

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO  
ESPECIALIZAÇÃO EM ENGENHARIA DE SOFTWARE

DIEGO STIEHL

**APLICAÇÃO DA ENGENHARIA DE REQUISITOS NO ESTUDO  
EXPERIMENTAL DE UM SISTEMA DE GERÊNCIA DE EVENTOS**

MONOGRAFIA DE ESPECIALIZAÇÃO

MEDIANEIRA

2011

DIEGO STIEHL

**APLICAÇÃO DA ENGENHARIA DE REQUISITOS NO ESTUDO  
EXPERIMENTAL DE UM SISTEMA DE GERÊNCIA DE EVENTOS**

Monografia apresentada como requisito parcial à obtenção do título de Especialista na Pós Graduação em Engenharia de Software, da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Medianeira.

Orientador: Prof. Me. Alan Gavioli.

Coorientador: Prof. Me. Fernando Schütz.

MEDIANEIRA

2011



---

## TERMO DE APROVAÇÃO

### **Aplicação da Engenharia de Requisitos no Estudo Experimental de um Sistema de Gerência de Eventos**

Por  
**Diego Stiehl**

Esta monografia foi apresentada às 10h20min do dia 15 de dezembro de 2011 como requisito parcial para a obtenção do título de Especialista no curso de Especialização em Engenharia de Software, da Universidade Tecnológica Federal do Paraná, *Câmpus* Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Me. Alan Gavioli  
UTFPR – Câmpus Medianeira  
(orientador)

---

Prof. Me. Fernando Schutz  
UTFPR – Câmpus Medianeira

---

Prof. M.Eng. Juliano Lamb  
UTFPR – Câmpus Medianeira

## RESUMO

STIEHL, Diego. Aplicação da Engenharia de Requisitos no Estudo Experimental de um Sistema de Gerência de Eventos. Monografia (Especialização em Engenharia de Software). Universidade Tecnológica Federal do Paraná, Medianeira, 2011.

Este trabalho tem por objetivo apresentar a engenharia de requisitos, suas ferramentas e técnicas, através da aplicação destas em um estudo experimental que visa o levantamento e a especificação dos requisitos de um sistema de controle de eventos para a Universidade Tecnológica Federal do Paraná, Câmpus Medianeira. Os produtos da realização deste estudo experimental são os documentos que contêm a declaração formal do que deve ser projetado e implementado pelo sistema proposto.

**Palavras-chave:** *software*, documento, requisito, levantamento, especificação.

## ABSTRACT

STIEHL, Diego. Application of Requirements Engineering in an Experimental Study of an Event Manager System. Monografia (Especialização em Engenharia de Software). Universidade Tecnológica Federal do Paraná, Medianeira, 2011.

This paper aims to present the requirements engineering, its tools and techniques, by applying them to experimental study that aims the elicitation and specification of requirements for an event manager system that will be used by Universidade Tecnológica Federal do Paraná, Campus Medianeira. The products of this study case are documents that contain the formal statement of what must be designed and implemented by the proposed system.

**Keywords:** software, document, requirement, elicitation, specification.

## LISTA DE FIGURAS

Figura 1 - Tipos de requisitos não funcionais .....	21
Figura 2 - Visão espiral do processo de engenharia de requisitos .....	41
Figura 3 - Relação do modelo de análise com outras etapas.....	45
Figura 4 - Classificação dos tipos de diagrama da UML .....	47
Figura 5 – Representação dos atores em diagramas de casos de uso.....	49
Figura 6 – Representação de casos de uso em diagrama de casos de uso .....	49
Figura 7 - Diagrama de caso de uso .....	50
Figura 8 - Ações em um diagrama de atividades .....	51
Figura 9 – Nós indicativos de início (a) e fim (b) de diagrama de atividades.....	51
Figura 10 - Diagrama de atividades .....	52
Figura 11 - Diagrama de classes.....	53
Figura 12 - Diagrama de sequência .....	54
Figura 13 - Usuários de um documento de requisitos .....	65
Figura 14 - Diagrama de casos de uso do sistema de controle de eventos .....	77
Figura 15 - Diagrama de classes do sistema de controle de eventos .....	83

## LISTA DE QUADROS

Quadro 1 - Envolvidos no processo de desenvolvimento de <i>software</i> .....	25
Quadro 2 - Exemplo de cenário principal e extensões .....	37
Quadro 3 - Modelo para apresentação dos problemas do projeto .....	61
Quadro 4 - Estrutura de apresentação do resumo dos usuários .....	61
Quadro 5 - Apresentação da lista de riscos.....	63
Quadro 6 - Exemplo de requisito com caso de uso.....	79
Quadro 7 - Requisitos não funcionais do sistema de controle de eventos .....	82

## LISTA DE TABELAS

Tabela 1 - Forma de especificação de requisitos .....	39
Tabela 2 - Tipos de diagramas oficiais da UML .....	46
Tabela 3 - Propriedades do plano de gerenciamento de requisitos .....	56
Tabela 4 - Responsabilidades da equipe de <i>software</i> .....	57
Tabela 5 - Tipos de requisitos .....	58
Tabela 6 - Critérios para classificação de fornecedores de requisitos .....	59
Tabela 7 - Estrutura do documento de visão.....	60
Tabela 8 - Estrutura de um documento de requisitos.....	66



## LISTA DE SIGLAS

CORBA	Common Object Request Broker Architecture
FUNTEF	Fundação de Apoio à Educação, Pesquisa e Desenvolvimento Científico e Tecnológico da UTFPR
OMG	Object Management Group
RE	Requirements Engineering
SRS	Software Requirements Specification
UML	Unified Markup Language
UP	Unified Process
UTFPR	Universidade Tecnológica Federal do Paraná

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	OBJETIVO GERAL	12
1.2	OBJETIVOS ESPECÍFICOS	12
1.3	JUSTIFICATIVA	12
1.4	ORGANIZAÇÃO DO TRABALHO	14
<b>2</b>	<b>ENGENHARIA DE REQUISITOS</b>	<b>15</b>
2.1	ENGENHARIA DE SOFTWARE E A ENGENHARIA DE REQUISITOS	15
2.2	REQUISITOS	17
2.2.1	Requisitos Funcionais	18
2.2.2	Requisitos não Funcionais	19
2.2.3	Atributos de Requisitos	21
2.2.3.1	Benefício	22
2.2.3.2	Estabilidade	22
2.2.3.3	Situação	23
2.2.3.4	Risco	23
2.2.3.5	Outros Atributos	24
2.3	INTERESSADOS	24
2.3.1	Identificação dos Fornecedores de Requisitos	26
2.4	LEVANTAMENTO DE REQUISITOS	26
2.4.1	Levantamento de Requisitos em Alto Nível	28
2.4.2	Levantamento de Requisitos Funcionais e não Funcionais	28
2.4.3	Técnicas para levantamento de requisitos	29
2.4.3.1	Entrevistas	29
2.4.3.2	Etnografia	31
2.4.3.3	Outras técnicas para levantamento de requisitos	32
2.4.4	Cenários	33
2.4.5	Casos de Uso	34
2.4.5.1	Cenário principal e extensões	35
2.4.5.2	Outras propriedades de casos de uso	36
2.5	ESPECIFICAÇÃO DE REQUISITOS	37
2.6	VALIDAÇÃO DE REQUISITOS	40
<b>3</b>	<b>MODELAGEM DE REQUISITOS</b>	<b>44</b>
3.1	UML	45
3.1.1	Diagrama de Casos de Uso	48
3.1.2	Diagrama de Atividades	50

3.1.3	Diagrama de Classes .....	52
3.1.4	Diagrama de Sequência .....	54
<b>4</b>	<b>ARTEFATOS.....</b>	<b>55</b>
4.1	PLANO DE GERENCIAMENTO DE REQUISITOS .....	55
4.1.1	Gerenciamento de Requisitos.....	57
4.1.2	Identificação dos Requisitos .....	58
4.1.3	Fornecedores de Requisitos.....	59
4.2	DOCUMENTO VISÃO .....	59
4.2.1	Problema .....	60
4.2.2	Resumo dos Usuários .....	61
4.3	GLOSSÁRIO.....	61
4.4	LISTA DE RISCOS .....	62
4.5	LISTA DE REQUISITOS.....	63
4.6	DOCUMENTO DE REQUISITOS .....	64
<b>5</b>	<b>ESTUDO EXPERIMENTAL.....</b>	<b>68</b>
5.1	MATERIAIS E MÉTODOS .....	68
5.2	RESULTADOS E DISCUSSÃO .....	69
5.2.1	Início .....	70
5.2.2	Especificação de Requisitos de Negócio.....	72
5.2.3	Estudos de Viabilidade .....	74
5.2.4	Elicitação de Requisitos de Usuário .....	75
5.2.5	Especificação de Requisitos de Usuário.....	77
5.2.6	Prototipação.....	80
5.2.7	Elicitação de requisitos de sistema.....	80
5.2.8	Especificação e modelagem de requisitos de sistema .....	81
5.2.9	Revisões .....	83
<b>6</b>	<b>CONCLUSÃO.....</b>	<b>85</b>
6.1	CONSIDERAÇÕES FINAIS.....	85
6.2	SUGESTÕES DE TRABALHOS FUTUROS.....	86
	<b>REFERÊNCIAS.....</b>	<b>87</b>
	<b>APÊNDICE A - PLANO DE GERENCIAMENTO DE REQUISITOS.....</b>	<b>88</b>
	<b>APÊNDICE B – DOCUMENTO DE VISÃO .....</b>	<b>97</b>
	<b>APÊNDICE C – GLOSSÁRIO DO PROJETO .....</b>	<b>103</b>
	<b>APÊNDICE D – DOCUMENTO DE REQUISITOS DO SISTEMA.....</b>	<b>104</b>

## 1 INTRODUÇÃO

Uma das tarefas mais complexas que é desempenhada pelos engenheiros de *software* é a etapa de reconhecimento e entendimento dos requisitos necessários aos produtos (*softwares*) com os quais irão estes trabalhar. Isto pode, inicialmente, parecer uma tarefa fácil, pois teoricamente, um cliente ou outra pessoa interessada em um sistema deveria ter um bom entendimento de suas características, mas não é isto que ocorre na prática (PRESSMAN, 2011).

Os requisitos de determinado sistema são, basicamente, as descrições do que o mesmo deve fazer, que tipos de serviços este oferece, bem como as restrições a seu funcionamento. Tais requisitos não são simplesmente impostos ou presumidos, e sim, são artefatos que refletem as necessidades e expectativas dos clientes e demais interessados com relação ao sistema ao qual os mesmos pertencem (SOMMERVILLE, 2011).

Quando fala-se do ato de descobrir, analisar, documentar e verificar os requisitos de um *software*, está-se tratando da subárea da engenharia de *software* denominada engenharia de requisitos (*Requirements Engineering* - RE) (SOMMERVILLE, 2011).

Este trabalho objetiva inicialmente estudar e aplicar técnicas e ferramentas da engenharia de requisitos que permitam tanto conhecer as características esperadas pelos interessados em qualquer tipo de aplicação, quanto gerenciar suas expectativas através do envolvimento destes, na medida do possível, no processo de discussão e validação destas características.

O estudo experimental apresentado neste trabalho justifica-se na necessidade do reconhecimento dos requisitos de um *software* para gerência, de forma genérica, de eventos oferecidos pela Universidade Tecnológica Federal do Paraná (UTFPR) Câmpus Medianeira. As técnicas e ferramentas descobertas e explanadas no momento da revisão bibliográfica serão aplicadas visando atingir este fim.

Após a completa utilização destas técnicas e ferramentas, espera-se como resultado um documento de requisitos deste *software*. Este documento será uma declaração oficial do que deve ser tratado pela equipe de desenvolvimento do *software* nas etapas futuras de seu projeto.

Para atingir este objetivo, sabe-se que há a necessidade de estudar e utilizar alguns recursos que permitam alizara modelagem e diagramação dos requisitos, o que permitirá uma melhor visualização dos mesmos, conforme sugerido por Pressman (2011), através da utilização da UML (*Unified Markup Language*).

## 1.1 OBJETIVO GERAL

Construir um documento formal de requisitos para viabilizar a implementação de um sistema de gerenciamento de eventos para a UTFPR Câmpus Medianeira.

## 1.2 OBJETIVOS ESPECÍFICOS

- Estudar conceitos, técnicas e ferramentas da engenharia de requisitos;
- Identificar e aplicar técnicas e ferramentas para modelagem orientada a objetos envolvidas no processo de engenharia de requisitos;
- Efetuar o levantamento, a classificação e a validação de requisitos junto aos interessados no sistema;
- Elaborar um documento formal de requisitos com base nos dados extraídos.

## 1.3 JUSTIFICATIVA

Muitas vezes é comum encontrar sistemas que são desenvolvidos sem qualquer tipo de controle mais minucioso da qualidade do código que é produzido, isto porque o código é simplesmente produzido, sendo deixadas para o momento de sua escrita (ou ignoradas) características consideradas importantes no processo de criação de uma aplicação, como a análise, diagramação ou prototipação, mas principalmente uma bem aplicada engenharia de requisitos, a qual fornece uma base sólida para o projeto e construção da mesma (PRESSMAN, 2011).

Sabe-se que todo projeto de *software* tem um conjunto de requisitos, que são definidos pelas necessidades e expectativas dos usuários que efetivamente farão uso do mesmo, relacionados, obviamente, aos objetivos de negócio de suas respectivas empresas (ENGHOLM JÚNIOR, 2010).

A engenharia de requisitos é, atualmente, considerada muito importante e, também, um passo fundamental para a obtenção de um *software* de qualidade, pois somente através da utilização dela é que se conseguirá obter um produto final (*software*) que realmente atenda ao esperado pelos interessados neste (PRESSMAN, 2011).

Também não é completamente válido dizer que as técnicas de engenharia de *software* são a solução absoluta para os desafios de comunicação e alinhamento com os interessados, mas elas efetivamente fornecem uma abordagem consistente para lidar com os estes (PRESSMAN, 2011).

Para o sucesso na aplicação da engenharia de requisitos sobre qualquer sistema, devem-se abordar técnicas e fazer uso de ferramentas que permitam ações como extração, refinamento, documentação, rastreamento e leitura dos mesmos. A engenharia de requisitos engloba uma série de tarefas de modelagem que levam a especificação dos requisitos e representação do projeto do sistema a ser construído (PRESSMAN, 2011).

Sabendo destas características, este trabalho objetiva aplicar a engenharia de requisitos sobre um estudo experimental específico: a necessidade de um sistema para controle genérico de eventos por parte da Universidade Tecnológica Federal do Paraná (UTFPR) – Câmpus Medianeira, sabendo que esta atualmente possui diversos aplicativos desenvolvidos para eventos específicos que são, sempre que possível, reutilizados para novos eventos. Porém sabe-se que aplicativos desenvolvidos com fins específicos dificilmente atenderão de forma completa outros fins (um dos motivos para isto, provavelmente é a divergência em seus requisitos), inclusive nesta área de gerência de eventos, onde as pessoas envolvidas com a organização de cada um podem necessitar gerenciar diferentes recursos de formas específicas e únicas.

A aplicação da engenharia de requisitos neste estudo experimental específico se dá através da elaboração de meios que permitam extrair e documentar as necessidades dos interessados no sistema, bem como identificar quem são os

próprios interessados, os quais estão inseridos no universo de funcionários, administrativos e docentes, do câmpus de Medianeira da UTFPR.

#### 1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho possui suas seções organizadas da seguinte maneira:

- **Capítulo 1:** Apresenta a introdução do trabalho, contendo as definições de objetivos e a justificativa do mesmo;
- **Capítulo 2:** Aborda, no formato de revisão bibliográfica, todas as características sobre engenharia de requisitos julgadas relevantes para este trabalho;
- **Capítulo 3:** Trata das formas de se realizar a modelagem dos requisitos levantados e especificados;
- **Capítulo 4:** Explora, ainda com base em revisão bibliográfica, os possíveis documentos de saída gerados pela engenharia de requisitos;
- **Capítulo 5:** Contém os procedimentos e resultados ligados ao estudo experimental realizado;
- **Capítulo 6:** Apresenta as considerações finais e sugestões de trabalhos futuros relacionados a este trabalho.

## 2 ENGENHARIA DE REQUISITOS

Objetivando a obtenção do controle sobre requisitos, cujas gerências estão entre as tarefas mais difíceis durante a criação de um sistema, desenvolveu-se a subárea da engenharia de *software* denominada engenharia de requisitos (ENGHOLM JÚNIOR, 2010).

Engenharia de requisitos envolve todas as atividades de ciclo de vida direcionadas a identificar os requisitos dos usuários, a análise destes requisitos para derivar novos, a documentação destes requisitos como uma especificação e suas respectivas validações de acordo com as necessidades dos usuários, bem como os processos que suportam todas estas atividades (HULL, JACKSON e DICK, 2011).

A engenharia de requisitos é o ramo da engenharia de *software* preocupado com os objetivos, funções e restrições de sistemas perante o mundo real. Ela também se preocupa com o relacionamento destes fatores com especificações precisas do comportamento do *software* e com suas evoluções com o passar do tempo (ZAVE, 1997).

Basicamente, a engenharia de requisitos é a área da engenharia de *software* preocupada com a descoberta, desenvolvimento, planejamento, análise, qualificação, comunicação e gerenciamento de requisitos que definem um sistema sob diversos níveis de abstração (HULL, JACKSON e DICK, 2011).

### 2.1 ENGENHARIA DE SOFTWARE E A ENGENHARIA DE REQUISITOS

O *software* vem, crescentemente, se tornado mais incorporado em praticamente todos os aspectos da vida das pessoas e cada vez mais pessoas têm procurado aderir aos recursos oferecidos por ele. Juntamente a esta incorporação, tem crescido a exigência por *softwares* de qualidade, pois os riscos de suas utilizações têm se tornando cada vez maiores (PRESSMAN, 2011).

A engenharia de *software* surgiu desta necessidade de aumento da qualidade dos produtos de *software*, bem como da diminuição de custos e riscos envolvidos em seu processo de desenvolvimento e da criação de rotinas que possam ser reaproveitadas em diversas fases deste processo, ou até em diferentes



produtos. Pode-se dizer que a engenharia de *software* surgiu objetivando a utilização de princípios de engenharia sobre os já existentes processos de desenvolvimento de *software* (ENGHOLM JÚNIOR, 2010).

Esta visão de engenharia gerou novas possibilidades de obtenção de *software* de qualidade, pois sabe-se que engenheiros fazem coisas funcionarem aplicando teorias, métodos e ferramentas onde julgarem mais apropriado. Mas fazem isto seletivamente, sempre buscando pela solução de problemas, mesmo quando não possuem métodos e teorias aplicáveis. Sabem também que a resolução do problema deve respeitar determinadas restrições, dentre elas organizacionais, financeiras e de cronograma (SOMMERVILLE, 2011).

Nas últimas duas décadas, o preço dos itens de hardware tem sofrido uma constante queda. Porém, devido a diversos fatores, como aumento da complexidade e tamanho, o *software* acabou se tornando a peça mais cara no orçamento geral da computação (ENGHOLM JÚNIOR, 2010).

Este citado aumento da complexidade, acabou ressaltando alguns problemas que se encontravam ocultos. Dentre estes problemas, pode-se citar, segundo Engholm Júnior (2010):

- Imprecisão na estimativa de prazos e custos;
- Dificuldade de se atender às demandas;
- Dificuldade de se obter profissionais qualificados para atender às demandas;
- Qualidade dos *softwares* inferior à necessária, causando insatisfação dos usuários.

Considerando-se isto, a abordagem do desenvolvimento de *software* como uma engenharia fez-se necessária, pois a partir dela foi possível abordar o assunto com ferramentas e técnicas que visam o aumento da produtividade e qualidade de tudo que for produzido. Portanto, devido à importância e alto custo dos produtos de *software*, a engenharia de *software* passa a ter um valor considerável, principalmente no que tange as questões de eficiência e redução de custos (ENGHOLM JÚNIOR, 2010).

A engenharia de *software* tem aplicação sobre todos os aspectos e estágios da criação de *software*, partindo das etapas iniciais de especificação até o momento em que o *software* está em uso e há a necessidade de realizar manutenção no mesmo (SOMMERVILLE, 2011).

Ela não se preocupa apenas com processos técnicos, e sim, vai além, viabilizando ou facilitando atividades de apoio à produção de *software*, como, por exemplo, a gerência de projetos (SOMMERVILLE, 2011).

A engenharia, não só a de *software*, preza pela seleção do método mais adequado para cada circunstância ou grupo de circunstâncias. Portanto, alguns casos podem exigir uma solução mais branda, menos formal. O não entendimento deste ponto acaba por gerar comentários negativos que acabam por denegrir a imagem da engenharia de *software*, sendo esta taxada como um procedimento custoso e não compensável (SOMMERVILLE, 2011).

Uma abordagem muito sistemática, disciplinada e quantificável pode ser perfeitamente viável e necessária em alguns casos (por exemplo, um grande sistema de automação industrial), porém as mesmas técnicas acabariam se tornando pesadas demais em outras perspectivas (por exemplo, uma microempresa que possui uma equipe limitada) (PRESSMAN, 2011).

Sabendo desta questão da necessidade de diferentes abordagens para diferentes casos, Sommerville (2011) identifica alguns grandes grupos de *software* (que possivelmente necessitarão de técnicas internas semelhantes), são eles:

- Aplicações *stand-alone*;
- Aplicações interativas baseadas em transações;
- Sistemas de controle embutidos;
- Sistemas de processamento de lotes;
- Sistemas de entretenimento;
- Sistemas para modelagem e simulação;
- Sistemas de coleta de dados.

## 2.2 REQUISITOS

Requisito é qualquer condição ou capacidade que deve ser implementada por determinado *software* ou componente deste para alcançar determinado fim (ENGHOLM JÚNIOR, 2010).

Para um requisito ser considerado adequado, este deve tanto endereçar uma necessidade direta ou indireta das pessoas que virão a utilizar o sistema quanto

ser formalmente aprovado pelos interessados no projeto, garantindo seu valor e alinhamento aos objetivos deste. Sendo assim, estabelecem-se como boas práticas documentar, organizar e disponibilizar os requisitos aos envolvidos no projeto, objetivando garantir que o entendimento destes seja compartilhado entre os clientes e a equipe de projeto (ENGHOLM JÚNIOR, 2010).

Sabe-se também que todos os projetos de *software* possuem seu conjunto de requisitos, os quais são definidos pelas necessidades e expectativas dos usuários, porém estes são obtidos sempre considerando os objetivos de negócio da empresa onde trabalham (ENGHOLM JÚNIOR, 2010).

Dentre os problemas que podem gerar falhas ou fracassos de projetos de *software*, vários deles estão diretamente ligados a requisitos, como o não entendimento da real necessidade e expectativa dos usuários pelos engenheiros ou pelos próprios usuários, a incompletude na especificação e a mudança dos requisitos com o projeto em andamento ou concluído (ENGHOLM JÚNIOR, 2010).

Existe uma grande variedade de tipos de requisitos, havendo desde requisitos referentes a características e funcionalidades esperadas do sistema pelos usuários até requisitos relacionados com o desempenho, a segurança e confiabilidade do *software* e das informações por ele manipuladas (ENGHOLM JÚNIOR, 2010).

### 2.2.1 Requisitos Funcionais

Requisitos funcionais são declarações de serviço que devem ser oferecidos pelo sistema, de como ele deve reagir a entradas específicas e de como deve se comportar em determinadas situações, podendo ainda explicitar o que o sistema não deve fazer (SOMMERVILLE, 2011).

Requisitos funcionais são os que estabelecem funcionalidades, características ou ações que o sistema deve fornecer. Requisitos funcionais normalmente podem ser visualizados com a ajuda de diagramas de caso de uso. O uso destas características normalmente envolve o atendimento a usuários e outros sistemas (ENGHOLM JÚNIOR, 2010).

