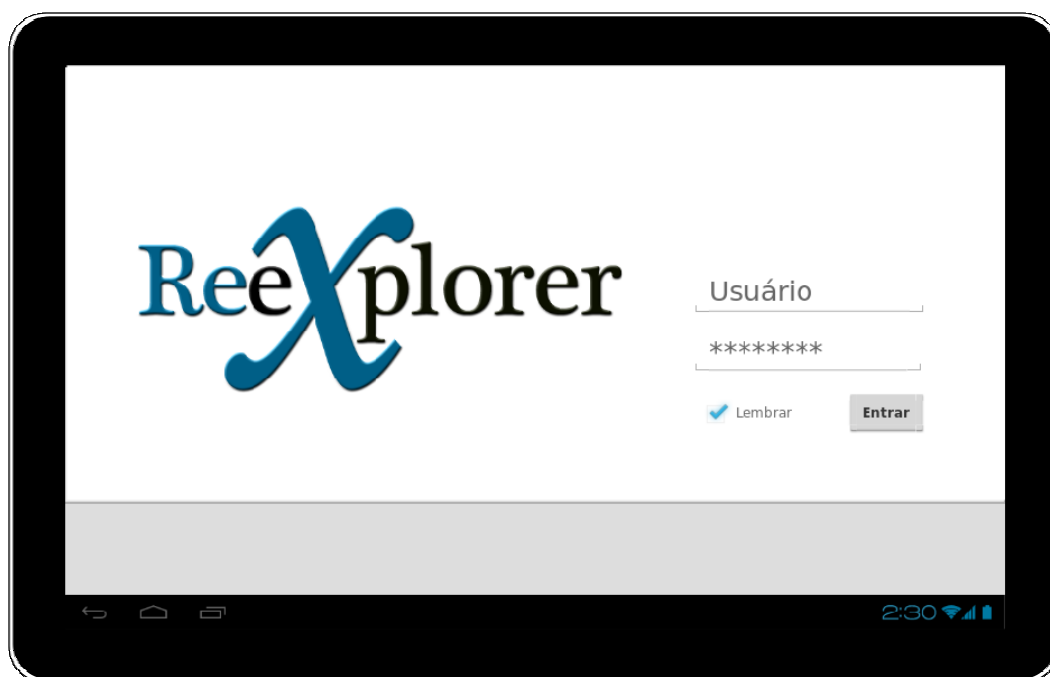
4 DESENVOLVIMENTO

Com a pesquisa e a definição dos objetivos da ferramenta foram criadas as telas principais da aplicação que aparecem logo a seguir. Como mostra o protótipo, para acessar a ferramenta, o usuário deverá logar-se na aplicação (ver tela 4.1.1).

Após a autenticação é apresentada a tela principal da aplicação (ver tela 4.1.2). A tela principal pode ser customizada pelo usuário para apresentar as funcionalidades mais utilizadas durante o processo de levantamento de requisitos. A lista completa de funcionalidades só é apresentada no modelo de pesquisa (ver tela 4.1.4). A tela de requisitos (ver tela 4.1.5) mostra como é feita a entrada de dados pelo usuário e a tela de exportação (ver tela 4.1.7) apresenta as opções de exportação de documentos que estão disponíveis para o usuário.

4.1 PROTÓTIPOS DE TELAS

4.1.1 Tela de login



4.1.2 Home



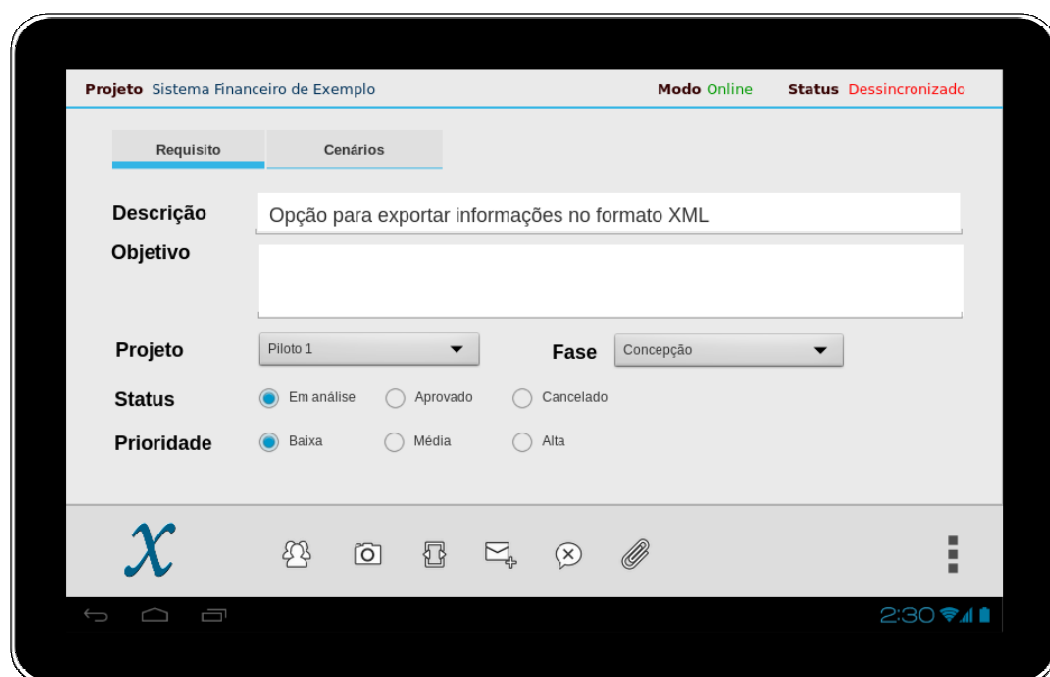
4.1.3 Menu principal



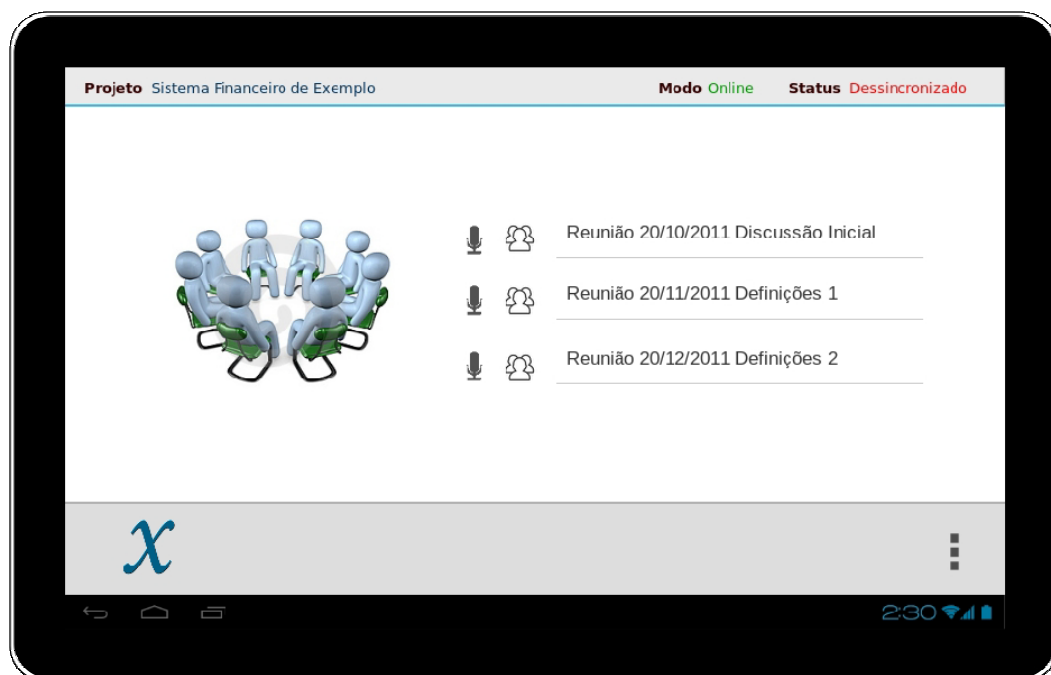
4.1.4 Tela de pesquisa



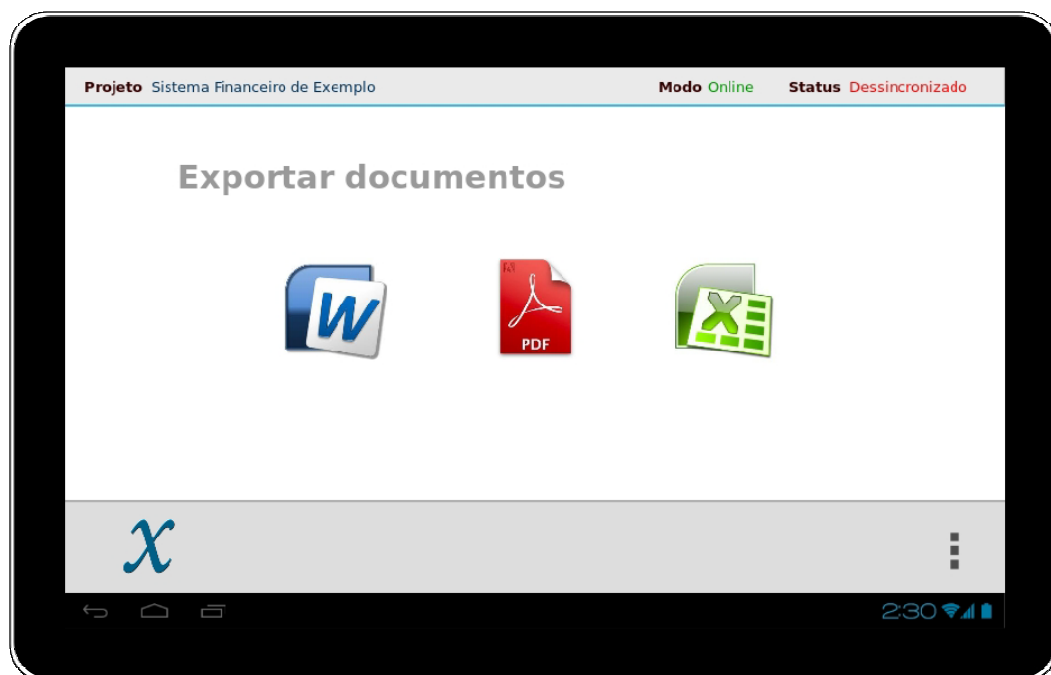
4.1.5 Tela de requisitos



4.1.6 Tela de reuniões

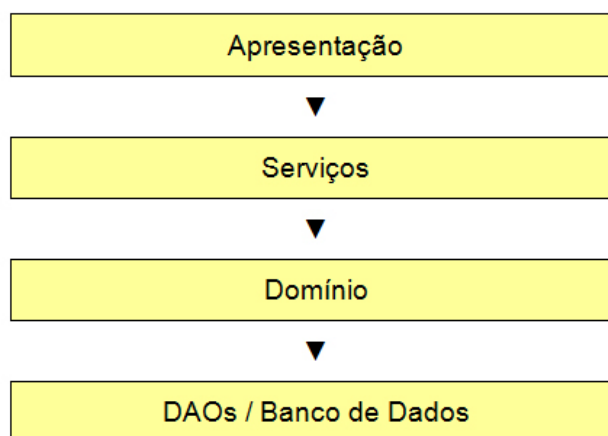


4.1.7 Tela de exportação de documentos



4.2 ARQUITETURA

A aplicação possui 4 camadas lógicas conforme a ilustração:



Apresentação camada responsável pela apresentação das interfaces com o usuário, foi desenvolvida em XML conforme os padrões definidos pela arquitetura de aplicações Android. Nesta camada também estão as Activities que servem para controlar as requisições de tela.

Serviços é a camada que cuida da implementação das regras de negócio contidas nos casos de uso. Camada desenvolvida na linguagem Java.

Domínio é a camada que concentra as classes de negócio da aplicação. Camada desenvolvida na linguagem Java.

Acesso aos dados camada responsável pela comunicação com o banco de dados. Esta camada também foi desenvolvida na linguagem Java.

4.3 REQUISITOS DO SOFTWARE

Os requisitos listados abaixo representam as funcionalidades elicitadas para a ferramenta Reexplorer. O ator principal é o analista de sistema e as funcionalidades listadas contemplam as atividades básicas para o processo de levantamento de requisitos. Os requisitos foram classificados de acordo com a Tabela 1, cuja finalidade é definir a importância dos requisitos de *software* do ponto de vista de cada analista, sendo os valores: **+2** para **concordo totalmente**, **+1** para **concordo parcialmente**, **0** para **estou incerto**, **-1** para **discordo parcialmente** e **-2** para **discordo totalmente**.

O analista poderá:

1. cadastrar projeto;
2. cadastrar envolvido no projeto;
3. cadastrar pergunta no banco de perguntas;
4. cadastrar evento: reunião ou entrevista;
5. cadastrar questionário para entrevista posterior;
6. cadastrar ata de reunião ou entrevista;
7. cadastrar respostas de entrevista;
8. cadastrar documento de visão, requisitos, testes, rastreamento, impacto e entrega;
9. cadastrar requisito;
10. verificar requisito;
11. solicitar mudança em requisitos;
12. anexar arquivo aos requisitos ou regras de negócio;
13. cadastrar caso de uso, incluindo os respectivos cenários e atores;
14. cadastrar caso de teste;
15. cadastrar regra de negócio;
16. imprimir documento de visão, requisitos, casos de uso, casos de teste, rastreamento, impacto, entrega e questionário;
17. exportar PDF de documentos;
18. enviar documento por e-mail;
19. elicitar classe de domínio por caso de uso;
20. elicitar transação de inclusão, pesquisa, alteração ou exclusão por caso de uso.