Lima (2007) afirma que requisitos funcionais “especificam ações que o sistema deve executar independente de exigências físicas ou tecnológicas e são

associados ao modelo conceitual, isto é, o conjunto de necessidades do cliente que devem ser satisfeitas para resolver um problema”.

Os requisitos funcionais dependem do tipo de *software* a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral utilizada pela equipe de *software* para escrever os requisitos. Quando estes são expressos como requisitos de usuário, normalmente apresentam uma forma altamente abstrata, para possibilitar ao máximo o entendimento do mesmo por parte dos interessados no projeto, porém, por outro lado, eles podem ser escritos como requisitos funcionais de sistema, acabando por serem bem mais específicos, descrevendo em detalhes as funções do sistema, suas entradas e saídas, exceções, dentre outras características (SOMMERVILLE, 2011).

A especificação de requisitos funcionais de um sistema deve ser completa e consistente, onde a completude significa que todos os serviços solicitados pelos interessados devem ser definidos e a consistência significa que diversos requisitos não podem apresentar definições e sentidos contraditórios, pois todas as características voltadas do sistema voltadas ao usuário serão construídas tomando por base as informações levantadas e especificadas nos requisitos funcionais deste (SOMMERVILLE, 2011).

A listagem a seguir apresenta, segundo Sommerville (2011), alguns exemplos de requisitos funcionais de um sistema específico:

- Um usuário deve ser capaz de pesquisar as listas de agendamentos para todas as clínicas;
- O sistema deve gerar a cada dia, para cada clínica, a lista do pacientes para as consultas daquele dia;
- Cada membro da equipe que usa o sistema deve ser identificado apenas por seu número de oito dígitos.

### 2.2.2 Requisitos não Funcionais

Os requisitos não funcionais também são características esperadas de determinado sistema, porém não estão diretamente ligados a serviços específicos que o sistema oferece a seus usuários, podendo estar relacionados às propriedades emergentes do sistema (SOMMERVILLE, 2011).

Os requisitos não funcionais normalmente especificam ou restringem as características de um sistema como um todo, sendo frequentemente mais críticos que requisitos funcionais individuais. Eles surgem das necessidades dos interessados, devido a restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de *software* ou *hardware*, ou a partir de fatores externos, como regulamentos de segurança ou até legislações de privacidade (SOMMERVILLE, 2011).

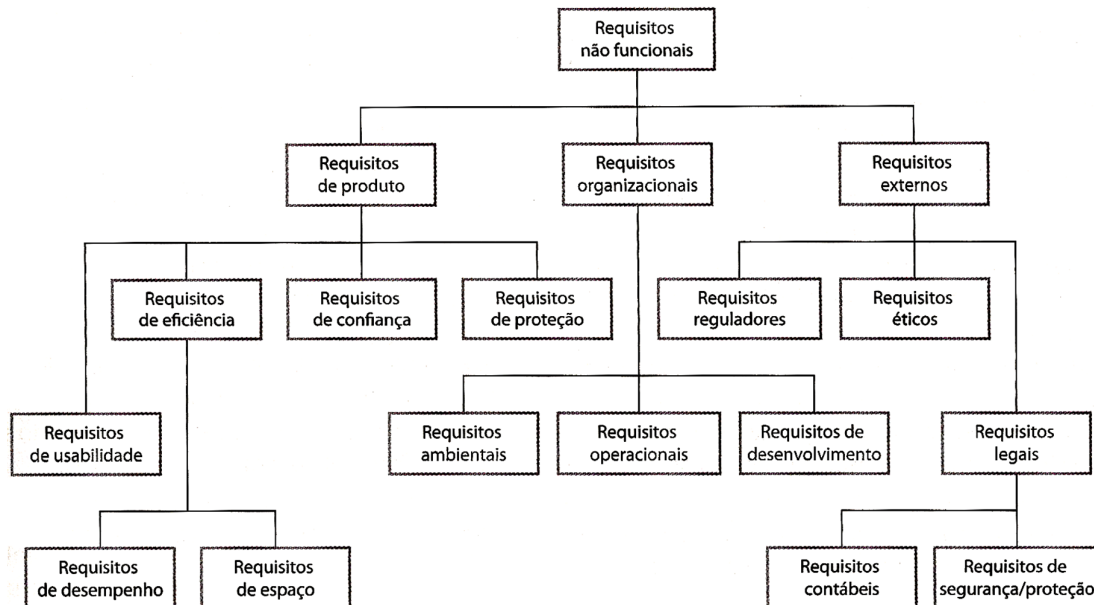
Segundo Engholm Júnior (2010), “requisitos não funcionais descrevem atributos do sistema ou do ambiente do sistema”, tais como:

- Extensibilidade
- Usabilidade
- Confiabilidade
- Desempenho
- Escalabilidade
- Reusabilidade
- Capacidade de manutenção
- Reutilização de código
- Performance
- Eficiência no desenvolvimento
- Confiabilidade nos dados apresentados

Sommerville (2011) sugere uma visão semelhante à de Engholm Júnior (2010), porém vai mais além, realizando a separação dos requisitos não funcionais em categorias hierarquizadas (Figura 1), que os dividem em três grandes grupos:

- **Requisitos de produto:** Especificam ou restringem o comportamento do sistema. Exemplos: a rapidez que o sistema deve executar, quanta memória ele requer, taxa aceitável de falhas, proteção e confiabilidade;
- **Requisitos organizacionais:** São derivados das políticas e procedimentos da organização do cliente e do desenvolvedor. Exemplos: definição de como o sistema será usado, qual linguagem de programação será utilizada, o ambiente de desenvolvimento e ambiente operacional do sistema;

- **Requisitos externos:** Derivam de processos externos ao sistema. Exemplos: especificação de sistema por parte do governo, requisitos legais, requisitos éticos, dentre outros.



**Figura 1 - Tipos de requisitos não funcionais**  
**Fonte: Sommerville (2011)**

### 2.2.3 Atributos de Requisitos

Para possibilitar uma gerência mais eficaz dos requisitos, há a necessidade da definição de atributos para cada um deles, cabendo ao gerente de projetos e o engenheiro de requisitos definir quais atributos são necessários (ENGHOLM JÚNIOR, 2010).

Cada requisito deve ser criado e revisado conjuntamente entre os interessados e a equipe de desenvolvimento do *software* e deve possuir, pelo menos, os atributos apresentados nas seções a seguir. As descrições e pesos dos valores de atributos não precisam, necessariamente, seguir os propostos para cada característica, podendo o gerente de projetos ou sua equipe de *software* definir seus próprios padrões (ENGHOLM JÚNIOR, 2010).

### 2.2.3.1 Benefício

Este atributo indica o grau de benefício, do ponto de vista do usuário, do requisito ao qual este é referente. Faz-se necessário o estabelecimento formal do benefício, pois diferentes interessados podem possuir expectativas diferentes referentes a um mesmo requisito, cabendo então ao engenheiro de *software* fazer a mediação entre eles. Esta etapa é bastante importante, pois permite à equipe de desenvolvimento do *software* saber quais requisitos têm preferência de implementação (ENGHOLM JÚNIOR, 2010).

O atributo de benefício pode ser classificado, segundo Engholm Júnior (2010), como:

- **Crítico:** quando o requisito é imprescindível para o sucesso do *software*;
- **Importante:** quando o não atendimento do requisito não gera fracasso do projeto, mas impacta na satisfação do usuário;
- **Desejável:** caso o requisito não for atendido, o impacto na satisfação do cliente será baixo ou nulo.

### 2.2.3.2 Estabilidade

Este atributo quantifica a possibilidade da ocorrência de mudanças em determinado requisito, o que pode ocorrer devido à alteração no grau de entendimento do mesmo pelos interessados ou pela equipe de *software*. Este grau de entendimento pode variar de acordo com a maturação de conceitos por parte dos interessados (ENGHOLM JÚNIOR, 2010).

O atributo de estabilidade pode ser classificado, segundo Engholm Júnior (2010), como:

- **Alta:** o requisito tem um alto grau de entendimento e baixa probabilidade de mudanças;
- **Média:** o requisito tem considerável chance de mudanças, pois provavelmente ainda não está bem entendido pela equipe de *software* ou ainda há pendências de definições por parte dos interessados;

- **Baixa:** o requisito possui alta probabilidade de mudanças, pois possui baixa maturidade, sendo este nível geralmente associado a requisitos de alta complexidade ou com muitas pendências de definição.

### 2.2.3.3 Situação

A situação de um requisito é um atributo importante, pois indica o estado de aprovação no qual se encontra cada requisito (ENGHOLM JÚNIOR, 2010).

O atributo da situação pode ser classificado, segundo Engholm Júnior (2010), como:

- **Proposto:** o atributo proposto ainda não pode ser utilizado na implementação (ou outras fases do *software*), pois ainda se encontra em análise pela equipe de *software* e muitas vezes até pelo cliente;
- **Aprovado:** o atributo está apto a ser utilizado, pois já obteve aprovação e acordo entre a equipe de *software* e os interessados;
- **Cancelado:** o requisito com esta situação deve ser descartado do projeto.

### 2.2.3.4 Risco

Neste atributo são tratados os riscos oferecidos por cada requisito, sendo que os mesmos serão futuramente tratados pela disciplina de gerenciamento de riscos. Entende-se como requisitos de risco os que podem vir a prejudicar o projeto direta ou indiretamente, principalmente nas áreas financeiras e de cronograma do mesmo. Exemplos de requisitos que devem ter os riscos minuciosamente traçados são: requisitos de transferência de dinheiro entre contas-correntes e requisitos que necessitem determinadas tecnologias, nas quais os profissionais encontram-se escassos no mercado (ENGHOLM JÚNIOR, 2010).

O risco pode ser classificado, segundo Engholm Júnior (2010), como:

- **Alto:** o requisito tem baixa estabilidade e alta complexidade, contendo dependências externas ao *software*;



- **Médio:** o requisito tem estabilidade e complexidade consideradas medianas, não contendo dependências externas;
- **Baixo:** o requisito tem baixo risco por ter alta estabilidade e baixa complexidade, não contendo dependências externas.

#### 2.2.3.5 Outros Atributos

Durante a definição da estrutura a ser utilizada para a documentação dos requisitos, é recomendável a utilização de atributos que a equipe de *software* julgar necessários para o completo entendimento do problema. Outros exemplos de atributos comumente utilizados em complemento aos apresentados são: o nome do responsável pelo requisito e uma área reservada para observações diversas sobre características especiais de determinado requisito (ENGHOLM JÚNIOR, 2010).

### 2.3 INTERESSADOS

Segundo Pressman (2011), “interessado é qualquer um que tenha interesse no êxito de um projeto – executivos, usuários finais, engenheiros de *software*, o pessoal do suporte etc.”. Estes interessados (ou, do inglês: *stakeholders*), também são conhecidos como fornecedores de requisitos, pois é deles que serão extraídos os conhecimentos de negócio e necessidades para serem aplicados em funcionalidades do *software* a ser desenvolvido (ENGHOLM JÚNIOR, 2010).

O interesse de envolvidos no sistema pode ser caracterizado por diferentes motivos, dentre eles: o interessado fará uso do sistema, será beneficiado com o sistema (financeiramente ou outra vantagem), será prejudicado pelo sistema (em termos de custo ou outro dano), será responsável pelo sistema ou será indiretamente afetado pelo uso do mesmo (HULL, JACKSON e DICK, 2011).

Pressman (2011) afirma que, “todo projeto de *software* é formado por interessados que podem ser categorizados sob cinco grupos”, são eles:

- **Gerentes sêniores:** definem os itens de negócio e, com frequência, exercem influência significativa no projeto;

- **Gerentes (técnicos) de projetos:** devem planejar, motivar, organizar e controlar os programadores;
- **Programadores:** devem ter habilidades técnicas necessárias para desenvolver a engenharia de um *software*;
- **Clientes:** especificam os requisitos para o *software* a ser submetidos ao processo de engenharia e outros envolvidos que têm interesses periféricos no produto final;
- **Usuários finais:** interagem com o *software* uma vez que liberado para uso operacional, em ambiente de produção.

Qualquer projeto de *software* é composto por pessoas e equipes que se encaixam nestes agrupamentos, sendo que para as equipes serem efetivas e produzirem os resultados esperados, cada uma delas deve estar organizada e possuir uma liderança interna, o que possibilitará maximizar a capacidade de cada profissional (PRESSMAN, 2011).

Engholm Júnior (2010) apresenta uma separação dos possíveis envolvidos em um projeto dos lados do cliente, do desenvolvimento, bem como envolvidos que participam de ambos (Quadro 1).

Cliente	Desenvolvimento
<p style="text-align: center;">Dono do negócio Gerentes Usuários</p>	<p style="text-align: center;">Gerente de projeto Analista de requisitos Analista de negócios Arquiteto de <i>software</i> Designer de <i>software</i> Programador</p>
<p style="text-align: center;">Analista de qualidade Especialista de implantação</p>	

**Quadro 1 - Envolvidos no processo de desenvolvimento de *software***  
**Fonte: ENGHOLM JÚNIOR (2010)**

### 2.3.1 Identificação dos Fornecedores de Requisitos

Uma ação de grande relevância no tanto no processo da engenharia de requisitos, quanto no projeto como um todo, é a etapa na qual se realiza a identificação das pessoas que virão a se tornar fornecedores de requisitos (ENGHOLM JÚNIOR, 2010).

Para tal, há a necessidade de realizar, logo de início, contato com as fontes iniciais de requisitos para identificar pessoas de determinados setores da empresa que podem vir a se tornar fontes de requisitos, sabendo que estas deverão ser impactadas de alguma forma pelo projeto (ENGHOLM JÚNIOR, 2010).

Estas pessoas costumam e devem ser das mais diversas áreas e competências: partindo do patrocinador ao usuário final, passando pela equipe de projeto e áreas técnicas do cliente como qualidade, produção, banco de dados, tecnologias etc. Qualquer fonte de requisitos que for esquecida nesta etapa pode gerar um incômodo retrabalho, frustração e até o fracasso do projeto (ENGHOLM JÚNIOR, 2010).

Um bom critério para a escolha dos fornecedores de requisitos costuma ser escolher os gestores do negócio e alguns usuários finais como fornecedores preferencias, porém pessoas com alçada para tomada de decisão também devem ser consideradas, junto a pessoas que avaliarão e aprovarão o produto final, o patrocinador, dentre outros (ENGHOLM JÚNIOR, 2010).

## 2.4 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos, o elicitacão de requisitos, é (ou deveria ser) a fase inicial de todo projeto, tendo um elevado nível de importância para o sucesso do mesmo, pois antes de iniciá-lo, precisa-se compreender claramente todos os objetivos e restrições pertinentes a este (ENGHOLM JÚNIOR, 2010).

É nesta etapa que se toma conhecimento das principais expectativas dos interessados no projeto, permitindo à equipe de *software* desenvolver um sistema que se adapte da melhor forma possível ao que estes desejam (ENGHOLM JÚNIOR, 2010).

Um trabalho bem realizado durante a etapa de levantamento de requisitos possibilita não somente o alcance de objetivos finais do projeto de *software*, como a satisfação do usuário, acarretando, também, em objetivos internos deste, como: a possibilidade de um correto planejamento das atividades, o estudo de viabilidade e a estimativa de custos gerais do projeto (ENGHOLM JÚNIOR, 2010).

O levantamento de requisitos pode ser uma tarefa muito difícil, pois depende diretamente da participação e manifestação opinião de diversos interessados no *software* e, como normalmente estes interessados são pessoas ocupadas pelos cargos que atuam em suas empresas, esta ação fica ainda mais prejudicada, pois os mesmos não possuem tempo, ou não dão devido valor, para as ações que visam proporcionar o levantamento de requisitos (ENGHOLM JÚNIOR, 2010).

Segundo Engholm Júnior (2010), “a elicitação de requisitos pode ser entendida como o processo de descoberta e entendimento do domínio do problema a ser atendido pelo projeto, além das necessidades de negócio a serem contempladas”, devendo este procedimento ser realizado em conjunto com os fornecedores de requisitos indicados pela organização cliente ou subárea de negócio desta, a ser atendida pelo projeto (ENGHOLM JÚNIOR, 2010).

As informações necessárias para a obtenção dos requisitos podem ser levantadas com a utilização de alguns recursos, como: entrevistas, reuniões, análise de documentos, análise de sistemas já utilizados pela empresa, observação de tarefas executadas no dia-a-dia da empresa (ou subárea desta), dentre outros, sendo sempre recomendada a formalização das informações levantadas em documentos específicos, como uma ata de reunião, que possibilite a ciência e concordância por parte dos interessados (ENGHOLM JÚNIOR, 2010).

Após o reconhecimento inicial das informações obtidas, estas devem ser filtradas e alinhadas ao problema de negócio a ser atendido no projeto, com o intuito de eliminar requisitos discrepantes ou repetitivos (ENGHOLM JÚNIOR, 2010).

Todos os procedimentos adotados durante a elicitação de requisitos devem estar registrados em um documento conhecido como “plano de gerenciamento de requisitos”. Este documento prescreve todas as orientações de interpretação e uso dos requisitos presentes no “documento de requisitos” do sistema em questão, devendo conter todos os procedimentos a serem adotados, como técnicas a serem utilizadas na elicitação de requisitos, atributos de requisitos, rastreabilidade (se desejada), dentre outros (ENGHOLM JÚNIOR, 2010).

### 2.4.1 Levantamento de Requisitos em Alto Nível

A elicitação de requisitos em alto nível refere-se à documentação do resultado obtido após os primeiros contatos entre equipe de desenvolvimento e demais interessados. Estes contatos têm por objetivo delimitar o escopo do sistema, levantar os objetivos do projeto, levantar todas as particularidades a serem disponibilizadas pelo sistema a ser desenvolvido e entender as expectativas da empresa a serem contempladas pelo sistema (ENGHOLM JÚNIOR, 2010).

Algumas das atividades do levantamento de requisitos em alto nível, segundo Engholm Júnior (2010), são:

- **Entendimento do problema:** no final deste processo deve haver um consenso entre a equipe de projeto e as fontes de requisitos no entendimento do problema, sendo esta uma condição de fundamental importância para o futuro do projeto;
- **Identificação de todos os envolvidos no projeto:** é de fundamental importância que todos os envolvidos sejam identificados e consultados nesta fase;
- **Identificação das necessidades:** com os interessados já identificados, precisa-se levantar todas as necessidades e expectativas para o *software* (o que o *software* deverá fazer), bem como deve-se identificar a necessidade de comunicação com outros sistemas;
- **Elaboração de glossário do projeto:** o glossário visa estabelecer um vocabulário comum entre a equipe de projeto e os fornecedores de requisitos, devendo conter abreviações, siglas e termos técnicos da área do cliente.

### 2.4.2 Levantamento de Requisitos Funcionais e não Funcionais

Neste subprocesso, dá-se o início, por parte do analista de requisitos, ao levantamento do domínio do *software* a ser desenvolvido, levantando propriedades e capacidades que ele deverá ofertar para atingir às necessidades de negócio do projeto (ENGHOLM JÚNIOR, 2010).

Nesta etapa objetiva-se, segundo Engholm Júnior (2010), “propor uma solução que atenda às necessidades dos interessados, identificando as capacidades que o sistema deve fornecer e os requisitos de *software* do aplicativo, levantar características do produto a ser desenvolvido e aprovar o escopo do sistema”.

O levantamento das características do produto a ser desenvolvido deve especificar todas as características e requisitos do mesmo. Segundo Engholm Júnior (2010), nesta etapa podem ser encontrados tanto requisitos funcionais, quanto requisitos não funcionais. Alguns exemplos destes são:

- O sistema deve utilizar a plataforma Java;
- O sistema deve ter perfis de acesso mostrando funcionalidades compatíveis com cada um;
- O sistema deve controlar o acesso de usuários;
- O sistema deve armazenar histórico de vendas.

Também deve-se partir do princípio que todo requisito deve ser rastreável, ou seja, deve ter sua origem em um caso de uso de negócio, devendo este ter sua origem nos fornecedores de requisitos ou ter a ciência e aprovação dos mesmos, pois só assim se evitará a implementação de requisitos indesejados (ENGHOLM JÚNIOR, 2010).

### 2.4.3 Técnicas para levantamento de requisitos

Para a realização da etapa de levantamento de requisitos, há a necessidade da aplicação de uma ou mais técnicas. Estas técnicas têm por objetivo auxiliar a equipe de *software* na extração das necessidades dos interessados no *software*.

#### 2.4.3.1 Entrevistas

Entrevistas realizadas com os interessados, sejam elas formais ou não, são as atividades mais realizadas durante a engenharia de requisitos. Em entrevistas a equipe de *software* questiona os interessados sobre o sistema que existe atualmente (caso exista), e sobre o sistema que será desenvolvido (SOMMERVILLE, 2011).

Segundo Sommerville (2011), entrevistas podem ser de dois tipos:

- **Fechadas:** os interessados respondem a um conjunto predefinido de perguntas;
- **Abertas:** Não existe uma agenda predefinida, ficando por conta da equipe de *software* explorar uma série de questões com os interessados no sistema, desenvolvendo, assim, a melhor compreensão das necessidades do sistema.

Na prática, as duas formas de entrevista acabam sendo aplicadas complementarmente, pois cada um dos formatos possui locais e momentos nos quais são melhores aplicadas. Discussões totalmente abertas raramente funcionam bem e um formulário totalmente preestabelecido provavelmente resultaria na incompletude ou falta de requisitos. Pode-se, por exemplo, começar uma entrevista com um questionário fechado, porém as respostas de algumas perguntas podem levar a outras questões, as quais podem vir a ser abertas, por serem mais abrangentes, acabando por exigir uma abordagem menos estruturada e mais flexível (SOMMERVILLE, 2011).

Entrevistas são úteis para extrair dos interessados informações gerais sobre que ações estes realizam na empresa, como desejam interagir com o novo sistema e quais as dificuldades que têm com sistemas já existentes. Normalmente as pessoas gostam de falar sobre seu trabalho e ficam felizes com as entrevistas por estarem podendo colaborar com o andamento do projeto. Porém, deve-se atentar ao fato de que entrevistas não são tão úteis para obter-se a compreensão dos requisitos do domínio da aplicação (SOMMERVILLE, 2011).

Segundo Sommerville (2011), “pode ser difícil elicitare conhecimento do domínio por meio de entrevistas por duas razões”, são elas:

- Especialistas em aplicações utilizam jargões e terminologias específicos a um domínio, achando impossível discutir os requisitos deste domínio sem os mesmos, fazendo seus usos de maneira precisa e sutil, o que acaba por dificultar o entendimento por parte dos engenheiros de requisitos;
- Como determinado conhecimento de domínio é tão familiar aos interessados, eles acabam tendo certa dificuldade ao explicá-lo, ou pensam que este é tão fundamental, que acabam por acreditar na não necessidade de não mencionar algumas características do mesmo.

Outro item exposto por Sommerville (2011) é a de que “entrevistadores eficazes têm duas características”, são elas:

- São abertos a novas ideias, evitando ideias preconcebidas sobre requisitos e estando dispostos a ouvir os interessados, mesmo que o que for por ele proposto o faça mudar de ideia sobre o sistema;
- Sabem estimular os entrevistados (interessados) a participar de discussões através da utilização de uma “questão-trampolim”, uma proposta de requisitos ou trabalhando em conjunto em um protótipo do sistema, pois quanto mais abstrato for o nível das perguntas por ele formuladas, mais abstratas, e inúteis, serão as respostas para cada uma delas.

As informações recolhidas em entrevistas são, normalmente, suplementares a outras informações sobre o sistema, oriundas de outras formas de extração, como documentos que descrevem os processos de negócio da empresa ou sistemas existentes, observações do usuário, dados históricos, dentre outros. A entrevista por si só pode acabar por não tratar de algumas informações essenciais, devendo esta, portanto, ser utilizada em conjunto com outras técnicas de elicitación de requisitos que forem julgadas necessárias pela equipe (SOMMERVILLE, 2011).

#### 2.4.3.2 Etnografia

Os sistemas não existem de forma isolada. Eles são utilizados dentro de um determinado contexto social e organizacional. Portanto, sabe-se que requisitos deste *software* podem ser derivados ou restringidos deste contexto, porque, geralmente, satisfazer estes requisitos sociais e organizacionais é crucial para o sucesso do *software* (SOMMERVILLE, 2011).

Segundo Sommerville (2011), “etnografia é uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos” e, quando se fala em engenharia de requisitos, etnografia refere-se ao ato de um analista de requisitos inserir-se no ambiente de trabalho no qual o *software* será utilizado, cabendo a ele observar e documentar o trabalho do dia-a-dia da equipe de futuros usuários.



Uma das vantagens da etnografia como forma de elicitação de requisitos é que ela ajuda a descobrir requisitos implícitos que refletem a forma real como as pessoas realizam determinadas ações, pois a real forma de aplicação destas ações pode diferir ou ignorar, por qualquer motivo, os processos formais estabelecidos pela organização, cabendo ao analista de requisitos identificar qual a forma a ser adotada no *software* (SOMMERVILLE, 2011).

Etnografia é uma técnica auxiliar a ser considerada durante o levantamento de requisitos, pois frequentemente as pessoas (usuários) julgam ser de grande dificuldade o ato de expressar os detalhes do trabalho por elas realizado, pois isto já é um ato automático para ela. Estas pessoas entendem e sabem realizar o próprio trabalho, porém não conseguem estabelecer a relação entre ele e o dos demais funcionários da organização (SOMMERVILLE, 2011).

Estudos etnográficos podem trazer à tona detalhes cruciais de processos que normalmente são ignorados por outras técnicas de levantamento de requisitos. Porém, como tem o foco no usuário final, esta abordagem nem sempre adequada a descoberta de requisitos organizacionais ou de domínio, não podendo, sempre, identificar novos recursos que devem ser incrementados no sistema. Por estes motivos, sabe-se que a etnografia não é uma abordagem completa para a elicitação de requisitos, devendo ser utilizada como forma complementar a outras técnicas (SOMMERVILLE, 2011).

#### 2.4.3.3 Outras técnicas para levantamento de requisitos

Além das entrevistas e da imersão da equipe de *software* no ambiente de trabalho dos futuros usuários do *software*, outras técnicas para levantamento de requisitos podem ser aplicadas, cabendo à equipe de *software* analisar quando cada uma delas pode vir a ser necessária. Exemplos destas são:

- **Workshops:** apresentação de ideias e resultados aos interessados através da utilização de palestras;
- **Questionários:** elaboração de questionários, com perguntas abertas e fechadas, direcionando-os às possíveis ou comprovadas fontes de requisitos;

- **Brainstorms:** são “tempestades de ideias” que são obtidas através de reuniões onde interessados e equipe de software oferecem, discutem e documentam suas opiniões sobre o funcionamento do *software*;
- **Observação:** um membro da equipe de software observa a utilização de *software* legado pelo usuário.

#### 2.4.4 Cenários

As pessoas, principalmente as não pertencentes à área de *software*, conseguem mais facilmente compreender exemplos da vida real do que descrições abstratas (SOMMERVILLE, 2011).

Por este motivo, pode ser útil o estabelecimento de alguns cenários operacionais, que mostram como os usuários podem interagir com o *software*. Isto acaba por aumentar a capacidade por parte destes de compreender e criticar as características do mesmo (SOMMERVILLE, 2011).

À medida que os requisitos são levantados, começa-se a ter uma visão geral das funções e características do *software*, porém é difícil progredir para as próximas etapas da engenharia de requisitos sem que a equipe de *software* tenha entendido bem como estas funções e características serão utilizadas pelos diferentes tipos usuários do *software*. Por isto a equipe de *software* pode, juntamente aos usuários e interessados, criar um conjunto de cenários que identifique um roteiro de uso das mesmas. Estes cenários são também conhecidos como “casos de uso” (PRESSMAN, 2011).

Cenários são úteis para realizar a adição de detalhes a uma descrição geral de requisitos, pois são descrições de exemplos de sessões de interação, devendo cada cenário cobrir um pequeno número de interações. Portanto, diversos cenários devem ser desenvolvidos, com o intuito de oferecer diversos tipos de informação sob diferentes níveis de detalhamento do sistema (SOMMERVILLE, 2011).

Sommerville (2011) afirma que um cenário tem seu início através de um esboço da interação a ser realizada, porém, durante o processo de elicitación, detalhes são adicionados a este esboço, com o intuito de criar uma descrição completa desta interação. De uma forma geral, segundo o mesmo autor, os detalhes de cenários podem ser:

- Uma descrição do que o sistema e os usuários esperam quando o cenário de iniciar;
- Uma descrição do fluxo normal de eventos;
- Uma descrição do que pode dar errado e como isto deve ser tratado;
- Informações sobre outras atividades que podem acontecer ao mesmo tempo;
- Uma descrição do estado do sistema quando o cenário acaba.

O levantamento de requisitos baseado em cenários é caracterizado pelo trabalho conjunto com os interessados, para que estes possam identificar cenários e capturar detalhes dos mesmos, podendo estes cenários ser descritos de forma textual, suplementados por diagramas, telas, dentre outros artefatos. Porém há também a possibilidade de uma abordagem mais estruturada, nas quais podem ser utilizados cenários de eventos ou casos de uso.

#### 2.4.5 Casos de Uso

Casos de uso é uma técnica utilizada para a descoberta de requisitos que, apesar de, atualmente, possuir forte ligação à notação UML, não foi introduzida por esta e sim adaptada a ela (SOMMERVILLE, 2011).

O que realmente dá valor e sentido a um caso de uso é sua descrição textual, na forma de cenários, o que não é formalmente estabelecido pela notação UML, que apresenta somente uma definição bastante rala, pois “nada na UML descreve como você deve capturar o conteúdo de um caso de uso”. A notação foca na apresentação do diagrama de casos de uso, que mostra como utilizar os casos de uso relacionados entre si, o que acaba sendo de valor bastante limitado (FOWLER, 2005).

Um caso de uso, em sua forma mais pura, dá nome a uma interação entre o sistema e agentes externos a ele, conhecidos como atores. A interação é então suplementada por informações adicionais que descrevem a interação com o sistema, podendo esta informação adicional ser feita de forma textual ou utilizando outros modelos gráficos, como outras formas de diagramas (SOMMERVILLE, 2011).

Um ator é um papel desempenhado por um usuário perante o sistema. O termo usuário costuma causar confusão de entendimento desta característica, pois

os usuários em questão neste ponto não precisam ser usuários físicos ou humanos do sistema, podendo ser outros objetos ou até sistemas que interajam com o mesmo. Seguindo esta linha de pensamento, pode-se assumir que o termo ator não é tão bem aplicado quanto outras possibilidades válidas, como, por exemplo, “papel” (FOWLER, 2005).

Na fase de levantamento de requisitos, os casos de uso são arquivados em um diagrama de casos de uso de alto nível. Este diagrama contém um conjunto de casos de uso que representa todas as possíveis interações que serão descritas nas etapas seguintes da engenharia de requisitos (SOMMERVILLE, 2011).

Há uma forte ligação e comum confusão entre os conceitos de casos de uso e cenários, não havendo distinção entre os mesmos que seja simples e rápida. Há duas formas de se especificar casos de uso. Na primeira, considera-se cada caso de uso como um cenário único, porém, a outra forma especifica que um caso de uso pode encapsular uma série de cenários, sendo cada cenário um segmento através do caso de uso, onde há um cenário principal e os demais são cenários alternativos, que representarão a possibilidade do acontecimento de exceções na execução do primeiro (SOMMERVILLE, 2011).

Os casos de uso levantados na etapa de elicitação de requisitos devem ser expandidos nas próximas (ou até mesmo posteriormente nesta mesma etapa), devendo ser documentados com uma completa descrição textual que permita esclarecer todo o funcionamento da interação com o sistema da qual trata, podendo haver ligação (na forma de referências) com outros documentos e artefatos da engenharia de *software* como um todo, visando fornecer uma visão das características afetadas e que afetam os mesmos (SOMMERVILLE, 2011).

#### 2.4.5.1 Cenário principal e extensões

Sabe-se que casos de uso são formas padronizadas já estabelecidas para representação de um ou mais cenários de uso de um sistema. Porém, um caso de uso criado pode possuir subdivisões de cenários. Embora não formalizadas por uma notação específica, estas subdivisões são uma boa prática conhecida. Elas consistem em agrupar o fluxo interno de um caso de uso em dois formatos de cenário: cenário principal e extensões (FOWLER, 2005).

Lima (2007) os apresenta sob os nomes de, respectivamente, fluxo básico e fluxos alternativos e Larman (2004) chama o primeiro de cenário de sucesso principal.

O cenário principal descreve o caminho típico de sucesso que satisfaz aos desejos dos interessados que originaram o caso de uso. Para sua criação deve ser estabelecida uma sequência de ações numerada que consiga demonstrar toda a execução do caso de uso, sem considerar a ocorrência de desvios ou problemas durante a mesma (LARMAN, 2004).

As extensões indicam todos os outros cenários ou fluxos, sendo estes de sucesso ou de fracasso. Estes fluxos alternativos são ramos do cenário principal, estando estes, sempre ligados à determinada ação numerada especificada no cenário principal (LARMAN, 2004).

#### 2.4.5.2 Outras propriedades de casos de uso

Larman (2004) aponta algumas outras propriedades que fazem-se úteis durante a escrita de um caso de uso, são elas:

- **Ator principal:** ator que chama os serviços do sistema para atingir um objetivo;
- **Lista de interessados e interesses:** listagem baseada nos requisitos que originaram o caso de uso que mostra o que e quem o caso de uso visa satisfazer;
- **Pré-condições:** declaram o que deve ser verdadeiro antes de iniciar um cenário de caso de uso;
- **Pós-condições:** Declaram o que deve ser verdadeiro quando da bem-sucedida conclusão do caso de uso, passando ou não pelo seu fluxo alternativo;
- **Requisitos especiais:** utilizado caso deseje-se relacionar um requisito não funcional ao caso de uso;

O Quadro 2 apresenta um exemplo de especificação de um caso de uso do sistema.

<b>CDU001</b>	<b>Processar venda</b>
<b>Ator Principal</b>	Cliente.
<b>Interesses e Interessados</b>	Cliente: deseja realizar uma compra.
<b>Cenário Principal</b>	
<ol style="list-style-type: none"> <li>1. O cliente navega pelo catálogo e seleciona itens para comprar;</li> <li>2. O cliente vai para o caixa;</li> <li>3. O cliente preenche o formulário da remessa (endereço de entrega; opção de entrega imediata ou em três dias);</li> <li>4. O sistema apresenta a informação completa do faturamento, incluindo a remessa;</li> <li>5. O cliente preenche a informação de cartão de crédito;</li> <li>6. O sistema autoriza a compra;</li> <li>7. O sistema confirma imediatamente a venda;</li> <li>8. O sistema envia uma confirmação para o cliente por e-mail.</li> </ol>	
<b>Extensões</b>	
<p>3a: Cliente regular</p> <ol style="list-style-type: none"> <li>1. O sistema mostra a informação atual da remessa, a informação de preço e informação de cobrança;</li> <li>2. O cliente pode aceitar ou escrever por cima desses padrões, retornando ao passo 6.</li> </ol> <p>6a: O sistema falha na autorização da compra a crédito</p> <ol style="list-style-type: none"> <li>1. O cliente pode inserir novamente a informação do cartão de crédito ou cancelar.</li> </ol>	
<b>Requisitos especiais</b>	
<ul style="list-style-type: none"> <li>• A venda deve ser finalizada em, no máximo, 2 minutos.</li> </ul>	

**Quadro 2 - Exemplo de cenário principal e extensões**

Fonte: Adaptado de Fowler (2005) e Larman (2004)

## 2.5 ESPECIFICAÇÃO DE REQUISITOS

Este subprocesso objetiva a obtenção do detalhamento necessário de cada requisito, de forma que este possa explicitar suficientemente suas características, comprovando sua real utilidade para o projeto (ENGHOLM JÚNIOR, 2010).

Nesta atividade, a equipe de *software* deve trabalhar diretamente com os interessados, objetivando absorver informações detalhadas sobre o domínio da aplicação, os serviços que o sistema deve oferecer, sua expectativa de desempenho, as restrições de hardware, dentre outros (SOMMERVILLE, 2011).

Durante a especificação de requisitos acontece a descrição dos requisitos de usuário e sistema em um documento de requisitos. Espera-se que esta especificação produza requisitos claros, inequívocos, facilmente compreensíveis, completos e consistentes. Sabe-se que estas características nem sempre são facilmente alcançadas, pois cada interessado no projeto interpreta os requisitos de uma forma diferente, tendendo a voltar seu funcionamento às necessidades de sua área (departamento) dentro da organização, ou não dando o devido valor ao mesmo, por não ser de seu interesse ou por possuir desconhecimento dos detalhes do mesmo (SOMMERVILLE, 2011).

Nesta etapa reúnem-se diversos materiais levantados até o momento, dentre eles: as solicitações dos envolvidos, o documento visão do projeto, o glossário do projeto, as listas de requisitos funcionais e não funcionais e os cenários operacionais. Todos estes documentos devem ter sido extraídos e aceitos nas primeiras fases da elicitação de requisitos e devem, neste momento, ser analisados pelos analistas de sistemas e de requisitos, os quais devem formular uma descrição completa de cada um dos requisitos (ENGHOLM JÚNIOR, 2010).

Os requisitos devem ser especificados de forma que qualquer pessoa consiga interpretá-los, ou seja, neste caso, os interessados – que provavelmente não possuirão conhecimentos técnicos em informática – devem estar aptos a ler e compreender os mesmos, podendo, em seguida, dar um parecer sobre a aceitação ou não dos mesmos. Por este motivo, indica-se fortemente a utilização de uma linguagem natural e comum entre todos os envolvidos no projeto, com a apresentação de tabelas e dados de forma simples e compacta (SOMMERVILLE, 2011).

Sabendo desta necessidade por simplicidade e visão de usuário não técnico na apresentação da especificação dos requisitos, fica, também, clara a necessidade da não especificação de dados técnicos do funcionamento do *software*, como jargões desta área da informática, notações estruturadas ou notações formais (SOMMERVILLE, 2011).

Sommerville (2011) ainda indica, além da linguagem natural comum, outras notações para apresentação dos requisitos ao usuário, como: linguagens naturais estruturadas, linguagens de descrição de projeto, notações gráficas e especificações matemáticas (Tabela 1).

**Tabela 1 - Forma de especificação de requisitos**

<b>Notação</b>	<b>Descrição</b>
Sentenças em linguagem natural	Os requisitos são escritos em frases numeradas em linguagem natural. Cada frase deve expressar um requisito.
Linguagem natural estruturada	Os requisitos são escritos em linguagem natural em um formulário padrão ou template. Cada campo fornece informações sobre um aspecto do requisito.
Linguagem de descrição de projeto	Utiliza um formato parecido com linguagem de programação, porém mais abstrato, definindo um modelo operacional do sistema. É uma abordagem pouco utilizada atualmente.
Notações gráficas	Usado para definição dos requisitos funcionais, suplementados por anotações de texto. Diagramas UML são comumente utilizados.
Especificações matemáticas	Notações baseadas em conceitos matemáticos. Reduzem bastante a ambiguidade de requisitos, porém a maioria dos envolvidos não consegue compreendê-las.

**Fonte: SOMMERVILLE (2010)**

Mesma com a ciência desta necessidade por uma especificação clara dos requisitos para usuários, há casos nos quais não há a possibilidade de livrar-se de uma abordagem mais técnica. Este é o caso quando se especifica requisitos de sistema, nos quais, para atingir-se o necessário alto nível de detalhamento do funcionamento destes, é praticamente impossível eliminar todas as informações de projeto (SOMMERVILLE, 2011).

Sommerville (2011) sugere que, quando houver a utilização da linguagem natural estruturada, seja utilizado um formulário contendo, no mínimo, as seguintes informações para cada requisito:

- Descrição da função ou entidade a ser especificada;
- Uma descrição de suas entradas e de onde elas vieram;
- Uma descrição de suas saídas e para onde elas irão;
- Informações sobre a informação necessária para o processamento ou outras entidades usadas no sistema;
- Uma descrição da ação a ser tomada;
- Se uma abordagem funcional é usada, uma pré-condição define o que deve ser verdade antes que a função seja chamada, e é chamada uma pós-condição, especificando o que é verdade depois da função;
- Uma descrição dos efeitos colaterais da operação (caso haja algum).

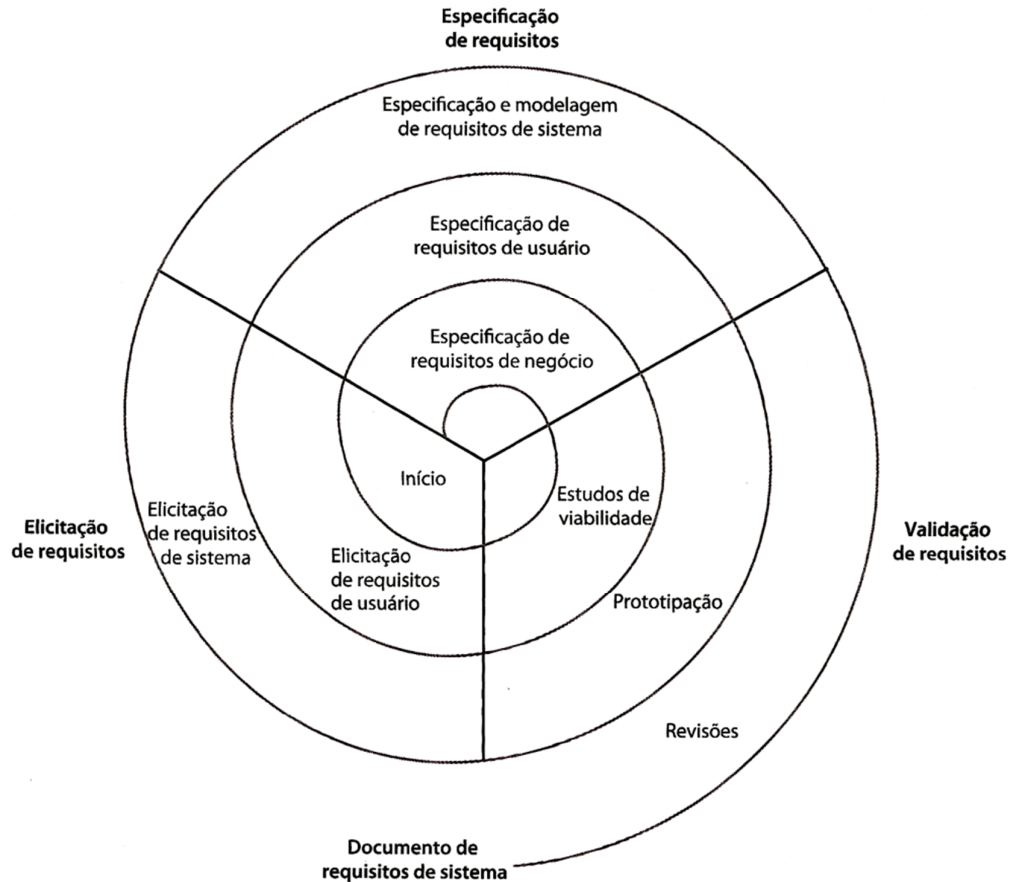


Conhecendo-se estas informações pode-se, segundo Sommerville (2010), elencar como necessários para inserção no formulário de especificação de requisitos, além dos identificadores referentes ao projeto, os seguintes atributos:

- Identificação única do requisito;
- Função;
- Descrição;
- Entradas;
- Fonte das entradas;
- Saídas;
- Destino das saídas;
- Ação;
- Requisitos;
- Pré-condições;
- Pós-condições;
- Efeitos colaterais.

## 2.6 VALIDAÇÃO DE REQUISITOS

A validação de requisitos é o subprocesso através do qual se verifica se os requisitos levantados e especificados realmente definem o sistema que o cliente deseja. Esta etapa ocorre sempre após as etapas de elicitação e especificação de requisitos de cada uma das iterações da engenharia de requisitos (Figura 2). Ela preocupa-se com a identificação de problemas com os requisitos e é importante, pois erros em um documento de requisitos podem gerar altos custos de retrabalho caso descobertos em etapas posteriores, como o desenvolvimento ou até com o sistema em produção. O custo de corrigir um erro na etapa em que se possui somente os requisitos é muitas vezes menor, pois as etapas posteriores são desenvolvidas de acordo com estes, e um erro em algum requisito certamente se propagará à aplicação final (SOMMERVILLE, 2011).



**Figura 2 - Visão espiral do processo de engenharia de requisitos**  
 Fonte: Sommerville (2011)

Na etapa de validação dos requisitos, qualquer artefato produzido pela engenharia de requisitos de determinado sistema deve ser avaliado quanto à qualidade, tendo sua especificação validada para garantir que todos os requisitos tenham sido declarados de maneira não ambígua e que inconsistências, omissões e erros tenham sido detectados e corrigidos, permanecendo somente artefatos autorizados pelos interessados (PRESSMAN, 2011).

A principal forma de validação de requisitos é a revisão técnica, na qual engenheiros de *software*, clientes, usuários, e outros interessados examinam a especificação em busca de erros em seu conteúdo ou interpretação, necessidades por maiores esclarecimentos, informações faltantes, inconsistências, requisitos conflitantes ou inatingíveis (PRESSMAN, 2011).

Nesta etapa, que também poder ser chamada de aprovação de requisitos, há a formalização do entendimento dos requisitos por parte de todos os envolvidos no projeto, sendo fundamental para a continuidade dos trabalhos do mesmo. Cabe à

equipe de *software* (junto aos usuários) validações de características técnicas, como riscos, restrições e prioridades, enquanto fica a cargo dos clientes (junto à equipe de *software*) a verificação do atendimento das suas necessidades por parte dos requisitos especificados (ENGHOLM JÚNIOR, 2010).

Segundo Sommerville (2011), “durante o processo de validação dos requisitos, diferentes tipo de verificação devem ser efetuados com os requisitos”, dentre elas estão:

- **Validade:** inicialmente os interessados podem ter uma visão sobre determinados requisitos e, após o processo de engenharia de requisitos, podem perceber que esta visão era equivocada, ou ela foi incorretamente interpretada pela equipe de *software*, o que acaba por gerar um requisito inválido, que deve ser revisto;
- **Consistência:** os diversos requisitos do documento de requisitos não podem entrar em conflito, ou seja, não pode haver restrições contraditórias entre requisitos ou descrições diferentes para um mesmo requisito;
- **Completezude:** o documento de requisitos do sistema deve conter requisitos que especifiquem todas as funções e restrições almejadas pelos usuários do sistema;
- **Realismo:** os requisitos devem possuir a capacidade de ser implementados, tendo como base as questões de orçamento e financeira do projeto do *software*;
- **Verificabilidade:** deve ser possível estabelecer um conjunto de testes para cada requisito, que possibilitem a verificação “automatizada” deste, a fim de saber se o conjunto de requisitos do sistema atende ao desejado.

Algumas técnicas de validação de requisitos podem ser utilizadas em conjunto. São elas, de acordo com Sommerville (2011):

- **Revisões de requisitos:** os requisitos são analisados sistematicamente por uma equipe, a qual procura por erros e inconsistências;

- **Prototipação:** um modelo executável do *software* pode ser criado, com o intuito de ilustrar de forma prática o funcionamento de determinados requisitos aos interessados;
- **Geração de casos de teste:** os requisitos devem ser testáveis através de verificações automatizadas que, normalmente, consegue revelar novos problemas. Quando um texto automatizado é difícil ou impossível de ser projetado, significa que seu requisito provavelmente é difícil de ser implementado e deve ser reconsiderado.