O sistema deverá:

21. gerar histórico de alteração de situação do projeto.
22. gerar histórico de alteração de documento;
23. gerar histórico de alteração de requisito;
24. gerar histórico de alteração de regra de negócio.
25. sincronizar dados do dispositivo com o sistema de backup na nuvem.

Tabela 1 – Classificação dos requisitos pela Escala de Likert.

Número	Classificação			Média
	Analista 1	Analista 2	Analista 3	
1	+2	+2	+2	+2
2	+2	+2	+2	+2
3	-1	0	+2	+0,33
4	+2	+2	+2	+2
5	-1	0	+2	+0,33
6	-2	-1	+1	-0,67
7	+2	-1	+2	+1
8	+2	+2	+2	+2
9	+2	+2	+2	+2
10	+2	+2	+2	+2
11	+2	+2	+2	+2
12	+2	+2	+2	+2
13	+2	+2	+2	+2
14	+2	+2	+2	+2
15	+2	+2	+2	+2
16	+2	+2	+1	+1,67
17	+2	+2	+1	+1,67
18	+2	+2	+2	+2
19	+2	+2	+2	+2
20	+2	+2	+2	+2
21	+2	+2	+2	+2
22	+2	+2	+2	+2
23	+2	+2	+2	+2
24	+2	+2	+2	+2
25	+2	+2	+2	+2

A média obtida apresenta em sua maioria resultados positivos, isto indica que os requisitos listados para a ferramenta são considerados importantes pelos analistas entrevistados.

4.4 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Os requisitos funcionais e não funcionais estão classificados por códigos. Requisitos funcionais iniciam com as letras RF, não funcionais com RNF e um número sequencial após as letras. Por meio do quadro é possível verificar se o

requisito é evidente, funcionalidade realizada pelo usuário através das telas do sistema, ou oculto, funcionalidade realizada em segundo plano pelo sistema. A prioridade do requisito indica se ele é essencial, precisa ser desenvolvido, importante, pode ser desenvolvido, desejável, seria interessante se fosse desenvolvido. Por fim, o quadro indica os requisitos não funcionais associados ao requisito funcional.

4.4.1 Requisitos Funcionais

CÓDIGO	RF01
DESCRIÇÃO	O analista poderá cadastrar projeto.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	RNF04, RNF05

CÓDIGO	RF02
DESCRIÇÃO	O analista poderá cadastrar envolvido no projeto.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF03
DESCRIÇÃO	O analista poderá cadastrar pergunta no banco de perguntas.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF04
DESCRIÇÃO	O analista poderá cadastrar evento: reunião ou entrevista.
VISIBILIDADE	<input checked="" type="checkbox"/> Evidente () Oculto
PRIORIDADE	() Essencial <input checked="" type="checkbox"/> Importante () Desejável
REQUISITOS ASSOCIADOS	RNF08

CÓDIGO	RF05
DESCRIÇÃO	O analista poderá cadastrar questionário para entrevista.
VISIBILIDADE	<input checked="" type="checkbox"/> Evidente () Oculto
PRIORIDADE	() Essencial <input checked="" type="checkbox"/> Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF06
DESCRIÇÃO	O analista poderá cadastrar ata de reunião ou entrevista.
VISIBILIDADE	<input checked="" type="checkbox"/> Evidente () Oculto
PRIORIDADE	() Essencial () Importante <input checked="" type="checkbox"/> Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF07
DESCRIÇÃO	O analista poderá cadastrar respostas de entrevista.
VISIBILIDADE	<input checked="" type="checkbox"/> Evidente () Oculto
PRIORIDADE	() Essencial <input checked="" type="checkbox"/> Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF08
DESCRIÇÃO	O analista poderá cadastrar documento de visão.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF09
DESCRIÇÃO	O analista poderá cadastrar requisito.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	RNF09, RNF10, RNF11, RNF15

CÓDIGO	RF10
DESCRIÇÃO	O analista poderá verificar requisito.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial (X) Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF11
DESCRIÇÃO	O analista poderá solicitar mudança em requisitos.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial (X) Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF12
DESCRIÇÃO	O analista poderá anexar arquivo aos requisitos ou regras negociais.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial () Importante (X) Desejável
REQUISITOS ASSOCIADOS	RNF16

CÓDIGO	RF13
DESCRIÇÃO	O analista poderá cadastrar caso de uso, incluindo os respectivos cenários e atores.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF14
DESCRIÇÃO	O analista poderá cadastrar caso de teste.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial (X) Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF15
DESCRIÇÃO	O analista poderá cadastrar regra comercial.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial (X) Importante () Desejável
REQUISITOS ASSOCIADOS	RNF12, RNF13, RNF14

CÓDIGO	RF16
DESCRIÇÃO	O analista poderá imprimir documento de visão, requisitos, casos de uso, casos de teste, rastreamento, impacto, entrega e questionário.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial (X) Importante () Desejável
REQUISITOS ASSOCIADOS	RNF17

CÓDIGO	RF17
DESCRIÇÃO	O analista poderá exportar PDF de documentos.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial () Importante (X) Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF18
DESCRIÇÃO	O analista poderá enviar documento por e-mail.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial () Importante (X) Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF19
DESCRIÇÃO	O analista poderá elicitar classe de domínio por caso de uso.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial () Importante (X) Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF20
DESCRIÇÃO	O analista poderá elicitar transação de inclusão, pesquisa, alteração ou exclusão por caso de uso.
VISIBILIDADE	(X) Evidente () Oculto
PRIORIDADE	() Essencial () Importante (X) Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF21
DESCRIÇÃO	O sistema deverá gerar histórico de alteração de situação do projeto.
VISIBILIDADE	() Evidente (X) Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF22
DESCRIÇÃO	O sistema deverá gerar histórico de alteração de documento.
VISIBILIDADE	() Evidente (X) Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF23
DESCRIÇÃO	O sistema deverá gerar histórico de alteração de requisito.
VISIBILIDADE	() Evidente (X) Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF24
DESCRIÇÃO	O sistema deverá gerar histórico de alteração de regra de negócio.
VISIBILIDADE	() Evidente (X) Oculto
PRIORIDADE	(X) Essencial () Importante () Desejável
REQUISITOS ASSOCIADOS	

CÓDIGO	RF25
DESCRIÇÃO	O sistema deverá sincronizar dados do dispositivo com o sistema de backup na nuvem.
VISIBILIDADE	() Evidente (X) Oculto
PRIORIDADE	() Essencial (X) Importante () Desejável
REQUISITOS ASSOCIADOS	RNF02