A equipe de *software* pode, opcionalmente, preparar uma apresentação didática para expor os requisitos à comunidade de interessados do projeto. Porém a leitura e estudo da documentação completa são imprescindíveis para o completo entendimento dos mesmos. Um instrumento que pode ajudar a ilustrar e facilitar a aceitação dos requisitos é a prototipação na qual um esqueleto funcional da aplicação é desenvolvido para fins demonstrativos (ENGHOLM JÚNIOR, 2010).

Sabe-se que é bastante complexa a tarefa de mostrar que determinado requisito atende, de fato, às necessidades de um ou mais usuários, pois este precisa conseguir imaginar o sistema em operação e se encaixando em sua forma de trabalho para conseguir emitir uma opinião sobre a validade ou não do requisito. Apesar de a tarefa de validação de requisitos fazer o possível para encontrar os problemas restantes nos requisitos do sistema, são raras as exceções nas quais são encontrados todos estes problemas, porém quanto menor a quantidade de equívocos restantes após o processo de engenharia de requisitos, menor será o retrabalho requerido nas etapas posteriores da engenharia de *software* (SOMMERVILLE, 2011).

É extremamente importante que a aceitação dos requisitos por parte dos interessados seja formalizada através de um documento de homologação assinado por todos (contrato), pois somente assim previne-se a ocorrência de problemas futuros relacionados à não conformidade e funcionamento de funções do novo sistema (ENGHOLM JÚNIOR, 2010).

### 3 MODELAGEM DE REQUISITOS

Modelagem de sistemas é o processo de criação de modelos abstratos de sistemas, no qual cada um destes modelos apresenta uma visão ou perspectiva diferente do sistema. Normalmente é utilizado algum tipo de notação gráfica que facilita a compreensão. Atualmente, a notação gráfica que é normalmente utilizada é a UML (SOMMERVILLE, 2011).

No processo de engenharia de requisitos, os modelos são utilizados para ajudar na extração e explicação dos requisitos dos interessados, mas também podem ser utilizados durante o processo de projeto, para descrever o sistema para os engenheiros que o implementarão, bem como, posteriormente, têm a função de servir como documentação da estrutura e operação do sistema (SOMMERVILLE, 2011).

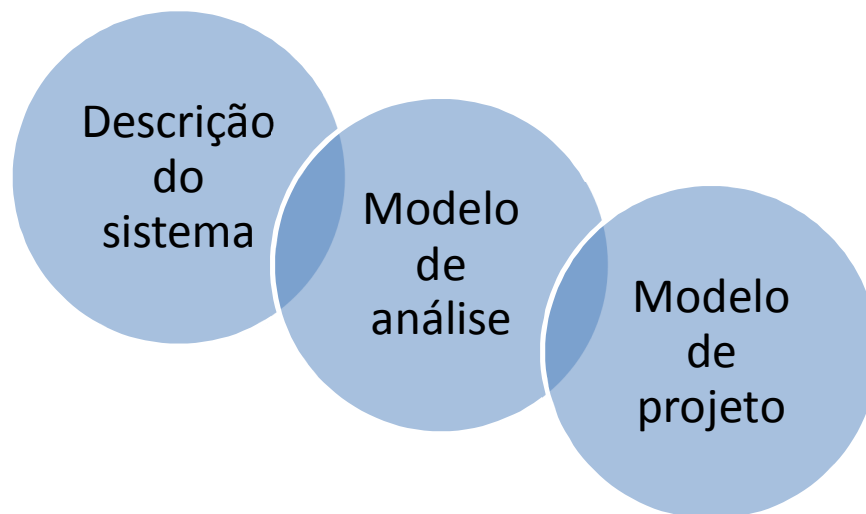
Segundo Sommerville (2011), “o aspecto mais importante de um modelo é que ele deixa de fora os detalhes”, pois “um modelo é uma abstração do sistema a ser estudado, e não uma representação alternativa dele”.

O ato de modelar requisitos resulta em um ou mais tipos de modelos, que segundo Pressman (2011), são:

- Modelos baseados em cenários de requisitos do ponto de vista de vários atores do sistema;
- Modelos de dados que representam o domínio de informações para o problema;
- Modelos orientados a classes que representam classes orientadas a objetos (atributos e operações) e a maneira por meio da qual as classes colaboram para atender aos requisitos do sistema;
- Modelos orientados a fluxos que representam os elementos funcionais do sistema e como eles transformam os dados à medida que percorrem o sistema;
- Modelos comportamentais que representam como o *software* se comporta em consequência de eventos externos.

Durante modelagem de requisitos, assim como na engenharia de requisitos em geral, o foco na necessidade da realização de determinadas ações é maior do que na forma de realização das mesmas. O modelo de requisitos, ou modelo de

análise dos requisitos, assim como a descrição textual dos requisitos, não supre sozinho todas as necessidades de especificação de requisitos funcionais, e sim, atua como um elemento para preenchimento da “lacuna” existente entre a descrição textual dos requisitos do sistema e o modelo do projeto do sistema (Figura 3) (PRESSMAN, 2011).



**Figura 3 - Relação do modelo de análise com outras etapas**  
Fonte: Pressman (2011)

Segundo Pressman (2011), “o modelo de requisitos deve alcançar três objetivos primários”, são eles:

- Descrever o que o cliente solicita;
- Estabelecer uma base para a criação de um projeto de *software*;
- Definir um conjunto de requisitos que possa ser validado assim que o *software* for construído.

### 3.1 UML

A Linguagem de Modelagem Unificada (*Unified Modeling Language* - UML) é uma linguagem visual para especificar, construir e documentar os artefatos de

sistemas. É uma linguagem de propósitos gerais que pode ser aplicada para qualquer tipo de domínio de aplicação e plataforma de implementação (OMG, 2011).

A UML surgiu em 1994, com o intuito de ser a notação diagramática padrão para a realização da modelagem orientada a objetos, porém somente em 1997 foi adotada pelo OMG (*Object Management Group*), passando, a partir de então, a ter atualizações constantes, sendo continuamente refinada através da adição e melhoria de recursos (LARMAN, 2004).

O OMG é um consórcio aberto de empresas que foi formado visando estabelecer padrões que suportassem interoperabilidade para o desenvolvimento objeto-orientado. O consórcio também é conhecido por ter definido o padrão CORBA (*Common Object Request Broker Architecture*) (FOWLER, 2005).

Segundo Fowler (2005), UML “é uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de *software*, particularmente naqueles construídos utilizando o estilo orientado a objetos”.

A versão atual da UML é a 2.4.1, que teve sua especificação liberada no dia 5 de agosto de 2011 e pode ser encontrada no site oficial das especificações da OMG (<http://www.omg.org/spec>) (OMG, 2011).

UML é uma notação bastante vasta, que possui diagramas e padrões para uma grande quantidade de casos da engenharia de *software* (além de estar continuamente crescendo) (LARMAN, 2004). Portanto, quando se fala em modelagem de requisitos faz-se necessária a escolha de alguns diagramas básicos que consigam representar corretamente a visão abstrata do sistema exigida pelos requisitos (SOMMERVILLE, 2011).

Fowler (2005) aponta que a UML possui 13 tipos de diagramas (Tabela 2), agrupados por determinados propósitos de uso (Figura 4), mas faz uma ressalva, comentando que os padrões UML não são rígidos ou fechados, permitindo a utilização dos diagramas para fins alternativos.

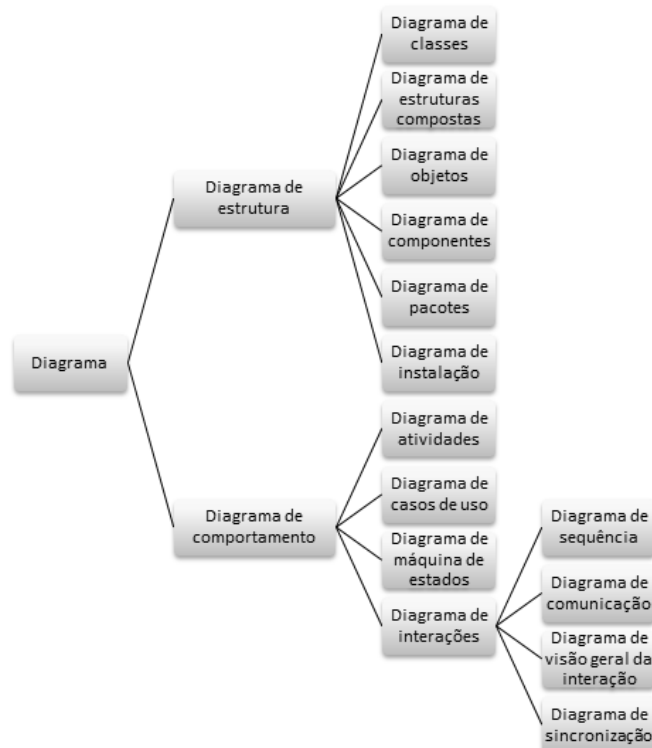
**Tabela 2 - Tipos de diagramas oficiais da UML**

<b>Diagrama</b>	<b>Objetivo</b>
Atividades	Comportamento procedimental e paralelo.
Classes	Classe, características e relacionamentos.

**Tabela 2 - Tipos de diagramas oficiais da UML**

Diagrama	Objetivo
Comunicação	Interação entre objetos, com ênfase nas ligações.
Componentes	Estrutura e conexão de componentes.
Estruturas compostas	Decomposição de uma classe em tempo de execução.
Distribuição ( <i>Deployment</i> )	Distribuição de artefatos de nós.
Visão geral da interação	Mistura de diagrama de sequência e de atividades.
Objetos	Exemplo de configurações de instâncias.
Pacotes	Estrutura hierárquica em tempo de compilação.
Sequência	Interação entre objetos, com ênfase na sequência.
Maquinas de estado	Como os eventos alteram um objeto no decorrer de sua vida.
Sincronismo ( <i>Timing</i> )	Interação entre objetos, com ênfase no sincronismo.
Casos de uso	Como os usuários interagem com um sistema.

Fonte: Fowler (2005)

**Figura 4 - Classificação dos tipos de diagrama da UML**

Fonte: Fowler (2005)



Quando se trabalha com análise de requisitos, é importante ter consciência de que o mais importante é a comunicação com os interessados do projeto, que, normalmente não são profissionais de informática, nem possuem conhecimentos técnicos avançados sobre mesma. Portanto, a UML deve ser utilizada neste sentido, sendo aproveitado apenas o mínimo necessário para atingir esta comunicação. Durante a modelagem de requisitos é preciso estar preparado para a quebra de regras da UML a qualquer momento, caso isto otimize a comunicação. Pois um diagrama que acaba por não ser entendido pelas pessoas que deveriam entendê-lo torna-se inútil (FOWLER, 2005).

Para a atividade de análise de requisitos, Fowler (2005) indica a utilização de três diagramas. São eles:

- **Diagrama de casos de uso:** para descrever como as pessoas interagem com o sistema;
- **Diagrama de classes (com perspectiva conceitual):** para conseguir construir um vocabulário rigoroso do domínio da aplicação;
- **Diagrama de atividades:** para poder exibir o fluxo de trabalho da organização, mostrando como o *software* e as atividades humanas interagem, podendo mostrar o contexto dos casos de uso e também os detalhes sobre como um caso de uso complicado funciona;

Outro diagrama, não citado, que também pode ser aproveitado na modelagem de requisitos é o diagrama de sequência.

### 3.1.1 Diagrama de Casos de Uso

Como um caso de uso define uma visão do sistema de acordo com a interação com agentes internos, é este fato que o diagrama de casos de uso da UML tenta expressar através de notação gráfica (PRESSMAN, 2011).

Ele não representa de maneira completa o funcionamento do caso de uso, que só pode ser trabalhada sob uma perspectiva textual. Um diagrama de caso de uso é uma visão geral de todos os casos de uso de determinado sistema e como estes se relacionam, fornecendo uma visão geral das funcionalidades do sistema (PRESSMAN, 2011).

No diagrama de casos de uso do sistema, atores são representados por símbolos que lembram pessoas (Figura 5).



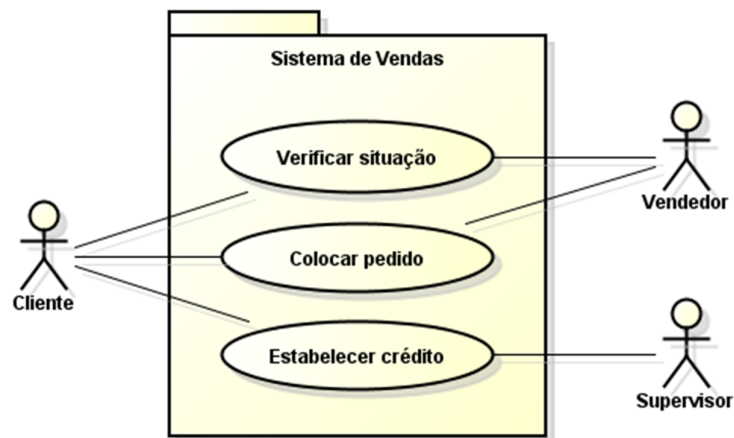
**Figura 5 – Representação dos atores em diagramas de casos de uso**  
Fonte: Lima (2007)

Os casos de uso são representados por elipses contendo o nome dos mesmos (Figura 6).



**Figura 6 – Representação de casos de uso em diagrama de casos de uso**  
Fonte: Lima (2007)

Linhas são utilizadas para “ligar” os atores aos casos de uso, representando assim um relacionamento entre eles e um retângulo grande encobrindo os casos de uso pode ser opcionalmente utilizado para demonstrar a fronteira do sistema (Figura 7) (LIMA, 2007).



**Figura 7 - Diagrama de caso de uso**  
**Fonte: Lima (2007)**

### 3.1.2 Diagrama de Atividades

O diagrama de atividades tem por objetivo mostrar o comportamento dinâmico de um sistema ou parte deste, através da utilização do fluxo de controle entre ações que o sistema executa, sendo similar a um fluxograma, com a diferença de que o diagrama de atividades tem a capacidade de mostrar fluxos concorrentes (PRESSMAN, 2011).

Um diagrama de atividades é um dos diagramas que compõem a visão dinâmica da UML e permite que o comportamento do sistema seja modelado, identificando os caminhos lógicos que determinado processo pode seguir, permitindo a visualização das pré e pós-condições de cada evento interno a este processo (LIMA, 2007).

Este diagrama especifica a coordenação de execuções de comportamentos, fazendo utilização de fluxo de controle e de dados. O fluxo de execução é modelado como nós de atividades, quem são conectados por extremidades. Cada um destes nós pode ser a execução de um comportamento subordinado, como, por exemplo, um cálculo, uma chamada a uma operação específica ou até a manipulação de conteúdos de objetos. Os nós de atividade também incluem construções de fluxo de controle, tais como: sincronização, decisão e controle de execuções paralelas. As atividades podem invocar hierarquias de outras atividades (LIMA, 2007).

O principal componente de um diagrama de atividades é o nó “ação”, que é representado por um retângulo com os cantos arredondados e o nome da ação em seu interior e corresponde a uma tarefa executada por determinado sistema (Figura 8). Setas que vão de um nó ação para outro indicam o fluxo de controle, ou seja, uma seta entre dois nós indica que depois que a primeira ação termina, a segunda se inicia. Um ponto preto sólido forma o nó inicial e um ponto preto sólido circunscrito indica o nó do diagrama (Figura 9).



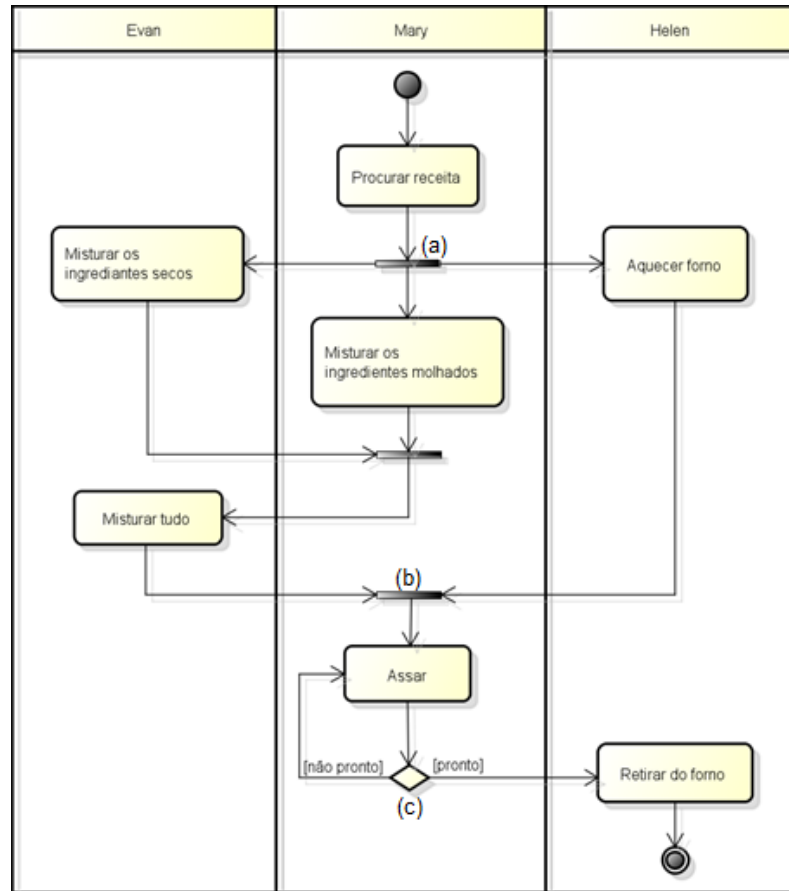
**Figura 8 - Ações em um diagrama de atividades**  
**Fonte: Lima (2007)**



**Figura 9 – Nós indicativos de início (a) e fim (b) de diagrama de atividades**  
**Fonte: Pressman (2011)**

O diagrama pode conter, ainda, os elementos “separação” (Figura 10 - a) e “junção” (Figura 10 - b), que são representados por barras sólidas e tem a função de denotar, respectivamente, o início e o fim de um processamento paralelo, onde várias ações estão executando concorrentemente. Há também a possibilidade da utilização de um nó de decisão (Figura 10 - c), que corresponde a uma ramificação no fluxo de controle baseada em uma condição, que é representado por um losango que recebe uma seta como entrada e pode ter duas ou mais como saída, devidamente identificadas. Outro item que pode, opcionalmente, ser utilizado é o recurso de raias, onde são definidos os executores de cada uma das ações do diagrama. As raias são representadas por grandes retângulos nomeados, com o nome ou papel do executor, possuindo ações em seus interiores (PRESSMAN, 2011).

Um exemplo de diagrama de atividades pode ser observado na Figura 10.



**Figura 10 - Diagrama de atividades**  
 Fonte: Pressman (2011)

### 3.1.3 Diagrama de Classes

O diagrama de classes é utilizado para modelar as classes de um sistema, incluindo seus atributos, operações e relações e associações que umas têm com as outras. Ele fornece uma visão estática, ou estrutural, do sistema, não mostrando a natureza dinâmica das comunicações entre objetos de classes do diagrama (PRESSMAN, 2011).

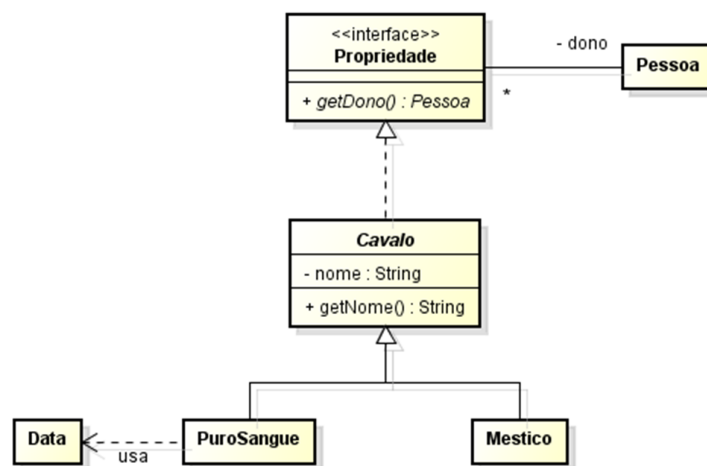
Os elementos principais são caixas, que representam classes e interfaces, onde cada caixa é dividida em partes horizontais, as quais possuem motivos específicos de existência: a parte superior apresenta o nome da classe, a do meio lista os atributos da classe e a terceira contém as operações da classe (PRESSMAN, 2011).

Atributos são informações que a classe que os contém conhece ou pode fornecer o tempo todo, sendo normalmente implementados como campos de classes, porém não precisam ser, podendo ser valores que a classe calcula a partir de suas variáveis de instância ou valores que a classe consegue adquirir de outros objetos dos quais esta é composta. Já as operações tratam de ações que uma classe pode executar, sendo normalmente implementadas no formato de métodos (PRESSMAN, 2011).

Em complemento à listagem dos atributos e operações, pode-se utilizar uma indicação da visibilidade dos mesmos, podendo ser utilizados os sinais -, #, ~ e +, que indicam, respectivamente, as visibilidades do tipo privada, protegida, pacote e pública. Também pode-se, opcionalmente, especificar o tipo dos atributos, bem como o tipo de retorno das operações (PRESSMAN, 2011).

Uma interface pode ser representada por um elemento semelhante ao da classe, porém, neste caso, ela deve fazer a utilização do estereótipo “<<interface>>”, enquanto uma classe abstrata pode ser apresentada com seu nome colocado em itálico. O diagrama de classes da UML possui uma notação bastante extensa para a apresentação dos relacionamentos entre classe, sendo estas de qualquer tipo: herança, associação, agregação, dentre outros. A multiplicidade de relacionamentos também pode ser expressa por elementos predefinidos para isto (PRESSMAN, 2011).

Um exemplo de utilização destas características pode ser visualizado no diagrama de classes exemplificado na Figura 11.



**Figura 11 - Diagrama de classes**  
Fonte: Pressman (2011)

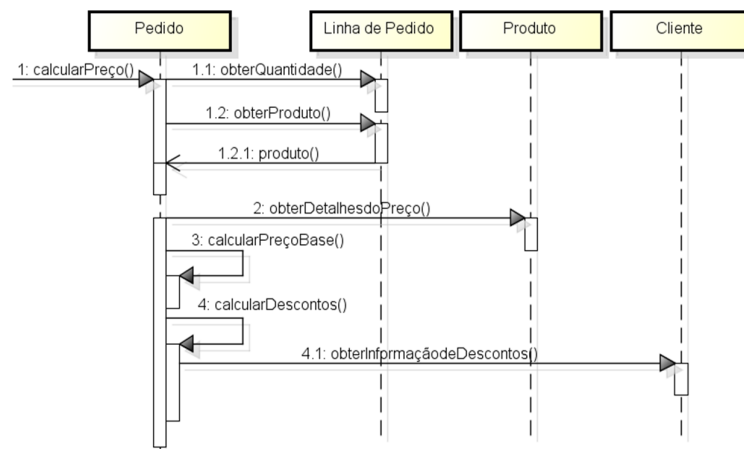
### 3.1.4 Diagrama de Sequência

Diferentemente de outros diagramas, que mostram a estrutura estática do sistema, o diagrama de sequência é utilizado para indicar as comunicações dinâmicas entre objetos durante a execução de uma tarefa, mostrando a ordem temporal do envio de mensagens entre objetos com o objetivo de executar cada tarefa. Ele pode ser usado para apresentar as interações em um caso de uso ou cenário de sistema (PRESSMAN, 2011).

O diagrama de sequência apresenta uma disposição de caixas, com uma linha tracejada em anexo. Estas caixas normalmente representam um objeto, mas podem vir a representar outros tipos de itens, como, por exemplo, classes ou atores. A linha anexa é denominada “linha de vida” do objeto. O eixo vertical do diagrama corresponde ao tempo, que aumenta à medida que se desce (PRESSMAN, 2011).

Um diagrama de sequência apresenta as chamadas e retornos (opcional) de métodos entre os objetos envolvidos através a utilização de setas nomeadas que ligam as denominadas barras de ativação que aparecem nas linhas verticais abaixo de seus respectivos objetos e representam os momentos de atividade dos objetos em questão.

A Figura 12 apresenta um exemplo de diagrama de sequência.



**Figura 12 - Diagrama de sequência**  
Fonte: Fowler (2005)

## 4 ARTEFATOS

Um artefato é algo que representa um elemento do mundo físico, podendo possuir propriedades ou operações, ser instanciado ou associado a outros artefatos. Na engenharia de *software*, entende-se como artefatos de um projeto tudo o que for produzido neste, ou faz parte do sistema de modo geral, como: arquivos de código-fonte, tabelas de dados, documentos de suporte, arquivos de ajuda, dentre outros (LIMA, 2007).

Para Larman (2004), que utiliza uma abordagem voltada a Processo Unificado (*Unified Process* – UP), “um artefato é o termo usado para qualquer produto do trabalho: código, gráficos para a Web, esquemas de bancos de dados, documentos em texto, diagramas, modelos e assim por diante”.

Como resultado do processo de engenharia de requisitos, alguns artefatos são produzidos. Estes artefatos possuem um nível de abstração alto e visam descrever da forma mais fiel que for possível as características acordadas entre os interessados e a equipe de *software* de um projeto, servindo como base para a aplicação das próximas etapas da engenharia de *software* (PRESSMAN, 2011).

### 4.1 PLANO DE GERENCIAMENTO DE REQUISITOS

O plano de gerenciamento de requisitos é um documento que apresenta a estratégia de levantamento, rastreabilidade, tipos de requisitos e sua estrutura, bem como a nomenclatura a sequência que será utilizada durante todo o ciclo de vida do projeto. Tem como finalidade, basicamente, garantir o relacionamento entre os objetos do projeto (ENGHOLM JÚNIOR, 2010).

Este documento começa a ser produzido já nas primeiras etapas da engenharia de requisitos, quando é feito o contato com os primeiros interessados e estabelece-se o escopo do sistema e o domínio a ser tratado, e vai evoluindo juntamente com todo o processo de engenharia de requisitos (ENGHOLM JÚNIOR, 2010).

Conforme afirmado por Engholm Júnior (2010), após a realização de todas as etapas da engenharia de requisitos, o produto gerado para o plano de



gerenciamento de requisitos é um documento que descreve algumas propriedades importantes para os requisitos, bem como para o próprio projeto do sistema, que podem englobar os itens apresentados na Tabela 3.

**Tabela 3 - Propriedades do plano de gerenciamento de requisitos**

<b>Propriedade</b>	<b>Descrição</b>
Finalidade	Explica resumidamente a finalidade do documento em si.
Escopo	Explica brevemente o escopo do plano de gerenciamento de requisitos, bem como, o problema que deseja-se suprir com a construção do sistema.
Referências	Mostra todas as referências relacionadas ao processo de gerenciamento de requisitos.
Visão geral	Apresenta a visão geral sobre os principais tópicos do documento.
Gerenciamento de requisitos	Neste item define-se nominalmente, a matriz de responsabilidades dos envolvidos nas tarefas relacionadas ao levantamento e o gerenciamento de requisitos, indicando o papel de cada um.
Ferramentas, ambiente e infraestrutura	Neste item, devem ser apresentadas as ferramentas, o ambiente e a infraestrutura que serão utilizados no processo de gerenciamento de requisitos. Por exemplo: a equipe deseja utilizar as ferramentas RequisitePro da IBM e o pacote Office da Microsoft.
Identificação dos requisitos	Neste item deve ser registrado o padrão de siglas utilizadas na identificação dos requisitos do projeto.
Tipos de requisitos e artefatos	Apresenta a identificação dos artefatos que serão gerados no processo de gerenciamento de requisitos, definindo um nome e sigla para cada um deles.
Fornecedores de requisitos	Mostra o levantamento de todos os fornecedores (e tipos de fornecedores) de requisitos conhecidos, devendo cada um deles possuir: nome, papel na empresa e projeto, responsabilidades no projeto e demais comentários pertinentes relacionados à disponibilidade da fonte de requisitos.
Estratégias de levantamento e validação de requisitos	Define quais formas de elicitação de requisitos serão utilizadas perante os fornecedores (reuniões, entrevistas, workshops, técnica Delphi, dentre outros). Este item pode ser agrupado ao anterior, de forma que para cada fornecedor ou grupo seja utilizada uma técnica.
Estratégia de gerenciamento de requisitos	Define como os requisitos serão rastreados.
Atributos dos requisitos	Este item apresenta os atributos de requisitos que serão utilizados no projeto, classificadas em: benefício, estabilidade, situação, risco, dentre outros atributos que forem julgados necessários.
Atividades previstas	Caso a realização de determinada tarefa seja esperada, esta deve ser descrita neste item.

**Fonte: Engholm Júnior (2010)**

#### 4.1.1 Gerenciamento de Requisitos

A propriedade de gerenciamento de requisitos envolve a identificação e classificação dos profissionais da equipe de *software* junto às atividades a serem desenvolvidas. Este é um trabalho que cabe ao gerente do projeto. Um exemplo de definição de responsabilidades, com exemplos de subprocesso e atividades que podem ser realmente aplicados a qualquer projeto, pode ser observado na Tabela 4 (ENGHOLM JÚNIOR, 2010).

**Tabela 4 - Responsabilidades da equipe de *software***

<b>Subprocesso / Atividade</b>	<b>Responsável</b>	<b>Papel Executado</b>
Planejar gerenciamento de requisitos	Nome do analista	Analista de Requisitos
Manter integridade dos requisitos	Equipe de projeto	Analista de Requisitos
Gerenciamento de configuração	Nome do analista	Analista de Requisitos
Identificar fornecedores de requisitos	Nome do gerente	Gerente de Projeto
Analisar problema	Equipe de projeto	Analista de Requisitos
Identificar necessidades	Equipe de projeto	Analista de Requisitos
Levantar características do produto	Equipe de projeto	Analista de Requisitos
Definir o sistema	Equipe de projeto	Analista de Requisitos
Validar escopo do sistema	Equipe do cliente	Patrocinador do Projeto e Analista de Processos
Especificar requisitos funcionais e não funcionais	Equipe do projeto e arquiteto de <i>software</i>	Analista de Requisitos
Especificar cenários operacionais	Equipe do projeto	Analista de Requisitos
Criar protótipos	Equipe do projeto	Analista de Sistemas e Designer
Refinar requisitos	Equipe do projeto	Analista de Requisitos
Validar requisitos com fornecedores de requisitos	Equipe do projeto	Analista de Requisitos
Aprovar requisitos	Equipe do cliente e equipe do	Patrocinador do Projeto e

<b>Subprocesso / Atividade</b>	<b>Responsável</b>	<b>Papel Executado</b>
	projeto	Analista de Processos
Manter rastreabilidade	Equipe do projeto	Analista de Requisitos
Estabelecer ambiente de requisitos	Equipe do projeto	Analista de Requisitos

**Fonte: Engholm Júnior (2010)**

#### 4.1.2 Identificação dos Requisitos

A propriedade de identificação dos requisitos pode utilizar o padrão conhecido como TIPSEQ, no qual estabelecem-se duas variáveis (TIP e SEQ), onde TIP é uma abreviação de três dígitos que identifica o tipo do requisito (Tabela 5) e SEQ é um número sequencial cuja contagem reinicia-se a cada novo tipo de requisito, porém esta nomenclatura fica a cargo da equipe de *software*, cabendo a esta adaptar esta técnica de forma que melhor atenda aos objetivos do projeto em questão (ENGHOLM JÚNIOR, 2010).

**Tabela 5 - Tipos de requisitos**

<b>Tipo de Requisito</b>	<b>Abreviação</b>
Necessidade do Interessado	NEC
Característica do Produto	CAR
Requisito de Interface	REI
Requisito Funcional	REF
Requisito Não Funcional	RNF
Evento	EVT

**Fonte: Engholm Júnior (2010)**

### 4.1.3 Fornecedores de Requisitos

Os critérios utilizados para a classificação dos fornecedores de requisitos podem seguir os apresentados na Tabela 6, podendo ser adicionados outros critérios, devendo, inclusive, ser listados anteriormente à efetiva listagem de fornecedores de requisitos.

**Tabela 6 - Critérios para classificação de fornecedores de requisitos**

<b>Categoria / Papel</b>	<b>Tipo (<sup>1</sup>)</b>	<b>Nível de Decisão / Influência</b>	<b>Identificação / Descrição</b>
Patrocinador do Projeto	CPA	Alto	Possui decisão final sobre questionamentos e sugestões em relação ao sistema.
Gerente do projeto	CPA	Alto	Responsável pelo gerenciamento do projeto.
Gestor / Usuário	UFI	Médio	Possui decisão final relacionada às funcionalidades do sistema
Analista técnico / processos	ATC	Alto	Decide sobre questões técnicas e funcionais esclarecidas pelos usuários.

<sup>1</sup> CPA = Cliente patrocinador / UFI = Usuário final / ATC = Área técnica do cliente

**Fonte: Engholm Júnior (2010)**

## 4.2 DOCUMENTO VISÃO

O documento visão é um artefato, assim como o plano de gerenciamento de requisitos, que é produto das primeiras etapas da engenharia de requisitos, surgindo a partir da elicitação de requisitos em alto nível (ENGHOLM JÚNIOR, 2010).

Este documento tem por objetivo apresentar a visão inicial do projeto, através da apresentação das primeiras necessidades e problemas a serem resolvidos. Também faz uma introdução à empresa e seu ramo de negócio, demonstrando o domínio da aplicação, os usuários e tenta descrever como o sistema vai se encaixar nestas características, bem como também traz o apontamento das restrições do projeto (econômicas e de cronograma), dentre outras informações (ENGHOLM JÚNIOR, 2010). A Tabela 7 apresenta uma possível estrutura para apresentação deste documento, segundo Engholm Júnior (2010).

**Tabela 7 - Estrutura do documento de visão**

<b>Tópico</b>	<b>Descrição</b>
Introdução	Apresenta o documento, devendo fornecer uma visão geral deste, incluindo seu propósito, escopo, definições e referências.
Oportunidade de negócio	Descreve resumidamente a oportunidade de negócio que se deseja alcançar através do projeto. Deve-se tentar descrever um nicho de mercado.
Problema	Deve ser fornecida uma declaração que sumarie os problemas que serão resolvidos pelo projeto, podendo ser utilizada uma visão estruturada dos mesmos.
Necessidades dos interessados	Apresenta as necessidades de usuário que deverão ser atendidas pelo projeto, especificando suas prioridades, preocupações e possíveis soluções. Este item pode ser agrupado sob uma visão estruturada ou manter características totalmente textuais.
Resumo dos usuários	Descreve o perfil dos usuários que utilizarão o sistema e as funcionalidades que cada um poderá ter acesso no mesmo.
Ambiente do sistema	É responsável por apresentar o ambiente no qual o sistema executará e suas restrições de arquitetura.
Visão geral do projeto	Sessão que deve descrever as características gerais do produto a ser desenvolvido. Pode conter uma apresentação geral, as perspectivas, um resumo das capacidades do sistema e uma enumeração das principais características do produto.
Restrições econômicas	Restrições financeiras do projeto e da etapa de engenharia de requisitos.
Restrições de prazo	Restrições de cronograma do projeto e da etapa de engenharia de requisitos.
Pendências e prioridades	Relaciona, de forma sequencial, as principais prioridades que devem ser atendidas pelo sistema.
Requisitos de documentação	Descreve a documentação que deve ser desenvolvida para a efetiva implantação do projeto.

**Fonte: Engholm Júnior (2010)**

#### 4.2.1 Problema

O tópico referente aos problemas pode ser apresentado, de acordo com Engholm Júnior (2010), contendo um formulário estruturado (Quadro 3) para cada problema identificado.

<b>O problema de</b>	[ Especificar de forma textual qual o problema ]
<b>Afeta</b>	[ Definir quais são os setores ou características afetados da empresa ]
<b>Cujo impacto é</b>	[ Explicar qual é o impacto exercido sobre o item anterior ]
<b>Benefícios de uma solução seriam</b>	[ Enumerar todos os benefícios conhecidos e cabíveis da solução deste problema ]

**Quadro 3 - Modelo para apresentação dos problemas do projeto**

Fonte: Engholm Júnior (2010)

#### 4.2.2 Resumo dos Usuários

Este tópico demonstra os possíveis tipos de usuário do sistema, descrevendo mais detalhes e responsabilidades de cada um. De acordo com Engholm Júnior (2010), pode ser utilizado como base o formulário apresentado no Quadro 4.

<b>Nome</b>	<b>Descrição</b>	<b>Responsabilidades</b>
Visitante	[ Descreve, de forma textual, qual o perfil do usuário do tipo visitante ]	[ Enumerar as responsabilidades de um visitante ]
Cliente	[ Descreve, de forma textual, qual o perfil do usuário do tipo cliente ]	[ Enumerar as responsabilidades de um cliente ]
Administrador	[ Descreve, de forma textual, qual o perfil do usuário do tipo administrador ]	[ Enumerar as responsabilidades de um administrador ]
Outros	[ Podem ser incluídos outros tipos de usuários que forem julgados necessários para o sistema ]	[ Cada novo tipo de usuário deverá possuir suas respectivas responsabilidades ]

**Quadro 4 - Estrutura de apresentação do resumo dos usuários**

Fonte: Engholm Júnior (2010)

#### 4.3 GLOSSÁRIO

A criação do glossário é um ponto extremamente importante do projeto. No glossário são definidos termos técnicos e siglas que podem não ser conhecidas por

algum dos lados do levantamento de requisitos: equipe de *software* ou fornecedores de requisitos. Seu objetivo é, basicamente, estabelecer um vocabulário comum entre estas duas equipes (ENGHOLM JÚNIOR, 2010).

O glossário é um documento simples, devendo conter, como nos outros documentos conhecidos, uma introdução e definições de propósito e escopo devidamente documentados, bem como um item para as definições de glossário, contendo os termos e suas explicações, que é seu motivo de existência. Durante a apresentação das definições é aconselhável assumir alguma forma de ordenação, como a alfabética, para facilitar o entendimento e a localização de termos. A existência deste documento deve ficar clara para todos os envolvidos no projeto (ENGHOLM JÚNIOR, 2010).

#### 4.4 LISTA DE RISCOS

A lista de riscos é um documento simples que contém uma enumeração de formulários que apresentam os detalhes dos riscos do projeto. A tarefa de identificação dos riscos é realizada desde o começo do projeto, devendo ser efetuada periodicamente durante toda sua execução (ENGHOLM JÚNIOR, 2010).

A lista pode apresentar tanto os riscos únicos para o projeto em questão, quanto os riscos genéricos, que podem afetar qualquer projeto de *software*, podendo cada um dos riscos ser apresentado no modelo sugerido no Quadro 5 (ENGHOLM JÚNIOR, 2010).

<b>Descrição</b>	[ Descrição do risco ]
<b>Categoria</b>	[ Categoria / área do risco ]
<b>Fatores</b>	[ Fatores que podem gerar o problema ]
<b>Descrição do impacto</b>	[ Descrição do impacto gerado pelo problema ]
<b>Probabilidade</b>	[ Probabilidade de acontecimento do problema especificado pelo risco (Baixa / Média / Alta) ]
<b>Impacto</b>	[ Apresentação do nível de força do impacto, caso o problema ocorra (Baixo / Médio / Alto) ]
<b>Perda esperada</b>	[ Classificação da perda (Baixa / Média / Alta) ]

<b>Estratégia</b>	[ Forma de tratamento do risco ]
<b>Plano de resposta ao risco</b>	[ Descrição completa e detalhada de como o risco será tratado ]

**Quadro 5 - Apresentação da lista de riscos**  
**Fonte: Engholm Júnior (2010)**

#### 4.5 LISTA DE REQUISITOS

A lista de requisitos é o documento que descreve os requisitos elicitados a partir das informações iniciais obtidas, tendo como base a requisição da área de negócio, a requisição de proposta do cliente e outras fontes. É, então, criada uma listagem preliminar dos requisitos do projeto, a qual apenas descreve as funcionalidades e características do *software* a nível macro, não se preocupando com os detalhes, que serão tratados nas próximas etapas (ENGHOLM JÚNIOR, 2010).

Na fase de especificação de requisitos, esta lista deve ser refinada, retificada e ratificada, porém somente no documento de requisitos do sistema, pois ela serve como base para o detalhamento completo dos requisitos e será utilizada em todos os processos posteriores do projeto, como a análise, desenho, implementação e teste (ENGHOLM JÚNIOR, 2010).

Segundo Engholm Júnior (2010), esta lista deve apenas apresentar o requisito com sua visão macro, conforme os exemplos:

- A aplicação deverá ser web;
- O sistema deverá permitir a conexão simultânea de vários usuários;
- O sistema deve permitir ao administrador administrar o cadastro de miniaturas do sistema;
- O sistema deve permitir ao administrador visualizar dados de cliente selecionado;
- O sistema deve mostrar relação de categorias de produtos cadastradas no sistema.



## 4.6 DOCUMENTO DE REQUISITOS

O documento de requisitos do *software*, também conhecido como Especificação de Requisitos de *Software* (*Software Requirements Specification - SRS*), é a declaração oficial do que os desenvolvedores devem implementar (SOMMERVILLE, 2011).

Para Engholm Júnior (2010), o documento de requisitos do sistema “captura, de uma maneira muito mais detalhada do que o documento de visão do projeto, o escopo, os objetivos de negócio e requisitos do sistema proposto”, porém ainda dando ênfase no que o sistema deve fazer e não se importando como isto será realizado.

Ele deve incluir tanto os requisitos de usuário, quanto uma descrição bastante rica em detalhes dos requisitos de sistema da aplicação que é o motivo da sua existência. Em alguns casos, estes dois tipos de requisitos são integrados em uma mesma descrição, porém, neste caso, os requisitos de usuário são definidos em uma introdução à especificação de requisitos de sistema. Caso haja um número bastante elevado de requisitos, os requisitos de sistema podem ser apresentados em um documento separado (SOMMERVILLE, 2011).

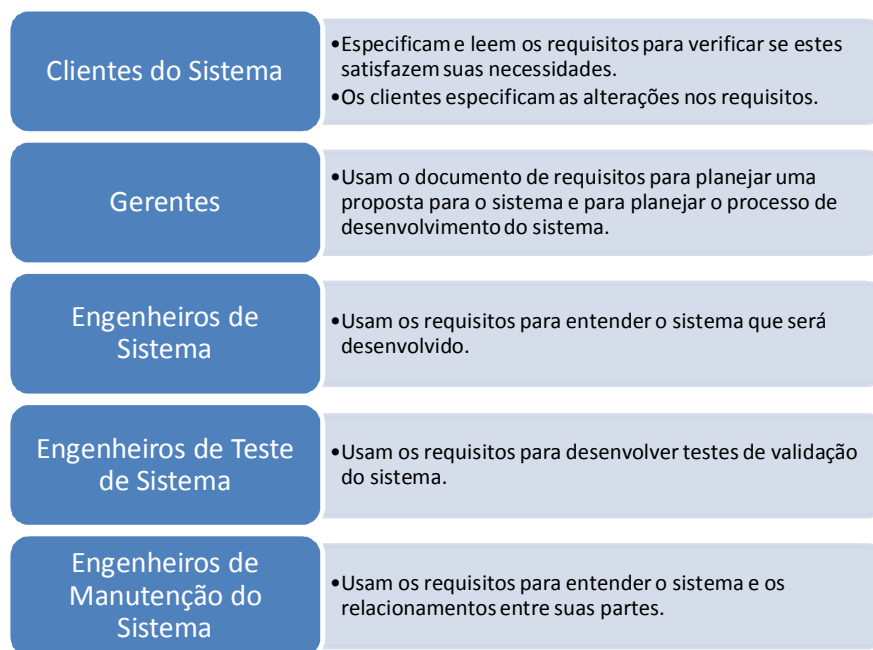
Documentos de requisitos são extremamente necessários principalmente nos casos onde há o desenvolvimento de um *software* por um contratante externo à organização, fato que, atualmente, é bastante comum (SOMMERVILLE, 2011).

Os adeptos de métodos ágeis de desenvolvimento têm pregado a não utilização do documento de requisitos, com o argumento de que os requisitos mudam tão rapidamente que ao final de sua escrita, já está ultrapassado, sendo todo o esforço desperdiçado. Embora isto seja, em partes, verdade, pois não há como evitar uma evolução natural das ideias, a utilização do documento de requisitos continua indicada, pois somente com ele consegue-se ter uma visão geral dos requisitos do sistema como um todo, e não somente uma visão em “tempo real” das próximas características a serem implementadas, como sugerem estes adeptos (SOMMERVILLE, 2011).

Sommerville (2011) ainda afirma que métodos ágeis de desenvolvimento de *software* “são uma boa abordagem para os sistemas de negócio em que os requisitos são instáveis”, porém, ainda assim, indica “escrever um pequeno

documento de apoio no qual estejam definidos os requisitos de negócio e de confiança para o sistema”, pois pode-se acabar esquecendo dos requisitos que afetam o sistema como um todo.

O documento de requisitos de um sistema possui um conjunto de usuários, que englobam, basicamente, os envolvidos no projeto deste. Sabe-se, ainda, que cada uma destas pessoas (ou conjunto delas) possui determinadas responsabilidades sobre o documento e sua utilização (Figura 13) (SOMMERVILLE, 2011).



**Figura 13 - Usuários de um documento de requisitos**  
**Fonte: Sommerville (2011)**

O documento existe justamente para integrar e alinhar as ideias de todos os interessados no projeto já no momento de concepção do mesmo, quando ainda não há muito a perder, caso haja necessidade de alterações. Ele precisa ser um compromisso com a comunicação dos requisitos para os clientes, a definição detalhada e precisa dos requisitos para os desenvolvedores e testadores e a adição de informações sobre a esperada evolução do sistema. Esta perspectiva de evolução do sistema pode conter um tópico alertando sobre as possíveis mudanças que podem ocorrer, o que pode ajudar os projetistas a evitar decisões de projeto

restritivas, além de ajudar os engenheiros no momento da adaptação do sistema à entrada de novos requisitos (SOMMERVILLE, 2011).

O nível de detalhamento de um documento de requisitos depende de alguns fatores, vários definidos pela equipe de *software*, de acordo com a conveniência. Porém, pode-se incluir como padrão dois atributos nesta lista: o tipo do sistema a ser desenvolvido e o processo de desenvolvimento utilizado. Sistemas críticos precisam ter seus requisitos mais cuidadosamente descritos, pois a segurança e a proteção devem ser estudadas em detalhes. Caso o sistema esteja sendo desenvolvido por uma companhia externa à companhia cliente, as especificações dos requisitos do sistema devem ser detalhadas e precisas. Por outro lado, caso seja utilizado um processo iterativo, por exemplo, o documento de requisitos pode ser bem menos detalhado, podendo algumas pendências, como ambiguidades, ser resolvidas no momento do desenvolvimento (SOMMERVILLE, 2011).

Sommerville (2011) expõe um modelo de estrutura para um documento de requisitos (Tabela 8). A estrutura apresentada é um modelo genérico, podendo ser adaptada para usos específicos.

**Tabela 8 - Estrutura de um documento de requisitos**

Capítulo	Descrição
Prefácio	Define os possíveis leitores do documento, bem como descreve o histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das alterações realizadas em cada versão.
Introdução	Descreve a necessidade do sistema, bem como, de forma breve, suas funções e a integração com outros sistemas. Também pode descrever a forma como o sistema pretende atingir aos objetivos da organização.
Glossário	Estabelece os termos técnicos utilizados no documento.
Definição de requisitos de usuário	Descreve os serviços fornecidos ao usuário, bem como os requisitos não funcionais, podendo utilizar linguagem natural, diagramas ou outras notações compreensíveis para os clientes.
Arquitetura do sistema	Apresenta uma visão geral em alto nível da arquitetura do sistema, mostrando a distribuição de funções dentre seus módulos. Componentes de arquiteturas que são reutilizados devem ser destacados.
Especificação de requisitos do sistema	Descreve, em detalhes, os requisitos funcionais e não funcionais, podendo ser adicionados mais detalhes a estes. As interfaces com outros sistemas também podem ser definidas nesta área.
Modelos do sistema	Pode incluir gráficos que mostram os relacionamentos entre os componentes do sistema, o sistema em si e seu ambiente.