4.4.2 Requisitos Não Funcionais

CÓDIGO	RNF01
DESCRIÇÃO	A ferramenta de gerenciamento de requisitos será desenvolvida para tablets que rodam o sistema operacional Android.
CATEGORIA	Restrição de projeto
PRIORIDADE	(X) Essencial () Importante () Desejável

CÓDIGO	RNF02
DESCRIÇÃO	Os dados dos projetos serão armazenados no dispositivo em um banco de dados SQLite.
CATEGORIA	Persistência de dados
PRIORIDADE	(X) Essencial () Importante () Desejável

CÓDIGO	RNF03
DESCRIÇÃO	As telas do sistema ficarão restritas ao máximo de 10 elementos/componentes visuais por tela.
CATEGORIA	Usabilidade
PRIORIDADE	() Essencial (X) Importante () Desejável

CÓDIGO	RNF04
DESCRIÇÃO	Os projetos terão as seguintes situações: Ativo, Inativo, Cancelado ou Concluído.
CATEGORIA	Restrição de projeto
PRIORIDADE	(X) Essencial () Importante () Desejável

CÓDIGO	RNF05
DESCRIÇÃO	Os projetos nas situações Inativo, Cancelado ou Concluído ficarão bloqueados para edição.
CATEGORIA	Restrição de projeto
PRIORIDADE	() Essencial (X) Importante () Desejável

CÓDIGO	RNF06
DESCRIÇÃO	Para editar um projeto Inativo, Cancelado ou Concluído o usuário precisará primeiramente reativá-lo.
CATEGORIA	Restrição de projeto
PRIORIDADE	() Essencial (X) Importante () Desejável

CÓDIGO	RNF07
DESCRIÇÃO	O usuário poderá abrir um projeto pré-existente em diferentes aparelhos desde que o dispositivo esteja registrado no sistema de backup na nuvem.
CATEGORIA	Segurança
PRIORIDADE	() Essencial () Importante (X) Desejável

CÓDIGO	RNF08
DESCRIÇÃO	Para agendar um evento os envolvidos deverão estar previamente cadastrados.
CATEGORIA	Restrição de projeto
PRIORIDADE	(X) Essencial () Importante () Desejável

CÓDIGO	RNF09
DESCRIÇÃO	Os requisitos terão as seguintes situações: Aberto, Recusado ou Aprovado.
CATEGORIA	Restrição de projeto
PRIORIDADE	(X) Essencial () Importante () Desejável

CÓDIGO	RNF10
DESCRIÇÃO	Os requisitos nas situações Recusado ou Aprovado ficarão bloqueados para edição.
CATEGORIA	Restrição de projeto
PRIORIDADE	() Essencial () Importante (X) Desejável

CÓDIGO	RNF11
DESCRIÇÃO	Para editar um requisito na situação Recusado ou Aprovado o usuário precisará primeiramente reabri-lo.
CATEGORIA	Restrição de projeto
PRIORIDADE	() Essencial () Importante (X) Desejável

CÓDIGO	RNF12
DESCRIÇÃO	As regras de negócio terão as seguintes situações: Aberta, Recusada ou Aprovada.
CATEGORIA	Restrição de projeto
PRIORIDADE	(X) Essencial () Importante () Desejável

CÓDIGO	RNF13
DESCRIÇÃO	As regras negociais nas situações Recusada ou Aprovada ficarão bloqueadas para edição.
CATEGORIA	Restrição de projeto
PRIORIDADE	() Essencial () Importante (X) Desejável

CÓDIGO	RNF14
DESCRIÇÃO	Para editar uma regra de negócio na situação Recusada ou Aprovada o usuário precisará primeiramente reabri-la.
CATEGORIA	Restrição de projeto
PRIORIDADE	() Essencial () Importante (X) Desejável

CÓDIGO	RNF15
DESCRIÇÃO	Para alterar um requisito na situação Aprovado será obrigatório informar uma solicitação de mudança.
CATEGORIA	Restrição de projeto
PRIORIDADE	(X) Essencial () Importante () Desejável

CÓDIGO	RNF16
DESCRIÇÃO	Os anexos ficarão limitados ao tamanho máximo de 5 MB.
CATEGORIA	Restrição de armazenamento
PRIORIDADE	() Essencial (X) Importante () Desejável

CÓDIGO	RNF17
DESCRIÇÃO	O sistema deve mapear as impressoras através conexão wireless, bluetooth ou nuvem.
CATEGORIA	Restrição de armazenamento
PRIORIDADE	() Essencial (X) Importante () Desejável

4.5 RELAÇÃO ENTRE REQUISITOS FUNCIONAIS E CASOS DE USO

Requisito	Caso de Uso
RF01	UC01 – Cadastrar projeto
RF02	UC02 – Cadastrar envolvido
RF03	UC03 – Cadastrar pergunta
RF04	UC04 – Cadastrar evento
RF05	UC05 – Cadastrar questionário
RF06	UC06 – Cadastrar ata
RF07	UC07 – Cadastrar resposta
RF08	UC08 – Cadastrar visão
RF09	UC09 – Cadastrar requisito
RF10	UC10 – Verificar requisito
RF11	UC11 – Solicitar mudança em requisitos

RF12	UC12 – Anexar arquivo
RF13	UC13 – Cadastrar caso de uso
RF14	UC14 – Cadastrar caso de teste
RF15	UC15 – Cadastrar regra de negócio
RF16	UC16 – Imprimir documento
RF17	UC17 – Exportar PDF
RF18	UC18 – Enviar documento
RF19	UC19 – Elicitar classe
RF20	UC20 – Elicitar transação
RF21	UC21 – Gerar histórico de alteração no projeto
RF22	UC22 – Gerar histórico de alteração de documento
RF23	UC23 – Gerar histórico de alteração de requisito
RF24	UC24 – Gerar histórico de alteração de regra de negócio
RF25	UC25 – Sincronizar dados

Algumas funcionalidades indicadas nos casos de uso são realizadas pelo analista e outras pelo sistema. Os casos de usos relacionados ao analista são funcionalidades executadas pelo usuário através das telas do sistema. Os casos de uso executados pelo sistema ocorrem em segundo plano, de acordo com a interação do usuário, alteração dos dados ou mesmo disponibilidade de conexão com a internet.

Os casos de uso estão codificados; iniciam com as letras UC e um número sequencial após a sigla. Cada especificação (Ver Apêndice) apresenta o ator que realiza o caso de uso, uma descrição do caso de uso e suas pré e pós condições de execução. Além disso, são apresentados os fluxos básico e alternativo de acordo com a complexidade de cada caso de uso.

4.6 RASTREAMENTO DE REQUISITOS

As matrizes de rastreabilidade, mostradas nas tabelas, mapeiam os requisitos funcionais e não funcionais para trás identificando o solicitante da funcionalidade e para frente identificando os artefatos que foram criados com base nos requisitos.