---

<b>Capítulo</b>	<b>Descrição</b>
Evolução do sistema	Descreve os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução do hardware, de mudanças de necessidades do usuário, dentre outras.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas ao sistema, além de descrições de hardware (configurações mínimas e recomendadas) e banco de dados (organização lógica dos dados e seus relacionamentos), por exemplo.
Índice	Vários índices podem ser incluídos no documento, podendo haver índices alfabéticos, de diagramas, de funções, dentre outros que forem julgados pertinentes.

---

**Fonte: Sommerville (2011)**

## 5 ESTUDO EXPERIMENTAL

A aplicação deste trabalho teve como objetivo aplicar a engenharia de requisitos sobre um estudo experimental proposto a fim de identificar os requisitos necessários a um sistema para controle de eventos que consiga operar de maneira genérica, ou seja, independente dos tipos de requisitos esperados por determinado tipo ou área de evento a ser realizado, da Universidade Tecnológica Federal do Paraná (UTFPR) – Câmpus Medianeira, tendo como produto final um documento de requisitos (e documentos auxiliares a este) que contenham a especificação do que deve ser projetado e implementado nas próximas etapas da engenharia de *software*.

### 5.1 MATERIAIS E MÉTODOS

Este trabalho baseou-se no formato de pesquisa exploratória do tipo “estudo experimental”, tendo seus dados coletados através de levantamento bibliográfico, entrevistas, observação, análise de documentos e de sistemas legados.

Os métodos adotados para coletar os dados têm suas utilizações devidamente embasadas e justificadas nos tópicos referentes à revisão bibliográfica, onde estudou-se a necessidade de aplicação das seguintes técnicas:

- Entrevistas abertas com o universo de pessoas interessadas no desenvolvimento do sistema;
- Observação da utilização de sistemas legados com propósitos equivalentes à proposta de *software* a ser criado;
- Análise de sistemas legados com propósitos equivalentes à proposta de *software* a ser criado;
- Análise de documentos relacionados com a área de aplicação do *software*.

Para a realização das atividades práticas do estudo experimental, bem como para a elaboração do material de documentação (documento de requisitos do sistema de controle de eventos e adendos) houve a necessidade da utilização de alguns materiais. Estes são brevemente descritos a seguir.

- **Microsoft Word 2010:** a ferramenta Microsoft Word 2010 é um editor de textos comercial, distribuído junto ao pacote conhecido como Microsoft Office 2010, que foi utilizado na confecção dos documentos obtidos no estudo experimental;
- **Astah\* Community 6.5:** o Astah\* Community 6.5 é uma ferramenta *open-source* que foi utilizada para o processo de elaboração de diagramas UML para o estudo experimental proposto;
- **Modelos de Documentos Formais:** os modelos de documentos formais, que foram utilizados como base para a obtenção dos documentos de requisitos do estudo experimental (documento de requisitos do sistema (APÊNDICE D) e seus documentos auxiliares: plano de gerenciamento de requisitos (APÊNDICE A), documento de visão (APÊNDICE B) e glossário do projeto (APÊNDICE C)) foram extraídos e livremente adaptados de Engholm Júnior (2010) e Sommerville (2011).

## 5.2 RESULTADOS E DISCUSSÃO

Para iniciar os comentários sobre a forma adotada para se chegar ao proposto documento de requisitos de um sistema para controle de eventos, faz-se necessário lembrar do conteúdo apresentado na Figura 2, proposta por Sommerville (2011). Esta apresenta uma visão espiral da sequência de procedimentos que devem ser adotados para a confecção de um documento de requisitos de qualquer sistema.

Os documentos desenvolvidos tiveram características extraídas das diversas fontes bibliográficas estudadas, possuindo em suas estruturas, uma junção de atributos indicados por diversos autores. As fontes diretas para a montagem da estrutura destes documentos foram Engholm Júnior (2010), Pressman (2011) e Sommerville (2011). Além destes dados disponibilizados pelos autores, houve a necessidade de adaptação em algumas características, para que estes documentos se aproximassem ao máximo da forma de abordagem ideal para o estudo experimental proposto.

As subseções desta seção de resultados e discussão estão organizadas de tal forma que seguem a sequência abordada pela figura acima citada. Sequência esta, que foi utilizada como linha-guia para a produção do documento de requisitos do sistema proposto.

### 5.2.1 Início

Na etapa denominada “Início” estabeleceram-se os primeiros contatos com um interessado (também tratado por “interessado inicial”) no projeto tido como chave para a organização geral das necessidades iniciais do mesmo, bem como para a identificação dos demais possíveis interessados e seus setores dentro da UTFPR Câmpus Medianeira. Esta pessoa teve seu nome inserido como chave por possuir grande experiência na coordenação de eventos, tendo organizado e gerenciado diversos eventos na área de informática no próprio Câmpus de Medianeira da UTFPR e em outras localidades.

É importante ressaltar que nesta primeira etapa normalmente estabelece-se qual é a equipe de *software* que fará todo o trabalho de comunicação com os clientes (interessados), bem como o cronograma e tarefas a serem realizadas, porém para este caso específico deve assumir que a equipe de *software* está limitada a somente uma pessoa: o pesquisador.

Através de uma entrevista com este primeiro interessado, houve o esclarecimento geral sobre as principais necessidades do projeto, bem como a identificação dos primeiros requisitos funcionais (em alto nível de abstração) dos mesmos e a identificação de outros dois interessados no projeto: uma servidora da mesma instituição, representante da Assessoria de Cerimonial e Eventos e um servidor, lotado na FUNTEF (Fundação de Apoio à Educação, Pesquisa e Desenvolvimento Científico e Tecnológico da UTFPR), que possuem respectivamente, dentre outras, as responsabilidades de organização e coordenação geral dos diversos eventos ocorridos no câmpus da instituição e de acompanhamento financeiro de questões relacionadas a eventos.

Ainda deste mesmo interessado, foi obtido um documento chamado “projeto de evento”, que especifica diversos detalhes que devem ser abordados no momento da concepção de determinado evento. Este documento é feito para cada evento a

ser realizado na UTFPR Câmpus Medianeira e tem por objetivo fazer com que o coordenador do evento para o qual este documento foi produzido consiga informar a FUNTEF sobre os detalhes organizacionais e, principalmente, financeiros do evento e ser realizado. A análise deste documento foi de grande valia para o levantamento inicial de requisitos, pois vários requisitos funcionais e detalhes importantes destes puderam ser avaliados tendo como base um documento formal, que já vem sendo utilizado em diversos eventos, independentemente da utilização, ou não, de um sistema de informação para a gerência dos mesmos.

Em um passo seguinte, porém ainda nesta mesma etapa da sequência de processos apresentados anteriormente, foram realizadas as entrevistas com os outros dois interessados identificados, dos quais foram extraídas informações relacionadas à como seus respectivos setores interagem e como gostariam de interagir com um sistema que viesse a gerenciar eventos.

Nesta etapa também houve a observação da utilização de dois sistemas já existentes por parte da interessada da Assessoria de Cerimonial e Eventos, onde um destes tem por objetivo realizar a gerência e divulgação de eventos, propósito este semelhante ao sistema proposto no documento de requisitos deste trabalho, e o outro objetiva a geração de crachás identificadores de participantes, certificados e declarações dos eventos cobertos pelo primeiramente citado. Esta observação dos sistemas pode identificar, além de requisitos funcionais importantes, alguns problemas que poderiam vir a ser sanados de maneira bastante eficaz na aplicação proposta por este trabalho, como:

- A elevada necessidade da posse de diversos conhecimentos técnicos específicos da área de informática por parte do usuário destes sistemas;
- A necessidade por uma interface de página pública mais atrativa e facilmente automaticamente adaptável a diferentes eventos;
- A falta de integração entre os dois *softwares* identificados e seus dados, fazendo-se necessária a presença de uma pessoa da área técnica da informática sempre que há a necessidade de realizar a comunicação entre eles;
- A dificuldade de reaproveitamento de uma mesma instalação do primeiro sistema em diferentes eventos, fazendo-se necessária a



presença de uma pessoa da área técnica da informática sempre que há a necessidade de utilização deste em um novo evento;

- A não padronização da gerência dos acontecimentos internos a um evento, sendo cada um, tratado de uma forma completamente específica e aleatória, sendo que poderia haver tal maneira padronizada de tratamento, o que viria a agilizar o processo de desenvolvimento de módulos para novos formatos destes acontecimentos que venham a ocorrer.

O próximo passo foi compreendido pela realização da entrevista com o interessado da FUNTEF, o qual tem interesse apenas nos detalhes financeiros dos eventos realizados, tendo como função, falando-se em eventos, basicamente, a análise do projeto destes, efetivando suas respectivas aprovações (ou reprovações) e liberação (ou não) dos recursos requeridos neste. Outra função associada a este interessado é a área de recebimento dos valores de inscrições de participantes de eventos, de maneira presencial ou por pagamento via boleto bancário, cabendo ao mesmo realizar acesso ao sistema de controle de eventos e confirmar as inscrições, garantindo as vagas, dos participantes que o fizerem.

Esta primeira etapa não tem uma saída formal nem documentada. O material por ela gerado é, basicamente, dados esboçados durante as entrevistas, observação e análises, além do conhecimento adquirido por parte da equipe de *software*, o qual não pode, neste momento, ser expresso de maneira visual, porém deve-se entender que tais dados têm um importante papel para o estabelecimento das etapas subsequentes.

### 5.2.2 Especificação de Requisitos de Negócio

Na etapa de “Especificação de Requisitos de Negócio” foi produzida uma primeira descrição dos primeiros requisitos obtidos, sendo esta ainda de caráter informal, servindo apenas como base para a primeira etapa de validação que é tratada pelo próximo processo.

Nesta etapa, após avaliação por parte da equipe de *software*, já foi possível conhecer a função básica de boa parte dos requisitos funcionais do sistema. Eram eles:

- Manter dados cadastrais de eventos;
- Manter dados cadastrais de subeventos;
- Manter dados cadastrais de áreas;
- Manter dados cadastrais de salas;
- Manter dados cadastrais de palestrantes;
- Manter dados cadastrais de participantes;
- Gerenciar grupos de usuários e permissões;
- Manter dados cadastrais de patrocinadores de eventos;
- Realizar inscrição em evento;
- Gerar boleto de inscrição em evento;
- Escolher participação em subeventos;
- Efetivar confirmação de pagamento;
- Confirmar inscrição;
- Imprimir crachás;
- Lançar presenças;
- Visualizar site de evento;
- Gerenciar tipos de usuários;
- Manter dados cadastrais de locais;
- Manter dados cadastrais de atividades de eventos;
- Montar programação;
- Gerar projeto de evento;
- Manter dados cadastrais de despesas;
- Manter dados cadastrais de receitas;
- Gerar relatório de receitas e despesas de evento;
- Manter dados cadastrais de integrantes da comissão organizadora;
- Gerar certificados e declarações.

Apesar do resultado final desta etapa ser de natureza informal, nela que se deu o início da elaboração dos primeiros documentos que viriam a se tornar formais nas etapas subsequentes. Os documentos que foram concebidos nesta etapa foram (apenas as versões finais são apresentadas nos apêndices):

- Plano de Gerenciamento de Requisitos (APÊNDICE A)
- Documento de Visão (APÊNDICE B)
- Glossário do Projeto (APÊNDICE C)

### 5.2.3 Estudos de Viabilidade

A primeira aplicação da técnica de validação de requisitos ocorreu na etapa de estudos de viabilidade, na qual houve trocas de informações e mensagens de maneira informal, entre a equipe de *software* e o interessado inicial, sendo a base das discussões os documentos e requisitos especificados nas fases anteriores. Estas discussões visaram a aprovação (ou rejeição) dos requisitos propostos, bem como a discussão sobre alterações em características e funcionalidades dos mesmos, o que acabou por gerar novos requisitos (além dos já existentes) para serem tratados nas etapas seguintes.

A troca de informações nesta etapa foi realizada de forma oral, através de encontros presenciais entre a equipe de *software* e o interessado, e também com a utilização de *e-mails*.

Como resultado desta etapa, obteve-se a aprovação da lista de requisitos do sistema por parte dos interessados, sendo esta finalizada com os seguintes requisitos:

- O sistema deve manter os dados de evento
- O sistema deve manter os dados de subevento
- O sistema deve manter os dados de área
- O sistema deve manter os dados de instituição
- O sistema deve manter os dados de sala
- O sistema deve manter os dados de palestrante
- O sistema deve manter os dados de usuário
- O sistema deve manter os dados de patrocinador
- O sistema deve manter os dados de atividade
- O sistema deve manter os dados de receita/despesa de evento
- O sistema deve permitir gerenciar grupos de usuários e permissões
- O sistema deve permitir realizar inscrição em evento
- O sistema deve permitir escolher participação em subevento
- O sistema deve permitir gerar boleto de inscrição em evento
- O sistema deve permitir confirmar inscrição
- O sistema deve permitir realizar aprovação financeira do evento
- O sistema deve permitir imprimir crachás

- O sistema deve permitir lançar presença
- O sistema deve permitir visualizar site de evento
- O sistema deve permitir montar programação
- O sistema deve permitir gerar projeto de evento
- O sistema deve permitir gerar relatório de receitas e despesas de evento
- O sistema deve permitir gerar certificados e declarações

#### 5.2.4 Elicitação de Requisitos de Usuário

A etapa de elicitação de requisitos de usuário tem como produto de entrada as informações e documentos levantados nas etapas anteriores, tendo seu alicerce no estudo dos requisitos elencados, a fim de transformá-los em casos de uso que conseguissem representar os possíveis cenários de uso do sistema, sob a visão de diferentes usuários.

Como a listagem obtida na etapa anterior apresentava uma forte característica de requisitos sob a visão de usuários finais, não houve uma necessidade por grandes adequações destes requisitos para a criação dos casos de uso, sendo que cada um destes requisitos funcionais acabou por gerar um caso de uso. Os casos de uso obtidas nesta etapa foram:

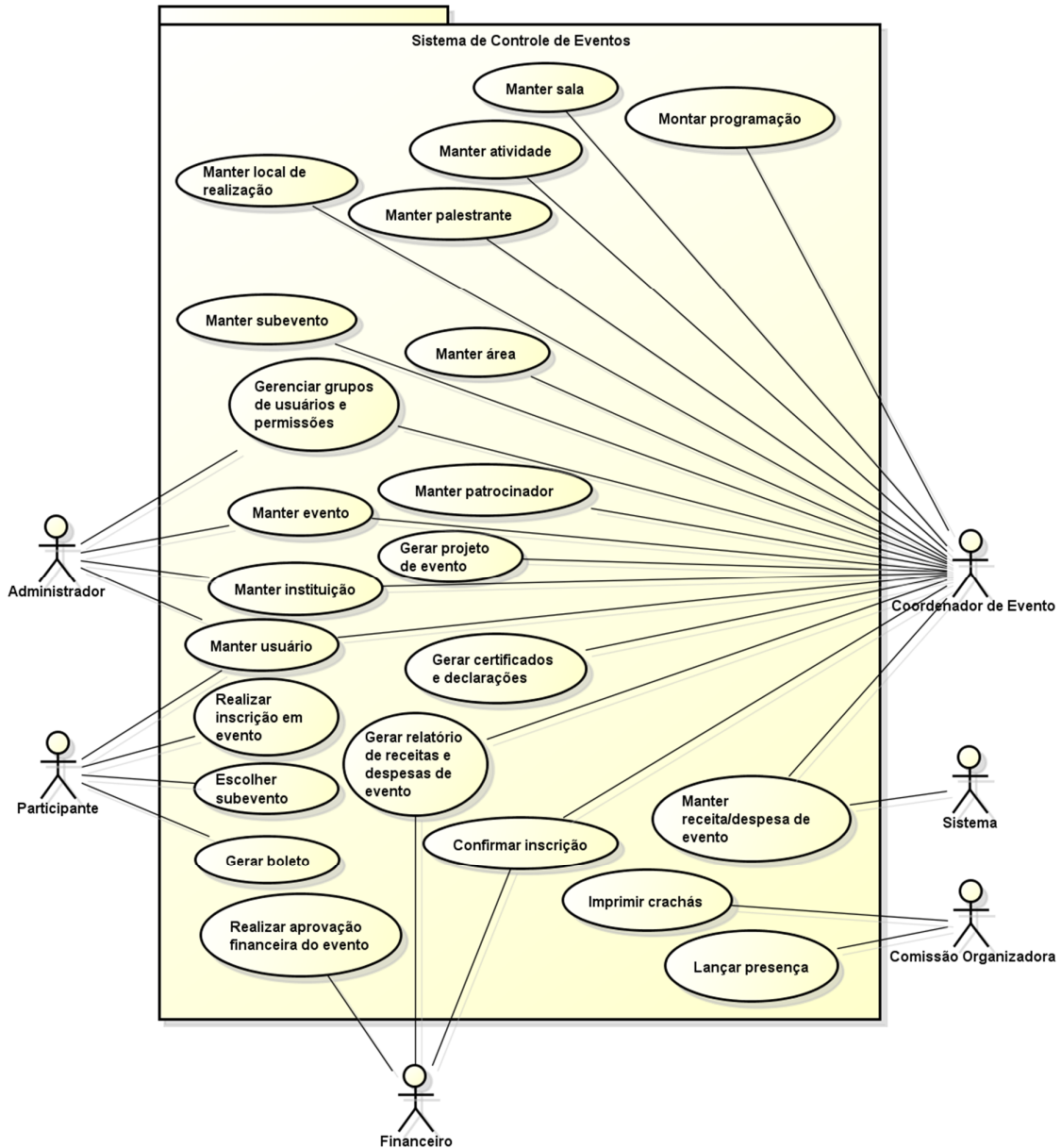
- Manter evento
- Manter subevento
- Manter área
- Manter instituição
- Manter local de realização
- Manter sala
- Manter palestrante
- Manter usuário
- Manter patrocinador
- Manter atividade
- Manter receita/despesa de evento
- Gerenciar grupos de usuários e permissões

- Realizar inscrição em evento
- Escolher subevento
- Gerar boleto
- Confirmar inscrição
- Realizar aprovação financeira do evento
- Imprimir crachás
- Lançar presença
- Visualizar site de evento
- Montar programação
- Gerar projeto de evento
- Gerar relatório de receitas e despesas de evento
- Gerar certificados e declarações

Juntamente à identificação destes casos de uso, foram levantados quais os atores que deveriam ser alocados juntamente a estes. O resultado deste levantamento foi a seguinte lista:

- Administrador
- Participante
- Coordenador de evento
- Financeiro
- Comissão organizadora
- Visitante
- Sistema

O passo seguinte ao levantamento dos casos de uso e seus atores envolveu a criação de um diagrama UML de casos de uso que mostrasse, de forma clara, a interação entre os mesmos (Figura 14).



**Figura 14 - Diagrama de casos de uso do sistema de controle de eventos**

### 5.2.5 Especificação de Requisitos de Usuário

Na etapa de especificação de requisitos de usuário os casos de uso anteriormente levantados tiveram os detalhes de seus funcionamentos expandidos, sendo adotada uma abordagem adaptada da apresentada durante a revisão bibliográfica. Esta adoção deu-se pelo fato de que os casos de uso foram elicitados e especificados juntamente aos requisitos funcionais do sistema, sendo assim, se o

fosse seguido modelo idêntico ao proposto, seriam desnecessariamente geradas informações duplicadas sobre os requisitos e casos de uso.

O Quadro 6 apresenta o requisito “O sistema deve permitir realizar inscrição em evento” que encontra-se especificado no Documento de Requisitos do Sistema (APÊNDICE 4). Nele é possível notar a existência de atributos identificados como pertencentes a requisitos, como: código de identificação, nome do requisito, função, entradas, fonte, saídas destino, ação, efeitos colaterais e responsáveis, bem como os atributos de classificação já conhecidos; bem como requisitos pertencentes a casos de uso, como nome do caso de uso, atores e cenários; e alguns itens que podem ser considerados como pertencentes aos dois grupos, como: pré-condições, pós-condições e observações.

Para este trabalho, assumiu-se que durante a especificação de requisitos de usuário, cada requisito funcional acarretaria em um caso de uso do sistema.

<b>REF013</b>	<b>O sistema deve permitir realizar inscrição em evento</b>
<b>Função</b>	Permitir a um visitante do site personalizado de um evento cadastrar-se, informando seus dados pessoais, com o objetivo de vincular seu nome aos participantes deste evento.
<b>Entradas</b>	Dados cadastrais do participante.
<b>Fonte das entradas</b>	Informação por parte do usuário visitante.
<b>Saídas</b>	Participante inscrito no evento, mas ainda não confirmado. Valor da inscrição-base calculado.
<b>Destino das saídas</b>	Cadastro de usuários. Inscrições do evento.
<b>Ação</b>	Um usuário do tipo visitante pode acessar a opção fornecida para a realização da inscrição no evento. Preenche, então, seus dados pessoais e outros dados cabíveis em um formulário onde o campo identificador é o CPF da pessoa, possibilitando o reaproveitamento de dados pessoais de uma inscrição realizada para outro evento.
<b>Pré-condições</b>	O usuário deve ter acessado o site personalizado do evento. O usuário deve conhecer os dados pessoais que serão dele cobrados.
<b>Pós-condições</b>	O usuário passa a ser conhecido como participante. O usuário está cadastrado e inscrito no evento, mas ainda não está confirmado. O usuário pode acessar o painel do participante e realizar as ações a ele permitidas.

<b>REF013</b>	<b>O sistema deve permitir realizar inscrição em evento</b>		
<b>Efeitos colaterais</b>	A inscrição não confirmada até data especificada acarreta na perda da vaga no evento.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Alto
<b>Observações:</b> Este requisito possui estabilidade baixa, pois possui alta volatilidade, estando diretamente ligado aos subeventos, que são objetos mutáveis e não totalmente conhecidos inicialmente. Mesmo com estabilidade alta, o risco deste requisito pode ser considerado alto, pois depende da manipulação do sistema por parte de um usuário não treinado para esta função, o que pode acarretar na invalidação de sua inscrição.			
<b>Caso de Uso</b>	<b>Realizar inscrição em evento</b>		
<b>Ator Principal</b>	<b>Visitante</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O visitante seleciona a opção de inscrição</li> <li>2. O sistema solicita o CPF do visitante</li> <li>3. O visitante informa o CPF</li> <li>4. O sistema solicita os dados cadastrais do visitante</li> <li>5. O visitante informa seus dados cadastrais</li> <li>6. O visitante grava a inserção dos dados cadastrais</li> <li>7. O usuário passa a ser participante não confirmado do evento</li> </ol>			
<b>Extensões</b>			
<p>3a. Já existe um usuário cadastrado com este CPF</p> <ol style="list-style-type: none"> <li>1. Sistema solicita a senha do usuário</li> <li>2. Usuário informa a senha Senha inválida Cancela o caso de uso</li> <li>3. Pular para o passo 7</li> </ol>			

**Quadro 6 - Exemplo de requisito com caso de uso**

Nesta etapa que começou a produção formal do documento de requisitos do sistema, com o objetivo de alocar todas as especificações e casos de uso produzidos. O documento de requisitos do sistema pode ser visualizado na íntegra no APÊNDICE D.



### 5.2.6 Prototipação

A etapa de prototipação não foi realizada neste estudo experimental, primeiramente por exigir um tempo e número de pessoas maior que as outras técnicas, o que acabou por torná-la inviável para este projeto e, em um segundo momento, pelo motivo dos interessados já terem contato com ferramentas de *software* para a gerência de eventos, o que poderia acabar tornando a prototipação um processo repetitivo e que não geraria os resultados esperados da mesma.

### 5.2.7 Elicitação de requisitos de sistema

Depois de levantados e especificados os requisitos funcionais do sistema, os esforços voltaram-se para as tentativas de extrair requisitos não funcionais, tendo como base o material produzido para os requisitos funcionais e informações levantadas nas entrevistas com os interessados.

O processo de elicitação de requisitos não funcionais foi menos moroso que os demais processos, tendo como produto de saída o documento de requisitos do sistema, com as alterações relacionadas aos mesmos. Os requisitos não funcionais do sistema inicialmente encontrados foram:

- A funcionalidade de subeventos do sistema deve conseguir trabalhar no formato de plugin;
- Cada evento deve possuir uma página personalizada;
- O visitante da página do evento deve conseguir facilmente realizar as ações básicas dele esperadas;
- Um usuário pode ser de diversos tipos, podendo ter diversas atribuições dentro do sistema ou de um evento;
- Todos os registros gravados devem conter um atributo apontando qual foi o último usuário que o manipulou;
- Todas as manipulações de dados do sistema devem ser arquivadas em um sistema de log, apontando o usuário, data, hora, entidade e propriedade manipulada;
- O sistema deve ficar disposto 100% do tempo;

- No caso de sair do ar e haja eventos em processo, o sistema deve ser restabelecido em, no máximo, 20 minutos;
- Todos os dados cadastrados devem possuir retenção ilimitada, possibilitando a eterna consulta de histórico;
- O sistema deve ser executado pela web;
- O servidor de hospedagem deve oferecer possibilidade de envio de *e-mail*;
- O servidor de hospedagem deve permitir a instalação de qualquer ferramenta que for solicitada;
- O sistema deve conseguir ser instalado em mais de um ambiente operacional;
- O sistema deve utilizar linguagem de programação orientada a objetos;
- O sistema deve utilizar uma linguagem *open-source* ou gratuita;
- O sistema deve utilizar linguagem e ferramental multiplataforma;
- O sistema deve seguir a normativa para emissão de declarações e certificados;
- O dados cadastrais dos usuários não podem ser divulgados sem a permissão dos mesmos.

#### 5.2.8 Especificação e modelagem de requisitos de sistema

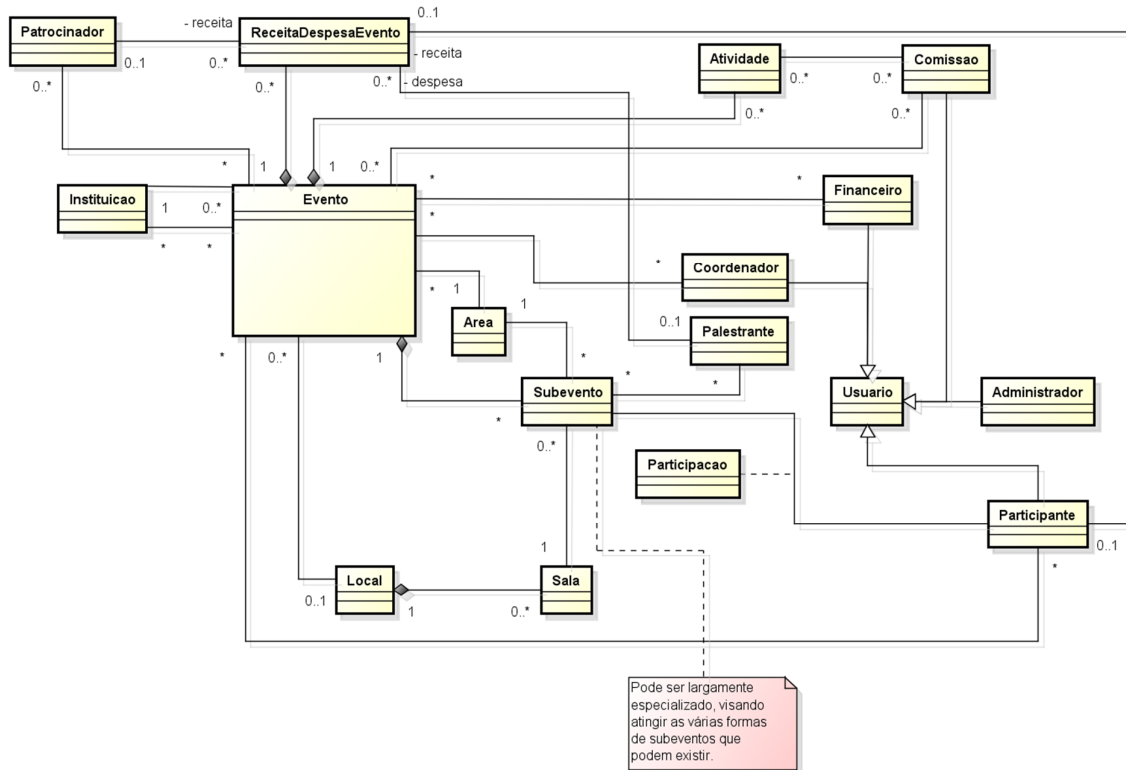
A etapa de especificação e modelagem de requisitos de sistema não foi utilizada somente com este propósito, sendo agregada a esta, também, excetuando-se o diagrama de casos de uso, as questões relacionadas à modelagem dos requisitos funcionais, que foram deixados para esta etapa por ter-se entendido que seria um momento de maior esclarecimento sobre o funcionamento e restrições gerais do sistema.

A especificação de requisitos de sistema, ou não funcionais, assim como a elicitação dos mesmos, foi um processo menos moroso que os demais, sendo realizada a divisão dos requisitos nas categorias não funcionais apresentadas por Sommerville (2011) e Engholm Júnior (2010). O resultado deste processo, presente do documento de requisitos do sistema, está representado no Quadro 7.

	<b>Categoria</b>	<b>Ident.</b>	<b>Descrição</b>
Requisitos de produto	Usabilidade	RNU001	A funcionalidade de subeventos do sistema deve conseguir trabalhar no formato de plugin.
		RNU002	Cada evento deve possuir uma página personalizada.
		RNU003	O visitante da página do evento deve conseguir facilmente realizar as ações básicas dele esperadas.
	Proteção	RNP001	Um usuário pode ser de diversos tipos, podendo ter diversas atribuições dentro do sistema ou de um evento.
		RNP002	Todos os registros gravados devem conter um atributo apontando qual foi o último usuário que o manipulou.
		RNP003	Todas as manipulações de dados do sistema devem ser arquivadas em um sistema de log, apontando o usuário, data, hora, entidade e propriedade manipulada.
	Confiança	RNC001	O sistema deve ficar disposto 100% do tempo.
		RNC002	No caso de sair do ar e haja eventos em processo, o sistema deve ser restabelecido em, no máximo, 20 minutos.
	Espaço	RNE001	Todos os dados cadastrados devem possuir retenção ilimitada, possibilitando a eterna consulta de histórico.
	Requisitos organizacionais	Ambiental	RNA001
RNA002			O servidor de hospedagem deve oferecer possibilidade de envio de <i>e-mail</i> .
RNA003			O servidor de hospedagem deve permitir a instalação de qualquer ferramenta que for solicitada.
RNA004			O sistema deve conseguir ser instalado em mais de um ambiente operacional.
Desenvolvimento		RND001	O sistema deve utilizar linguagem de programação orientada a objetos.
		RND002	O sistema deve utilizar uma linguagem <i>open-source</i> ou gratuita.
		RND003	O sistema deve utilizar linguagem e ferramental multiplataforma.
Requisitos externos	Reguladores	RNR001	O sistema deve seguir a normativa para emissão de declarações e certificados.
	Éticos	RNT001	Os dados cadastrais dos usuários não podem ser divulgados sem a permissão dos mesmos.

**Quadro 7 - Requisitos não funcionais do sistema de controle de eventos**

Outro processo abordado nesta etapa foi a criação do diagrama de classes com perspectiva conceitual, que tem por objetivo ajudar na visualização do modelo de dados do sistema (Figura 15).



**Figura 15 - Diagrama de classes do sistema de controle de eventos**

Os produtos da realização desta etapa são os quatro documentos abordados neste trabalho: o plano de gerenciamento de requisitos (APÊNDICE A), documento de visão (APÊNDICE B), glossário do projeto (APÊNDICE C) e o documento de requisitos do sistema (APÊNDICE D), neste momento, já em suas versões finais, prontos para serem lidos, avaliados e aprovados (ou rejeitados) pelos interessados nestes.

### 5.2.9 Revisões

O processo de revisões consistiu na liberação dos documentos obtidos como produto da engenharia de requisitos (o plano de gerenciamento de requisitos

(APÊNDICE A), documento de visão (APÊNDICE B), glossário do projeto (APÊNDICE C) e o documento de requisitos do sistema (APÊNDICE D)) para os interessados, que já estavam cientes dos trabalhos que estavam sendo realizados através de validações esporádicas através de conversas e interações através de *e-mails*.

## 6 CONCLUSÃO

### 6.1 CONSIDERAÇÕES FINAIS

Considerando-se o fato de que uma das tarefas mais complexas que é desempenhada pelos engenheiros de *software* é a etapa de reconhecimento e entendimento dos requisitos necessários a um sistema, pode-se chegar à conclusão de que a produção de um sólido material para documentação destes requisitos faz-se extremamente necessária já nas fases iniciais do projeto, pois se o mesmo não for feito, os requisitos terão que ser gradualmente descobertos, o que pode vir a gerar uma possibilidade mais de se desenvolver funções do sistema que não reflitam o desejado pelos interessados neste.

Tendo como base este fato, este trabalho objetivou mostrar o estudo realizado, considerando as técnicas e ferramentas da subárea da engenharia de *software* denominada engenharia de requisitos, com o objetivo de conhecer e documentar os requisitos de um proposto sistema de controle de eventos para a Universidade Tecnológica Federal do Paraná (UTFPR) câmpus Medianeira, que tem sua criação justificada na necessidade por um sistema do gênero que consiga gerenciar os diversos eventos dos diferentes formatos e áreas que ocorrem anualmente na instituição.

O processo de aplicação das técnicas sobre este estudo experimental gerou resultados parciais, que podem ser classificados em três diferentes áreas: elicitación, especificación e validación. Porém, de forma prática, o resultado final do mesmo gerou uma documentação dos requisitos dos usuários do sistema, que é a declaração do que deve ser projetado e desenvolvido, quando tais etapas da engenharia de software forem atingidas por este projeto. Estes produtos da aplicação da engenharia de requisitos são compostos, basicamente, pelo documento de requisitos do sistema (APÊNDICE D) e seus documentos auxiliares: plano de gerenciamento de requisitos (APÊNDICE A), documento de visão (APÊNDICE B) e glossário do projeto (APÊNDICE C).

Portanto, considera-se o seguinte sobre os objetivos específicos deste trabalho:

- Estudar conceitos, técnicas e ferramentas da engenharia de requisitos: atingido através da revisão bibliográfica de diversos autores reconhecidos na área de engenharia de software e de requisitos;
- Identificar e aplicar técnicas e ferramentas para modelagem orientada a objetos envolvidas no processo de engenharia de requisitos: atingindo através do estudo e aplicação de UML e seus diagramas no estudo experimental proposto;
- Efetuar o levantamento, a classificação e a validação de requisitos junto aos interessados no sistema: atingido através da realização de entrevistas com interessados, análise de documentos e de sistemas, levantamento, especificação e validação de requisitos;
- Elaborar um documento formal de requisitos com base nos dados extraídos: atingido através da criação do documento de requisitos do sistema de controle de eventos da UTFPR e seus documentos auxiliares.

## 6.2 SUGESTÕES DE TRABALHOS FUTUROS

Espera-se que este trabalho possa ser utilizado na real projeção e implementação do sistema para controle de eventos para a UTFPR proposto em seu estudo experimental.

Toma-se também como sugestão, com o intuito de lapidar a documentação produzida, a aplicação de técnicas envolvidas com a engenharia de requisitos inicialmente descartadas, como a prototipação, a rastreabilidade de requisitos e o gerenciamento de riscos.

## REFERÊNCIAS

- ENGHOLM JÚNIOR, H. **Engenharia de Software na prática**. São Paulo: Novatec, 2010.
- FOWLER, M. **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. 3. ed. Porto Alegre: Bookman, 2005.
- HULL, E.; JACKSON, K.; DICK, J. **Requirements Engineering**. 3. ed. Londres: Springer, 2011.
- LARMAN, C. **Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao Processo Unificado**. 2. ed. Porto Alegre: Bookman, 2004.
- LIMA, A. D. S. **UML 2.0: do requisito à solução**. 2. ed. São Paulo: Érica, 2007.
- OMG, O. M. G. **Unified Modeling Language Infrastructure.**, 2011. Disponível em: <<http://www.omg.org/cgi-bin/doc?formal/2011-08-05.pdf>>. Acesso em: 14 nov. 2011.
- PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. 7. ed. Porto Alegre: McGraw-Hill, 2011.
- SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
- ZAVE, P. Classification of Research Efforts in Requirements Engineering. **ACM Computing Surveys**, 29, Dezembro 1997. 315-321.



## APÊNDICE A - PLANO DE GERENCIAMENTO DE REQUISITOS

### SISTEMA DE CONTROLE DE EVENTOS - PLANO DE GERENCIAMENTO DE REQUISITOS

#### INTRODUÇÃO

Este plano de gerenciamento de requisitos contém a estratégia de elicitação, tipos de requisitos e sua estrutura, bem como, nomenclatura e sequência que será utilizada durante todo o ciclo de vida do projeto. Sua finalidade básica é garantir o relacionamento de um elemento de projeto a outros elementos relacionados.

#### FINALIDADE

A finalidade deste documento é apresentar o planejamento de requisitos em relação ao controle e monitoramento do escopo do projeto do sistema para controle de eventos da UTFPR – Câmpus Medianeira.

#### ESCOPO

Este artefato descreve o plano de gerenciamento de requisitos a ser utilizado no projeto do sistema para controle de eventos da UTFPR Câmpus Medianeira, com o intuito de viabilizar o atendimento aos requisitos identificados para o sistema.

Este projeto teve origem na necessidade da UTFPR por um sistema que controlasse, de forma genérica, os eventos de diferentes áreas que ocorrem anualmente na instituição.

Deste modo, o escopo deste projeto é possibilitar a automatização do gerenciamento destes eventos, tendo em vista que existem diversos setores da mesma instituição interessados no correto funcionamento do mesmo.

#### DEFINIÇÕES, ACRÔNIMOS E ABREVIACÕES

Vide Glossário do Projeto.

#### VISÃO GERAL

A estrutura geral deste documento contempla os seguintes itens:

- **Gerenciamento de requisitos:** mostra a estrutura elaborada para o tratamento dos requisitos ao longo do projeto;

- **Fornecedores de requisitos:** identifica os fornecedores de requisitos do projeto;
- **Estratégia de gerenciamento de requisitos:** mostra as estratégias a serem utilizadas para o levantamento de requisitos;
- **Marcos:** mostra datas relacionadas ao projeto no que concerne o gerenciamento de requisitos.

## GERENCIAMENTO DE REQUISITOS

### Responsabilidades

Subprocesso / Atividade	Responsável	Papel Executado
Planejar gerenciamento de requisitos	Analista de Requisitos	Analista de Requisitos
Manter integridade dos requisitos	Equipe de projeto	Analista de Requisitos
Gerenciamento de configuração	Analista de Requisitos	Analista de Requisitos
Identificar fornecedores de requisitos	Interessado inicial	Gerente de Projeto
Analisar problema	Equipe de projeto	Analista de Requisitos
Identificar necessidades	Equipe de projeto	Analista de Requisitos
Levantar características do produto	Equipe de projeto	Analista de Requisitos
Definir o sistema	Equipe de projeto	Analista de Requisitos
Validar escopo do sistema	Equipe do cliente	Patrocinador do Projeto e Analista de Processos
Especificar requisitos funcionais e não funcionais	Equipe do projeto e arquiteto de software	Analista de Requisitos
Especificar cenários operacionais	Equipe do projeto	Analista de Requisitos
Refinar requisitos	Equipe do projeto	Analista de Requisitos
Validar requisitos com fornecedores de requisitos	Equipe do projeto	Analista de Requisitos
Aprovar requisitos	Equipe do cliente e equipe do	Patrocinador do Projeto e

<b>Subprocesso / Atividade</b>	<b>Responsável</b>	<b>Papel Executado</b>
	projeto	Analista de Processos
Estabelecer ambiente de requisitos	Equipe do projeto	Analista de Requisitos

## **FERRAMENTAS, AMBIENTE E INFRAESTRUTURA**

Para a realização da etapa de engenharia de requisitos será necessária a utilização do editor de textos Microsoft Word 2010, que será utilizado na elaboração de todos os documentos oriundos deste processo. Também faz-se necessário o uso de uma ferramenta de modelagem UML, sendo escolhida a Astah Community 6.5.

## **IDENTIFICAÇÃO DOS REQUISITOS**

Este projeto obedecerá a uma numeração de requisitos do tipo TIPSEQ, onde TIP é um conjunto de 3 caracteres que caracteriza o tipo do requisito e SEQ é um número sequencial, formatado com zeros à esquerda de forma que sejam obtidos 3 caracteres, incrementado para cada tipo.

Os valores válidos para TIP estão descritos na tabela a seguir:

<b>Tipo de Requisito</b>	<b>Abreviação</b>
Requisito Funcional	REF
Requisito de usabilidade	RNU
Requisito de desempenho	RNS
Requisito de proteção	RNP
Requisito de confiança	RNC
Requisito de espaço	RNE
Requisito ambiental	RNA
Requisito de desenvolvimento	RND
Requisito operacional	RNO

Exemplos válidos para apresentação de nomes de possíveis requisitos são:

- REF001
- RNU020

- RNP100
- RNO999

## TIPOS DE REQUISITOS E ARTEFATOS

Este item identifica os artefatos gerados pelo processo de engenharia de requisitos do projeto, estando previstos os seguintes:

- **Plano de gerenciamento de requisitos:** documento que apresenta planejamento relacionado à gerência de requisitos do projeto, servido de base para o controle e monitoramento do escopo do produto. Identifica os fornecedores de requisitos;
- **Documento de visão:** responsável por registrar o problema de negócio do cliente, necessidades e expectativas. Identifica também as características macro que a solução deverá possuir para atender a estas necessidades, podendo apresentar necessidades, macro características do produto, requisitos funcionais e não funcionais e eventos;
- **Glossário do projeto:** documento necessário para descrever termos e abreviações utilizadas na documentação e no projeto;
- **Lista de requisitos:** documento que contém a listagem de todos os requisitos conhecidos para o sistema, tendo sido inserida, em sua versão final, no contexto do documento de requisitos do sistema;
- **Documento de requisitos do sistema:** documento que apresenta os requisitos do sistema de forma completa e detalhada, possibilitando o total entendimento das necessidades e funcionalidades do sistema para qualquer pessoa que venha a lê-lo. Este documento pode vir a agregar funcionalidades antes exercidas por outros documentos, como, por exemplo, a lista de requisitos. A partir do momento de tal agregação, a criação ou atualização destes documentos poderá não fazer-se necessária.

## ORGANIZAÇÃO ESTRUTURAL

Todos os documentos e demais artefatos produzidos no processo de engenharia de requisitos deste sistema serão mantidos e atualizados a partir de um mesmo diretório específico localizado no computador do analista de requisitos responsável pelo mesmo.

## FORNECEDORES DE REQUISITOS

Esta seção apresenta a identificação, categorização e classificação dos fornecedores de requisitos para o sistema.

### Critérios utilizados

Os fornecedores podem, basicamente, ser agrupados sob os seguintes itens:

<b>Categoria / Papel</b>	<b>Tipo (1)</b>	<b>Nível de Decisão / Influência</b>	<b>Identificação / Descrição</b>
Patrocinador do Projeto	CPA	Alto	Possui decisão final sobre questionamentos e sugestões em relação ao sistema.
Gerente do projeto	CPA	Alto	Responsável pelo gerenciamento do projeto.
Gestor / Usuário	UFI	Médio	Possui decisão final relacionada às funcionalidades do sistema
Analista técnico / processos	ATC	Alto	Decide sobre questões técnicas e funcionais esclarecidas pelos usuários.

<sup>1</sup>: CPA = Cliente patrocinador / UFI = Usuário final / ATC = Área técnica do cliente

### Fornecedores

<b>Nome</b>	Interessado inicial
<b>Cargo na empresa</b>	Professor / organizador de eventos da área de informática
<b>Papel no projeto</b>	Gerente do projeto
<b>Responsabilidades no projeto</b>	Fornecimento de requisitos gerais e responsável pela identificação dos demais fornecedores de requisitos.
<b>Comentários</b>	Pode ser encontrado na sala I21 da UTFPR, podendo ser abordado somente nos momentos em que não estiver em aula.
<b>Estratégias para levantamento de requisitos</b>	Entrevistas, reuniões e análise de sistemas já existentes e de documentos.

<b>Nome</b>	Interessada em eventos
<b>Cargo na empresa</b>	Funcionária da Assessoria de Cerimoniais e Eventos da UTFPR
<b>Papel no projeto</b>	Usuário
<b>Responsabilidades no projeto</b>	Fornecimento de requisitos específicos referentes à área de gerenciamento de eventos.
<b>Comentários</b>	Pode ser encontrada em sua sala nos horários matutino e vespertino.
<b>Estratégias para levantamento de requisitos</b>	Entrevistas, reuniões, questionários e análise de sistemas já existentes.

<b>Nome</b>	Interessado financeiro
<b>Cargo na empresa</b>	Funcionário da FUNTEF
<b>Papel no projeto</b>	Usuário
<b>Responsabilidades no projeto</b>	Fornecimento de requisitos específicos referentes à área financeira de eventos.
<b>Comentários</b>	Pode ser encontrado na sala da FUNTEF nos horários matutino e vespertino.
<b>Estratégias para levantamento de requisitos</b>	Entrevistas, reuniões e questionários.

## **ESTRATÉGIA DE GERENCIAMENTO DE REQUISITOS**

Este projeto não conterá controle e rastreamento de requisitos.

### **ESTRATÉGIAS**

Como estratégias para o levantamento de requisitos do projeto, deverão ser identificados os seguintes itens:

- Necessidades dos fornecedores de requisitos e características que as atendam;
- Características que se relacionam com requisitos funcionais e não funcionais;
- Relacionamento entre requisitos funcionais e não funcionais;
- Eventos relacionados às funcionalidades do sistema.

## ATRIBUTOS DOS REQUISITOS

Cada requisito levantado para o sistema pode possuir determinados atributos que facilitam sua identificação e classificação. São eles:

### **Benefício**

Este atributo indica o grau de benefício, do ponto de vista do usuário, do requisito ao qual este é referente, podendo ser classificado como:

- **Crítico:** quando o requisito é imprescindível para o sucesso do software;
- **Importante:** quando o não atendimento do requisito não gera fracasso do projeto, mas impacta na satisfação do usuário;
- **Desejável:** caso o requisito não for atendido, o impacto na satisfação do cliente será baixo ou nulo.

### **Estabilidade**

Este atributo quantifica a possibilidade da ocorrência de mudanças em determinado requisito, o que pode ocorrer devido à alteração no grau de entendimento do mesmo pelos interessados ou pela equipe de software. Este grau de entendimento pode variar de acordo com a maturação de conceitos por parte dos interessados, podendo ser classificado como:

- **Alta:** o requisito tem um alto grau de entendimento e baixa probabilidade de mudanças;
- **Média:** o requisito tem considerável chance de mudanças, pois provavelmente ainda não está bem entendido pela equipe de software ou ainda há pendências de definições por parte dos interessados;
- **Baixa:** o requisito possui alta probabilidade de mudanças, pois possui baixa maturidade, sendo este nível geralmente associado a requisitos de alta complexidade ou com muitas pendências de definição.