De acordo com as matrizes de rastreabilidade para trás (ver quadros 4.6.1 e 4.6.2), todos os requisitos funcionais e não funcionais foram solicitados pelo analista.

Isto significa que quaisquer alterações nestes requisitos precisam ser validadas com o analista responsável pela definição inicial. Na matriz de rastreabilidade para frente de requisitos funcionais (ver quadro 4.6.3) é possível identificar cada caso de uso que se originou baseado no requisito, isto implica que qualquer alteração no requisito terá um impacto direto no caso de uso relacionado. A matriz de rastreabilidade para frente de requisitos não funcionais (ver quadro 4.6.4) é possível identificar mais de um requisito relacionado a apenas um caso de uso, isto implica que o caso de uso terá que ser alterado na modificação de qualquer um dos requisitos.

4.6.1 Matriz de rastreabilidade para trás de Requisitos Funcionais

Requisito/Envolvido	Analista
RF01	x
RF02	x
RF03	x
RF04	x
RF05	x
RF06	x
RF07	x
RF08	x
RF09	x
RF10	x
RF11	x
RF12	x
RF13	x
RF14	x
RF15	x
RF16	x
RF17	x
RF18	x
RF19	x
RF20	x
RF21	x
RF22	x

RF23	x
RF24	x
RF25	x

4.6.2 Matriz de rastreabilidade para trás de Requisitos Não Funcionais

Requisito/Envolvido	Analista
RNF01	x
RNF02	x
RNF03	x
RNF04	x
RNF05	x
RNF06	x
RNF07	x
RNF08	x
RNF09	x
RNF10	x
RNF11	x
RNF12	x
RNF13	x
RNF14	x
RNF15	x
RNF16	x
RNF17	x

5 RESULTADOS

Inicialmente, espera-se que a ferramenta facilite a atividade de levantamento de requisitos realizada por analistas e engenheiros de *software*. Esta atividade tem sido realizada ao longo dos anos por meio de anotações sobre papel, algumas vezes por dispositivos de áudio e e-mails que ajudam na compreensão e definição das necessidades dos interessados. Com o surgimento dos *tablets*, novas possibilidades surgem e novas ferramentas podem ser criadas para ajudar nesta atividade. Como a proposta deste trabalho foi desenvolver uma ferramenta que fornece suporte aos profissionais da área, facilitando assim as atividades relacionadas à documentação de *software*, também foram desenvolvidas telas para a definição de templates para casos de uso e casos de teste que permitem ao usuário gerar os documentos básicos para o desenvolvimento de um projeto.

6 CONSIDERAÇÕES FINAIS

No decorrer deste trabalho foi possível observar a existência de duas linhas de pensamento, uma delas favorável ao processo clássico de engenharia que se vale de documentos para o desenvolvimento de *software*, e outra, que dispensa o tempo gasto para elaboração de documentos, valorizando a presença de um representante do cliente no ambiente de desenvolvimento para sanar as dúvidas dos desenvolvedores.

O trabalho contribuiu para o entendimento de que a documentação de *software* é uma característica específica da engenharia clássica de *software* e pouco utilizada na modelo de desenvolvimento ágil. Ele foi baseado essencialmente nas ideias da linha clássica, mas busca evoluir as atividades de documentação para se aproximar da agilidade defendida pela linha de pensamento ágil.

Futuramente, com uma base de dados formada, serão realizadas pesquisas visando identificar novas abordagens para geração de documentos. A ferramenta será utilizada inicialmente no desenvolvimento de um projeto de forma clássica e posteriormente para estudar sua utilização em um ambiente de desenvolvimento ágil.

7 REFERÊNCIAS BIBLIOGRÁFICAS

AMBLER, S. W. *Introduction to User Stories*. Disponível em:
<<http://www.agilemodeling.com/artifacts/userStory.htm>> Acesso em: 05/03/13. 2012.

BECK, K.; et al. *Manifesto para Desenvolvimento Ágil de Software*, 2001.
Disponível em: <<http://agilemanifesto.org/iso/ptbr/>>. Acesso em: 23/05/2013.

BOEHM, B.W. *A spiral model of software development and enhancement*, 1988.
Disponível em: <<http://csse.usc.edu/csse/TECHRPTS/1988/usccse88-500/usccse88-500.pdf>>. Acesso em: 29/09/2013.

BOGAZZI, S. *Clarifying Requirements – User Needs*. Julho, 2008. Disponível em:
<<http://techdoertimes.wordpress.com/2008/07/23/clarifying-requirements-user-needs/>> Acesso em: 07/03/2013.

DIJKSTRA, E. *The Humble Programmer*, 1972. Disponível em:
<<http://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD340.html>>. Acesso em: 20/05/2013.

GODOWSKI, M; CZYRNEC, D. *Requirement Managment in Praticce*, 2005.
Disponível em: <<http://dl.acm.org/citation.cfm?id=1565166>>. Acesso em: 18/05/2013.

HEWLETT-PACKARD. *Seven best practices for business-ready applications*. In:
Business White paper. Ver. 2. Maio, 2012.

IBM Corporation. Rational Unified Process: *Elementos Essenciais do Processo*.
Versão 7.1.1, 2007. Disponível em:
<http://www.wthreex.com/rup/v711_ptbr/index.htm>. Acesso em: 10/03/2013.

JACOBSON, I., BOOCH, G., RUMBAUGH, J., *The Unified Software Development Process*. Addison-Wesley, 1999.

LEFFINGWELL, D. **Agile Software Requirements**. 1st edition. Addison-Wesley, 2011. Kindle Edition.

LEFFINGWELL, D.; WIDRIG, D.; **Managing Software Requirements**. Addison Wesley, Second Edition, 2003.

MARTIN, R. C. **Agile Software Development, Principles, Patterns, and Practices**. 1st edition. Prentice Hall, 2002.

PRESSMAN, R. S. **Software Engineering, a Practitioner's Approach – Alternate Edition**. Seventh Edition. McGraw Hill, 2010.

RUPP, C. **Is the industry just getting rid of requirements?**, 2011. Disponível em: <<http://crisys.cs.umn.edu/re2012/keynotes.shtml#3>>. Acesso em: 23/05/2013.

SOARES, I.; FERREIRA, L. **Melhorando a comunicação com Estórias do Usuário**: Uma forma eficaz de aprimorar a comunicação com seus clientes. Revista Java Magazine. Rio de Janeiro: Ed. 86, ano 7, p. 68, 70, 2010.

SOMMERVILLE, I. **Software Engineering**. Addison-Wesley. Boston: 9th Edition, 2011.

ZIELCZYNSKI, P. **Traceability from Use Cases to Test Cases**. 2006. Disponível em: <<http://www.ibm.com/developerworks/rational/library/04/r-3217/>> Acesso em: 07/03/13.

1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica no menu principal e seleciona a opção Incluir Novo Projeto (FA1)	
	4. O sistema apresenta a tela de cadastro de projeto
5. O analista preenche os dados do projeto: Nome, Descrição, Data Inicial e Previsão de Conclusão	
6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva o novo projeto e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Projeto	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza os projetos que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema define o projeto como padrão e abre-o para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir o projeto?" Com as opções Confirmar e Cancelar (FA3)

3. O analista clica no botão Confirmar	
	4. O sistema exclui o projeto e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna à tela pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC02 – Cadastrar envolvido
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de novo envolvido ou alteração ou exclusão de envolvido pré-existente.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto.
PÓS-CONDIÇÕES	Novo envolvido cadastrado ou envolvido pré-existente alterado ou excluído.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica no menu principal e seleciona a opção Incluir Novo Envolvido (FA1)	
	4. O sistema apresenta a tela de

	cadastro de envolvido
5. O analista preenche os dados do envolvido: Nome, Sobrenome, E-mail, Envolvimento	
6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva o novo envolvido criado e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Envolvido	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza os envolvidos que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre o envolvido para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir o envolvido?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui o envolvido e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA

1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se os campos foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC03 – Cadastrar pergunta
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de nova pergunta ou alteração ou exclusão de uma pergunta pré-existente.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto.
PÓS-CONDIÇÕES	Nova pergunta cadastrada ou pergunta pré-existente alterada ou excluída.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica no menu principal e seleciona a opção Incluir Nova Pergunta (FA1)	
	4. O sistema apresenta a tela de cadastro de pergunta
5. O analista preenche os dados da pergunta	
6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva a nova pergunta e apresenta a mensagem de sucesso

Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Pergunta	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza as perguntas que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre a pergunta para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir a pergunta?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui a pergunta e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo

	obrigatório não tenha sido preenchido o sistema apresenta a mensagem: “Preencher campos obrigatórios!”. O caso de uso retorna ao passo 5 do fluxo básico.
--	---

NOME	UC04 – Cadastrar evento	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de novo evento ou alteração ou exclusão de evento pré-existente.	
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto.	
PÓS-CONDIÇÕES	Novo evento cadastrado ou evento pré-existente alterado ou excluído.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O analista clica no menu principal e seleciona a opção Incluir Novo Evento (FA1)	
		4. O sistema apresenta a tela de cadastro de evento
	5. O analista preenche os dados do evento	
	6. O analista clica no botão Salvar (FA4)	
		7. O sistema salva o novo evento e apresenta a mensagem de sucesso
Fluxo Alternativo 1		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O usuário seleciona a opção Localizar Evento	
		2. O sistema apresenta a tela de pesquisa

3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza os eventos que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre o evento para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: “Tem certeza que deseja excluir o evento?” Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui o evento e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: “Preencher campos obrigatórios!”. O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC05 – Cadastrar questionário	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de novo questionário ou alteração ou exclusão de questionário pré-existente.	
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto.	
PÓS-CONDIÇÕES	Novo questionário cadastrado ou questionário pré-existente alterado ou excluído.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O analista clica no menu principal e seleciona a opção Incluir Novo Questionário (FA1)	
		4. O sistema apresenta a tela de cadastro de questionário
	5. O analista preenche os dados do questionário	
	6. O analista clica no botão Salvar (FA4)	
		7. O sistema salva o novo questionário criado e apresenta a mensagem de sucesso
Fluxo Alternativo 1		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O usuário seleciona a opção Localizar Questionário	
		2. O sistema apresenta a tela de pesquisa
	3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
		4. O sistema localiza os questionários que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão

5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre o questionário para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir o questionário?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui o questionário e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC06 – Cadastrar ata
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de nova ata ou alteração ou exclusão de ata pré-existente.

PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. A reunião ou entrevista que originou a ata ter sido previamente cadastrada.	
PÓS-CONDIÇÕES	Nova ata cadastrada ou ata pré-existente alterada ou excluída.	
Fluxo Básico		
AÇÃO DOS ATORES	AÇÃO DO SISTEMA	
1. O analista solicita a abertura do aplicativo		
	2. O sistema apresenta a tela principal	
3. O analista clica no menu principal e seleciona a opção Incluir Nova Ata (FA1)		
	4. O sistema apresenta a tela de cadastro de ata	
5. O analista preenche os dados da ata		
6. O analista clica no botão Salvar (FA4)		
	7. O sistema salva a nova ata e apresenta a mensagem de sucesso	
Fluxo Alternativo 1		
AÇÃO DOS ATORES	AÇÃO DO SISTEMA	
1. O usuário seleciona a opção Localizar Ata		
	2. O sistema apresenta a tela de pesquisa	
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar		
	4. O sistema localiza as atas que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão	
5. O usuário clica no botão Selecionar (FA2)		
	6. O sistema abre a ata para edição	
Fluxo Alternativo 2		

AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: “Tem certeza que deseja excluir a ata?” Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui a ata e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: “Preencher campos obrigatórios!”. O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC07 – Cadastrar resposta
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de nova resposta ou alteração ou exclusão de resposta pré-existente.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O questionário e as perguntas que originaram as respostas terem sido previamente cadastrado.
PÓS-CONDIÇÕES	Nova resposta ou resposta pré-existente alterada ou excluída.
Fluxo Básico	

AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica no menu principal e seleciona a opção Incluir Novo Resposta (FA1)	
	4. O sistema apresenta a tela de cadastro de resposta
5. O analista preenche os dados da resposta	
6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva a nova resposta e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Resposta	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza as respostas que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre a resposta para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir a resposta?" Com

	as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui a resposta e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC08 – Cadastrar visão
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de nova visão ou alteração ou exclusão de visão pré-existente.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto.
PÓS-CONDIÇÕES	Nova visão cadastrada ou visão pré-existente alterada ou excluída.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica no menu principal e	

seleciona a opção Incluir Novo Visão (FA1)	
	4. O sistema apresenta a tela de cadastro de visão
5. O analista preenche os dados da visão	
6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva a nova visão e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Visão	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza as visões que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre a visão para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir a visão?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui a visão e retorna para a tela de pesquisa
Fluxo Alternativo 3	

AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC09 – Cadastrar requisito
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de novo requisito ou alteração ou exclusão de requisito pré-existente.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O envolvido que solicitou o requisito ter sido previamente cadastrado.
PÓS-CONDIÇÕES	Novo requisito cadastrado ou requisito pré-existente alterado ou excluído.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica no menu principal e seleciona a opção Incluir Novo Requisito (FA1)	
	4. O sistema apresenta a tela de cadastro de requisito
5. O analista preenche os dados do requisito	

6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva o novo requisito e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Requisito	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza os requisitos que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre o requisito para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir o requisito?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui o requisito e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa

Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC10 – Verificar requisito
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a verificação de requisito.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O requisito ter sido previamente cadastrado.
PÓS-CONDIÇÕES	Requisito verificado.

Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica na opção Localizar Requisito	
	4. O sistema apresenta a tela de pesquisa
5. O analista preenche os dados de pesquisa e clica no botão Pesquisar	
	6. O sistema localiza todos os registros que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
7. O analista seleciona o registro para edição	
	8. O sistema apresenta a tela de

	verificação
9. O analista verifica o Requisito e clica no botão Salvar (FA1)	
	10. O sistema salva o registro e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 4 do fluxo básico.

NOME	UC11 – Solicitar mudança em requisitos
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a solicitação de mudança para requisito previamente aceito.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O requisito ter sido previamente cadastrado e aceito.
PÓS-CONDIÇÕES	Solicitação gravada.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista seleciona a opção Solicitação de Mudança (FA1)	
	4. O sistema apresenta a tela de solicitação
5. O analista preenche os dados da solicitação	
6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva o registro e

	apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Solicitação	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza os registros que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre o registro para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir a solicitação?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui o registro e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA

	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 4 do fluxo básico.
--	---

NOME	UC12 – Anexar arquivo	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a anexação de arquivo ao requisito ou regra de negócio.	
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O requisito ou a regra de negócio que será anexado o arquivo ter sido previamente cadastrado ou estar em cadastramento.	
PÓS-CONDIÇÕES	Arquivos anexado.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O analista seleciona a opção Requisito ou Regra de Negócio	
		4. O sistema apresenta a tela de edição
	5. O analista clica em anexar arquivo	
		6. O sistema apresenta a tela de seleção de arquivo
	7. O analista seleciona o arquivo e clica em confirmar (FA1)	
		8. O sistema anexa o arquivo ao requisito ou à regra de negócio
	9. O analista clica no botão Salvar (FA1)	
		10. O sistema salva o registro e apresenta a mensagem de sucesso

Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 4 do fluxo básico.

NOME	UC13 – Cadastrar caso de uso
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de novo caso de uso ou alteração ou exclusão de caso de uso pré-existente.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O requisito que vai originar o caso de uso previamente cadastrado.
PÓS-CONDIÇÕES	Novo caso de uso cadastrado ou caso de uso pré-existente alterado ou excluído.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista clica no menu principal e seleciona a opção Incluir Novo Caso de Uso (FA1)	
	4. O sistema apresenta a tela de cadastro de caso de uso
5. O analista preenche os dados do caso de uso	
6. O analista clica no botão Salvar (FA4)	
	7. O sistema salva o novo caso de uso e apresenta a mensagem de sucesso
Fluxo Alternativo 1	

AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Localizar Caso de Uso	
	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza todos os casos de uso que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre o caso de uso para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir o caso de uso?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui o caso de uso e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo

	obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.
--	---

NOME	UC14 – Cadastrar caso de teste	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de novo caso de teste ou alteração ou exclusão de um caso de teste pré-existente.	
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O requisito referente ao caso de teste ter sido previamente cadastrado.	
PÓS-CONDIÇÕES	Novo caso de teste cadastrado ou caso de teste pré-existente alterado ou excluído.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O analista clica no menu principal e seleciona a opção Incluir Novo Caso de Teste (FA1)	
		4. O sistema apresenta a tela de cadastro de caso de teste
	5. O analista preenche os dados do caso de teste	
	6. O analista clica no botão Salvar (FA4)	
		7. O sistema salva o novo caso de teste e apresenta a mensagem de sucesso
Fluxo Alternativo 1		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O usuário seleciona a opção Localizar Caso de Teste	

	2. O sistema apresenta a tela de pesquisa
3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	4. O sistema localiza todos os casos de teste que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre o caso de teste selecionado para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: “Tem certeza que deseja excluir o caso de teste?” Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui o caso de teste e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: “Preencher campos obrigatórios!”. O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC15 – Cadastrar regra de negócio	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a inclusão de nova regra de negócio ou alteração ou exclusão de regra de negócio pré-existente.	
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O envolvido que apresentou a regra de negócio ter sido previamente cadastrado.	
PÓS-CONDIÇÕES	Nova regra de negócio cadastrada ou regra de negócio pré-existente alterada ou excluída.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O analista clica no menu principal e seleciona a opção Incluir Nova Regra de Negócio (FA1)	
		4. O sistema apresenta a tela de cadastro de regra de negócio
	5. O analista preenche os dados da regra de negócio	
	6. O analista clica no botão Salvar (FA4)	
		7. O sistema salva a nova regra de negócio e apresenta a mensagem de sucesso
Fluxo Alternativo 1		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O usuário seleciona a opção Localizar Regra de Negócio	
		2. O sistema apresenta a tela de pesquisa
	3. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
		4. O sistema localiza todas as regra de negócio que atendam aos dados

	de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
5. O usuário clica no botão Selecionar (FA2)	
	6. O sistema abre a regra de negócio para edição
Fluxo Alternativo 2	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O usuário seleciona a opção Excluir	
	2. O sistema apresenta a mensagem: "Tem certeza que deseja excluir a regra de negócio?" Com as opções Confirmar e Cancelar (FA3)
3. O analista clica no botão Confirmar	
	4. O sistema exclui a regra de negócio e retorna para a tela de pesquisa
Fluxo Alternativo 3	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista clica no botão Cancelar	
	2. O sistema retorna para a tela de pesquisa
Fluxo Alternativo 4	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos obrigatórios foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC16 – Imprimir documento
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a impressão de documentos.

PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O documento que deve ser impresso ter sido previamente cadastrado.	
PÓS-CONDIÇÕES	Documento impresso.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O analista seleciona a opção Documento	
		4. O sistema apresenta a tela de pesquisa
	5. O analista preenche os dados de pesquisa e clica no botão Pesquisar	
		6. O sistema localiza os registros que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção
	7. O analista seleciona o documento para imprimir	
		8. O sistema apresenta a tela de impressão
	9. O analista seleciona a impressora	
	10. O analista preenche o número de cópias	
	11. O analista clica em imprimir	
		12. O sistema imprime o documento

NOME	UC17 – Exportar PDF
ATORES	Analista de sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a exportação de documentos.
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O documento que será exportado ter sido previamente cadastrado.
PÓS-CONDIÇÕES	Documento exportado.

Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O analista seleciona a opção Documento	
	4. O sistema apresenta a tela de pesquisa
5. O analista preenche os dados de pesquisa e clica no botão Pesquisar	
	6. O sistema localiza os registros que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção
7. O analista seleciona o documento para exportar	
	8. O sistema apresenta a tela de exportação
9. O analista seleciona a pasta destino do arquivo	
10. O analista clica em exportar	
	11. O sistema exporta o documento

NOME	UC18 – Enviar documento	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo o envio de documentos por e-mail.	
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O documento que será enviado ter sido previamente cadastrado.	
PÓS-CONDIÇÕES	Documento enviado.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O analista seleciona a opção Documento	
		4. O sistema apresenta a tela de pesquisa
	5. O analista preenche os dados de pesquisa e clica no botão Pesquisar	
		6. O sistema localiza os registros que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção
	7. O analista seleciona o documento para enviar	
		8. O sistema apresenta a tela de envio
	9. O analista seleciona o destinatário	
	10. O analista clica em enviar	
		11. O sistema envia o documento

NOME	UC19 – Elicitar classe	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a elicitação de classes de domínio existentes nos casos de uso do projeto em desenvolvimento.	

PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O caso de uso que será usado na elicitação ter sido previamente cadastrado.
PÓS-CONDIÇÕES	Elicitação gravada.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista solicita a abertura do aplicativo	
	2. O sistema apresenta a tela principal
3. O usuário seleciona a opção Localizar Caso de Uso	
	4. O sistema apresenta a tela de pesquisa
5. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
	6. O sistema localiza todos os casos de uso que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
7. O usuário clica no botão Selecionar	
	8. O sistema abre o caso de uso para edição
9. O analista identifica classes candidatas no caso de uso e clica em elicitar classe	
	10. O sistema apresenta a tela de elicitação
11. O analista preenche o nome da classe	
12. O analista clica no botão Salvar (FA1)	
	13. O sistema salva o registro e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos foram preenchidos. Caso algum campo obrigatório não

	<p>tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.</p>
--	--

NOME	UC20 – Elicitar transação	
ATORES	Analista de sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a elicitación de transações existentes nos casos de uso do projeto em desenvolvimento.	
PRÉ-CONDIÇÕES	O projeto ter sido previamente cadastrado e estar aberto. O caso de uso que será usado na elicitación ter sido previamente cadastrado.	
PÓS-CONDIÇÕES	Elicitación gravada.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista solicita a abertura do aplicativo	
		2. O sistema apresenta a tela principal
	3. O usuário seleciona a opção Localizar Caso de Uso	
		4. O sistema apresenta a tela de pesquisa
	5. O usuário preenche os dados de pesquisa e clica no botão Pesquisar	
		6. O sistema localiza todos os casos de uso que atendam aos dados de pesquisa e apresenta na forma de listagem com botões para seleção e exclusão
	7. O usuário clica no botão Selecionar	
		8. O sistema abre o caso de uso para edição
	9. O analista identifica transações candidatas no caso de uso e clica em elicitar transação	
		10. O sistema apresenta a tela de elicitación

11. O analista preenche o nome da transação	
12. O analista clica no botão Salvar (FA1)	
	13. O sistema salva o registro e apresenta a mensagem de sucesso
Fluxo Alternativo 1	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O sistema valida se todos os campos foram preenchidos. Caso algum campo obrigatório não tenha sido preenchido o sistema apresenta a mensagem: "Preencher campos obrigatórios!". O caso de uso retorna ao passo 5 do fluxo básico.

NOME	UC21 – Gerar histórico de alteração no projeto
ATORES	Sistema
DESCRIÇÃO	Este caso de uso tem por objetivo a geração de histórico de alterações realizadas na definição do projeto em desenvolvimento.
PRÉ-CONDIÇÕES	As informações do projeto em desenvolvimento terem sido alteradas.
PÓS-CONDIÇÕES	Histórico gravado.
Fluxo Básico	
AÇÃO DOS ATORES	AÇÃO DO SISTEMA
1. O analista faz alguma alteração nos dados do projeto e clica em salvar	
	2. O sistema salva o projeto
	3. O sistema grava registro de log da situação anterior do projeto, data de alteração e usuário

NOME	UC22 – Gerar histórico de alteração de documento
ATORES	Sistema

DESCRIÇÃO	Este caso de uso tem por objetivo a geração de histórico de alterações realizadas nos documentos do projeto em desenvolvimento.	
PRÉ-CONDIÇÕES	As informações do documento terem sido alteradas.	
PÓS-CONDIÇÕES	Histórico gravado.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista faz alguma alteração nos dados do documento e clica em salvar	
		2. O sistema salva o documento
		3. O sistema grava registro de log da situação anterior do documento, data de alteração e usuário

NOME	UC23 – Gerar histórico de alteração de requisito	
ATORES	Sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a geração de histórico de alterações realizadas nos requisitos do projeto em desenvolvimento.	
PRÉ-CONDIÇÕES	As informações do requisito terem sido alteradas.	
PÓS-CONDIÇÕES	Histórico gravado.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista faz alguma alteração nos dados do requisito e clica em salvar	
		2. O sistema salva o requisito
		3. O sistema grava registro de log da situação anterior do requisito, data de alteração e usuário

NOME	UC24 – Gerar histórico de alteração de regra de negócio	
ATORES	Sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a geração de histórico de alterações realizadas nas regras de negócio do projeto em desenvolvimento.	

PRÉ-CONDIÇÕES	As informações da regra de negócio terem sido alteradas.	
PÓS-CONDIÇÕES	Histórico gravado.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
	1. O analista faz alguma alteração nos dados da regra de negócio e clica em salvar	
		2. O sistema salva a regra de negócio
		3. O sistema grava registro de log da situação anterior da regra de negócio, data de alteração e usuário

NOME	UC25 – Sincronizar dados	
ATORES	Sistema	
DESCRIÇÃO	Este caso de uso tem por objetivo a sincronização dos dados do aplicativo com servidor.	
PRÉ-CONDIÇÕES	O dispositivo estar conectado ao servidor.	
PÓS-CONDIÇÕES	Histórico gravado.	
Fluxo Básico		
	AÇÃO DOS ATORES	AÇÃO DO SISTEMA
		1. O sistema identifica que o dispositivo está conectado ao servidor e os dados do servidor estão desatualizados ou que os dados do dispositivo estão desatualizados
		2. O sistema sincroniza os dados do dispositivo com o servidor e do servidor com o dispositivo

As principais classes da aplicação são as classes Requisito e Documento. Na classe Requisito estão os atributos que definem a especificação do *software* em desenvolvimento e a situação atual do requisito. Em seu entorno estão as classes secundárias com objetivo complementar o requisito de *software*, entre elas RequisitoFuncional e RequisitoNaoFuncional.

A classe Documento é a classe base para todos os documentos de *software*. Relacionados a ela temos as classes DocumentoImpacto, DocumentoEntrega DocumentoRastreamento e DocumentoVisao que são as principais classes relacionadas à documentação de *software* que a ferramenta produz.