### **Situação**

A situação de um requisito indica o estado de aprovação no qual este se encontra, podendo ser classificado como:

- **Proposto:** o atributo proposto ainda não pode ser utilizado na implementação (ou outras fases do software), pois ainda se encontra em análise pela equipe de software e muitas vezes até pelo cliente;
- **Aprovado:** o atributo está apto a ser utilizado, pois já obteve aprovação e acordo entre a equipe de software e os interessados;
- **Cancelado:** o requisito com esta situação deve ser descartado do projeto.

### Risco

Neste atributo são tratados os riscos oferecidos por cada requisito, sendo que estes são os requisitos que podem vir a prejudicar o projeto, direta ou indiretamente, principalmente nas áreas financeiras e de cronograma do mesmo. O risco pode ser classificado como:

- **Alto:** o requisito tem baixa estabilidade e alta complexidade, contendo dependências externas ao software;
- **Médio:** o requisito tem estabilidade e complexidade consideradas medianas, não contendo dependências externas;
- **Baixo:** o requisito tem baixo risco por ter alta estabilidade e baixa complexidade, não contendo dependências externas.

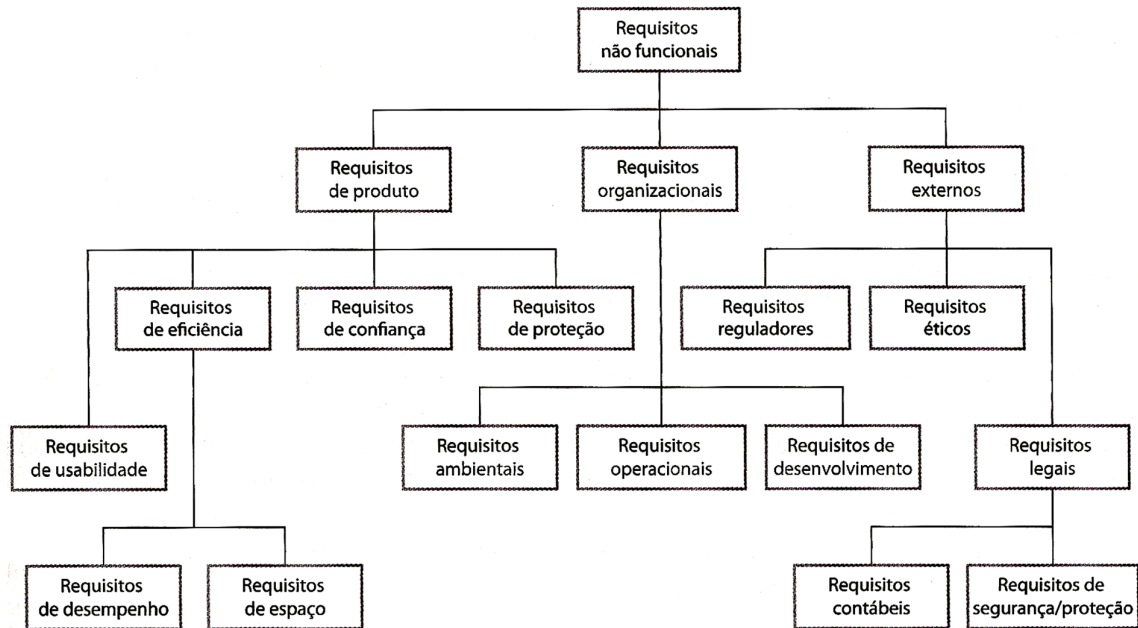
### Outros atributos

- **Responsável:** identifica, de maneira textual, quem é (são) o(s) fornecedor(es) responsável(is) pelo requisito;
- **Observações:** campo destinado à inserção de observações diversas sobre o requisito em questão.

## REQUISITOS NÃO FUNCIONAIS

As categorias de requisitos não funcionais deste projeto são apresentadas sob uma visão hierárquica de possibilidades, sugerida por Sommerville (2011):





## ATIVIDADES PREVISTAS

Criação e validação dos documentos de requisitos do sistema.

## APÊNDICE B – DOCUMENTO DE VISÃO

### SISTEMA DE CONTROLE DE EVENTOS – DOCUMENTO DE VISÃO

#### INTRODUÇÃO

A finalidade deste documento é coletar, analisar e definir as necessidades de nível macro do Sistema de Controle de Eventos, que terá seus requisitos expostos através da aplicação de técnicas de engenharia de requisitos, obtendo um nível de detalhamento que possibilite a realização das próximas etapas da engenharia de software. O foco inicial do sistema a ser desenvolvido é um software “piloto”, com as funcionalidades básicas requisitadas pelos futuros usuários do mesmo, podendo este, futuramente, ter funcionalidades agregadas.

#### OPORTUNIDADE DE NEGÓCIO

Atualmente, há uma grande necessidade de um sistema para controle genérico de eventos por parte da Universidade Tecnológica Federal do Paraná (UTFPR) Medianeira, sabendo que esta atualmente possui diversos aplicativos desenvolvidos para eventos específicos que são, sempre que possível, reutilizados para novos eventos. Porém sabe-se que aplicativos desenvolvidos com fins específicos dificilmente atenderão de forma completa outros fins (um dos motivos para isto, provavelmente é a divergência em seus requisitos), inclusive na área de gerência de eventos, onde as pessoas envolvidas com a organização de cada um destes pode necessitar gerenciar diferentes recursos de formas específicas e únicas.

#### PROBLEMAS

<b>O problema de</b>	Haver a necessidade de realizar a criação de um novo sistema/site para cada evento ocorrido na UTFPR.
<b>Afeta</b>	O desempenho da realização das diversas atividades de um evento.
<b>Cujo impacto é</b>	A desorganização, o retrabalho e os gastos gerados pela necessidade da construção de um novo site a cada novo evento.
<b>Benefícios de uma solução seriam</b>	<ul style="list-style-type: none"> <li>• Automatização e agilização do processo de lançamento do evento</li> <li>• Aumento do controle dos coordenadores dos eventos sobre os detalhes dos mesmos</li> <li>• Aumento da facilidade de gerenciamento de determinadas características, visto que determinados grupos de pessoas participarão da organização de diversos eventos.</li> <li>• Padronização do formato de trabalho de diferentes equipes.</li> </ul>

<b>O problema de</b>	Não possuir um método padronizado e funcional para participantes se informarem e inscreverem em eventos.
<b>Afeta</b>	Participantes de eventos.
<b>Cujo impacto é</b>	Dificuldade de obtenção de informações de realização da inscrição e pagamento para determinado evento.
<b>Benefícios de uma solução seriam</b>	<ul style="list-style-type: none"> <li>• Facilitar o acesso dos participantes por um portal padronizado</li> <li>• Permissão para participante realizar ações de gerenciamento e controle de sua inscrição</li> <li>• Atrair possíveis novos participantes</li> </ul>

<b>O problema de</b>	Não possuir uma forma automatizada e integrada para verificação de questões financeiras ligadas a eventos.
<b>Afeta</b>	Funcionários da FUNTEF.
<b>Cujo impacto é</b>	Dificuldade na integração de informações financeiras com detalhes de eventos e de participantes destes.
<b>Benefícios de uma solução seriam</b>	<ul style="list-style-type: none"> <li>• Possibilitar a visualização de detalhes financeiros de cada evento e participante, bem como a manipulação destes</li> <li>• Diminuir o tempo de confirmação de inscrições</li> <li>• Geração do projeto do evento, contendo detalhes financeiros</li> </ul>

## NECESSIDADES DOS INTERESSADOS

<b>Descrição</b>	Os coordenadores de eventos necessitam controlar, de forma padronizada e organizada, os detalhes de seus eventos.
<b>Prioridade</b>	Crítica.
<b>Preocupações</b>	Erros diversos durante processos de gestão de eventos.
<b>Solução proposta</b>	Desenvolvimento de um sistema para controle de eventos que consiga atender eventos de diferentes naturezas e necessidades.

<b>Descrição</b>	Os participantes de eventos precisam conseguir realizar inscrição, e gerenciar detalhes desta, no evento desejado.
<b>Prioridade</b>	Crítica.
<b>Preocupações</b>	Facilidade para o participante gerenciar os detalhes de sua participação no evento.
<b>Solução proposta</b>	Desenvolvimento de uma interface (no modelo de site), personalizável para cada evento, que exponha os dados destes e possibilite aos participantes realizarem suas inscrições, bem como gerenciar detalhes

	da mesma, como: participações em subeventos, impressão de boleto, confirmação de pagamento, entrar em contato, dentre outros.
--	---

<b>Descrição</b>	Os funcionários da FUNTEF necessitam visualizar e gerenciar os detalhes financeiros ligados a cada evento.
<b>Prioridade</b>	Crítica.
<b>Preocupações</b>	Integração dos dados de pagamentos a dados de inscrições.
<b>Solução proposta</b>	Desenvolvimento de uma área que possibilite aos funcionários conhecerem os detalhes de pagamento de cada participante, confirmando ou não suas inscrições.

## RESUMO DOS USUÁRIOS

Nome	Descrição	Responsabilidades
Visitante	Usuário que pode navegar pela área pública do site de cada evento, podendo visualizar os detalhes do mesmo e realizar sua inscrição.	<ul style="list-style-type: none"> <li>• Visualizar dados dos eventos</li> <li>• Visualizar detalhes de subeventos</li> <li>• Realizar inscrição em evento</li> </ul>
Participante	Usuário cadastrado em determinado evento que pode utilizar o site para gerenciar as características de sua participação no evento.	<ul style="list-style-type: none"> <li>• Visualizar dados dos eventos</li> <li>• Visualizar detalhes de subeventos</li> <li>• Gerenciar informações de inscrição em evento</li> <li>• Pagar inscrição em evento</li> </ul>
Coordenador de Evento	Usuário que está apto a gerenciar as informações de um evento a ele incumbido. Um usuário pode ser Coordenador de Evento em um determinado evento, porém em outros não.	<ul style="list-style-type: none"> <li>• Administrar informações cadastrais do evento</li> <li>• Administrar subeventos</li> <li>• Administrar participantes do evento</li> <li>• Administrar palestrantes do evento</li> <li>• Administrar datas e horários do evento</li> <li>• Gerar grade de subeventos</li> <li>• Administrar custos do evento</li> <li>• Criar projeto do evento</li> <li>• Visualizar resumos do evento</li> <li>• Imprimir certificados</li> <li>• Designar funções para o evento</li> <li>• Administrar patrocinadores do evento</li> <li>• Administrar receitas do evento</li> <li>• Administrar comissão organizadora do evento</li> </ul>

Nome	Descrição	Responsabilidades
		<ul style="list-style-type: none"> <li>• Administrar locais de realização do evento</li> <li>• Administrar instituições do evento</li> <li>• Administrar atividades do evento</li> </ul>
Financeiro	Usuário que pode visualizar e gerenciar dados de pagamentos de todos os eventos.	<ul style="list-style-type: none"> <li>• Confirmar inscrições</li> <li>• Visualizar pagamentos de inscrições</li> </ul>
Administrador	Usuário responsável por gerenciar características gerais do sistema que não estão na alçada do Coordenador de Eventos e do Financeiro	<ul style="list-style-type: none"> <li>• Cadastrar novo evento</li> <li>• Administrar usuários</li> <li>• Designar funções e permissões a outros usuários</li> <li>• Administrar características gerais do sistema</li> <li>• Administrar tipos de subeventos</li> </ul>
Comissão Organizadora	Usuário que tem funções auxiliares para determinado projeto, como emissão de crachás, recepção, conferência de presença, dentre outros. É um estado dependente de um evento e tem suas características alteradas de acordo com a necessidade do Coordenador do Evento em questão. As atribuições deste tipo de usuário perante o sistema são diversas, porém normalmente o usuário deste tipo normalmente terá poucas, uma ou nenhuma destas atribuições aplicadas a si.	<ul style="list-style-type: none"> <li>• Emitir de crachás</li> <li>• Conferir presenças</li> <li>• Visualizar informações cadastrais de participantes</li> </ul>

## AMBIENTE DO SISTEMA

O sistema deverá ser acessado e administrado, por qualquer um dos tipos de usuário, via web, não havendo a necessidade de nenhum software específico instalado no computador dos usuários.

## VISÃO GERAL DO PRODUTO

O Sistema de Controle de Eventos deve permitir a completa gerência dos detalhes pertinentes aos eventos ocorridos na UTFPR Câmpus Medianeira, possibilitando, na mesma instalação deste, a criação e gestão de diversos eventos com diferentes escopos, estratégias e datas.

O sistema deve prover uma página personalizável para cada evento, devendo esta possuir aspectos públicos, porém possibilitando a inscrição de participantes no evento em questão.

Os coordenadores de eventos devem conseguir ter total controle sobre as atualizações e detalhes que venham a surgir, relativos ao evento que está sob sua responsabilidade.

### **Perspectiva do produto**

Devido a esta primeira versão do sistema ser considerada como uma versão “piloto”, várias características interessantes, porém não essenciais, não serão criadas, sendo deixado reservado espaço para futuras manutenções. A perspectiva inicial é garantir a gerência de eventos diversos da maneira mais pura que for possível, porém atendendo a todas as necessidades impostas pelos interessados no sistema.

### **Resumo das capacidades do produto**

<b>Benefícios para o cliente</b>	<b>Recursos do sistema</b>
Visibilidade dos detalhes públicos de eventos no ambiente web	Site personalizado para cara evento
Possibilidade de realização de inscrição de participante via web	Site personalizado para cara evento
Administração eficaz funcionamento de eventos	Área administrativa do evento
Administração de todos os detalhes funcionais do sistema	Área administrativa do sistema
Realização de serviços auxiliares a cada evento	Área administrativa da comissão organizadora do evento

### **Características do produto**

O sistema deve:

- Exibir informações públicas de eventos;
- Permitir aos participantes se cadastrarem e pagarem a inscrição de eventos;
- Permitir a um coordenador de evento gerenciar todas as características que envolvem um evento, não importando seu escopo;
- Permitir à comissão organizadora realizar seu trabalho de auxiliar o coordenador de determinado evento;

- Permitir ao administrador do sistema administrar e visualizar todas as informações pertinentes a eventos em geral.

## **RESTRIÇÕES ECONÔMICAS**

Este projeto não possui detalhes financeiros, pois será desenvolvido por uma equipe de estagiários da própria instituição, a ser selecionada no momento devido, e gerenciado por algum servidor desta mesma que se julgue competente para o mesmo.

## **RESTRIÇÕES DE PRAZO**

Não há restrições de prazo predefinidas, pois não há prazos de conclusão ou implantação previstos para o sistema.

## **PENDÊNCIAS E PRIORIDADES**

O projeto tem como prioridade os seguintes itens (em ordem de relevância):

1. Cadastro de eventos
2. Site com informações dos eventos
3. Inscrição de participante
4. Confirmação de pagamento
5. Cadastro de subeventos
6. Manipulação de subeventos por parte dos participantes
7. Cadastro avançado de eventos
8. Cadastro avançado de subeventos

## **RESTRIÇÕES DE DOCUMENTAÇÃO**

Documento de Requisitos do Sistema.

## APÊNDICE C – GLOSSÁRIO DO PROJETO

### SISTEMA DE CONTROLE DE EVENTOS – GLOSSÁRIO DO PROJETO

#### INTRODUÇÃO

Este documento é utilizado para definir a terminologia específica do domínio do problema do projeto do Sistema de Controle de Eventos, explicando termos que podem ser desconhecidos do leitor das descrições de casos de uso do sistema ou de outros documentos do projeto.

#### PROPÓSITO E ESCOPO

Este documento captura o vocabulário comum para o projeto, especificando termos técnicos relacionados ao Sistema de Controle de Eventos e seu domínio.

#### DEFINIÇÕES

<b>Termo</b>	<b>Descrição</b>
UTFPR	Universidade Tecnológica Federal do Paraná.
FUNTEF	Fundação de Apoio à Educação, Pesquisa e Desenvolvimento Científico e Tecnológico da UTFPR.
Evento	Acontecimento ocorrido na instituição com o objetivo de reunir diversos participantes com interesses comuns.
Subevento	Divisão interna de um evento, ou seja, um evento é composto de diversos subeventos. Exemplo: palestra, minicurso, mesa redonda, visita técnica etc.
Manter dados cadastrais	Indica que podem ser realizadas as ações padrão de inserir, alterar, apagar e consultar elementos de determinado tipo, o qual deverá ser explicitamente informado.



## APÊNDICE D – DOCUMENTO DE REQUISITOS DO SISTEMA

### SISTEMA DE CONTROLE DE EVENTOS – DOCUMENTO DE REQUISITOS DO SISTEMA

#### INTRODUÇÃO

Este documento tem por finalidade descrever os requisitos funcionais e não funcionais de um Sistema de Controle de Eventos proposto para a Universidade Tecnológica Federal do Paraná (UTFPR) Medianeira.

#### ESCOPO

##### Definição do escopo

- Desenvolvimento de um sistema de controle de eventos;
- Atendimento a diferentes tipos de eventos;
- Simplificação da realização das atividades de inscrição por parte de participantes;
- Automatização e integração de ferramentas para gerência de eventos por parte dos coordenadores de eventos;
- Facilitação da criação de sites personalizados de eventos;
- Divulgação de eventos pela internet.

##### Fora do Escopo

- 

#### PREMISSAS DE REQUISITOS

Premissa	Impacto
Os interessados no projeto deverão se posicionar com relação aos documentos de requisitos entregues pela equipe de software em no máximo três dias, devendo aprova-los ou rejeitá-los formalmente neste período. Em caso de rejeição, seus motivos devem ser explicados.	Prazos.
Todo o levantamento de requisitos será executado nas dependências da UTFPR.	Não se aplica.
Este documento pode deixar de contemplar, nesta primeira versão, requisitos que forem julgados não essenciais ao funcionamento do sistema, podendo vir a ser inseridos nas	Prazos e custos.

Premissa	Impacto
próximas versões próximas versões do mesmo.	
Este documento apenas levanta o requisitos necessários ao sistema de controle de eventos, não implicando na necessidade de implementação do mesmo após sua aceitação.	Não se aplica.

## RESTRIÇÕES

Restrição	Impacto
A aplicação deverá ser acessada pela internet.	Arquitetura do sistema.
O desenvolvimento do sistema deverá utilizar o paradigma orientado a objetos.	Paradigma de desenvolvimento do sistema.
O sistema deverá utilizar uma linguagem de programação open-source ou de utilização gratuita.	Paradigma de desenvolvimento do sistema.
O resultado da implementação do sistema deve gerar possibilidade de futuras inclusões de novas funcionalidades.	Arquitetura do sistema.
Os subeventos devem ter a característica especial de possuir um desenvolvimento realizado através de plug-ins, pois além dos subeventos inicialmente conhecidos, outros surgirão conforme houver demanda, sempre afetando diversos fatores referentes ao evento em si.	Arquitetura do sistema.

## DEPENDÊNCIAS

Dependência	Descrição
Hospedagem	O sistema executará sobre a web, portanto há a necessidade da hospedagem dos arquivos de sistema em um servidor que ofereça disponibilidade muito próxima a 100% do tempo.

## DESCRIÇÃO GERAL DO SISTEMA

### Introdução

Esta sessão fornece uma descrição geral das características do sistema a ser desenvolvido, facilitando o entendimento dos requisitos a serem definidos nos próximos itens deste documento.

### Macro características do produto

A versão inicial do sistema de controle de eventos será um sistema totalmente novo, não possuindo integração com sistemas legados da UTFPR Câmpus Medianeira. Sua construção utilizará técnicas de desenvolvimento que permitam que ele seja

componentizado, orientado a objetos, executado sobre a web e utilizador de tecnologias gratuitas recentes.

### **Funcionalidades do produto**

- sistema deve manter os dados de evento
- sistema deve manter os dados de subevento
- sistema deve manter os dados de área
- sistema deve manter os dados de instituição
- sistema deve manter os dados de local de realização
- sistema deve manter os dados de sala
- sistema deve manter os dados de palestrante
- sistema deve manter os dados de usuário
- sistema deve manter os dados de patrocinador
- sistema deve manter os dados de atividade
- sistema deve manter os dados de receita/despesa de evento
- sistema deve permitir gerenciar grupos de usuários e permissões
- sistema deve permitir realizar inscrição em evento
- sistema deve permitir escolher participação em subevento
- sistema deve permitir gerar boleto de inscrição em evento
- sistema deve permitir confirmar inscrição
- sistema deve permitir realizar aprovação financeira do evento
- sistema deve permitir imprimir crachás
- sistema deve permitir lançar presença
- sistema deve permitir visualizar site de evento
- sistema deve permitir montar programação
- sistema deve permitir gerar projeto de evento
- sistema deve permitir gerar relatório de receitas e despesas de evento
- sistema deve permitir gerar certificados e declarações

### **Ambiente de operação**

O sistema deverá ser desenvolvido utilizando linguagem e ferramental open-source ou gratuitos de mercado, que devem possuir suporte multiplataforma, de tal forma que permitam a instalação da aplicação em qualquer tipo de servidor e sistema operacional.

### **Características dos usuários**

Os usuários do sistema de controle de eventos estão agrupados, basicamente, em dois grupos:

- Funcionários e alunos da UTFPR Câmpus Medianeira, que possuirão acesso a determinadas áreas administrativas do sistema;
- Participantes e interessados em eventos, que podem ser qualquer pessoa, inclusive os envolvidos no primeiro grupo.

## REQUISITOS FUNCIONAIS

<b>REF0001</b>	<b>O sistema deve manter os dados de evento</b>		
<b>Função</b>	Inserir, alterar e apagar evento.		
<b>Entradas</b>	Dados de evento.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais do evento mantidos.		
<b>Destino das saídas</b>	Cadastro de eventos.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de dados cadastrais do evento que coordena, ou um usuário do tipo administrador dirige-se até a opção de manutenção de eventos, e realiza a manutenção de um evento.		
<b>Pré-condições</b>	Nenhuma.		
<b>Pós-condições</b>	Dados cadastrais do evento mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b>			
<b>Caso de Uso</b>	<b>Manter evento</b>		
<b>Ator Principal</b>	<b>Coordenador do evento, Administrador</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O usuário acessa a opção de manutenção de evento</li> <li>2. O sistema mostra os eventos disponíveis</li> <li>3. O usuário escolhe o evento</li> <li>4. O usuário mantém o evento</li> </ol>			
<b>Extensões</b>			
<ol style="list-style-type: none"> <li>2a. O usuário é coordenador do evento <ol style="list-style-type: none"> <li>1. Não mostra opção</li> <li>2. Pula para o passo 4</li> </ol> </li> <li>3a. O usuário deseja incluir um novo evento <ol style="list-style-type: none"> <li>1. Pula para o passo 4</li> </ol> </li> <li>4a. O usuário é coordenador do evento</li> </ol>			

<b>REF0001</b>	<b>O sistema deve manter os dados de evento</b>
1. O sistema disponibiliza apenas a opção de alteração do evento	

<b>REF002</b>	<b>O sistema deve manter os dados de subevento</b>		
<b>Função</b>	Inserir, alterar e apagar subevento.		
<b>Entradas</b>	Dados de subevento.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais do subevento mantidos.		
<b>Destino das saídas</b>	Cadastro de subeventos.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de subeventos do evento que coordena e realiza a manutenção de um subevento.		
<b>Pré-condições</b>	O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Dados cadastrais do subevento mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<p><b>Observações:</b></p> <p>Um subevento é uma atividade coordenada que ocorre sempre relacionada a determinado evento. Ele envolve participantes, explicitamente inscritos ou não nesta, e pessoas com função de coordenação na mesma.</p> <p>Um evento só existe por causa de seus subeventos.</p> <p>Exemplos de subeventos:</p> <ul style="list-style-type: none"> <li>• Minicurso</li> <li>• Palestra</li> <li>• Visita técnica</li> <li>• Envio de trabalhos</li> <li>• Mostra de trabalhos</li> </ul>			
<b>Caso de Uso</b>	<b>Manter subevento</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O coordenador do evento acessa a opção de manutenção do subevento</li> <li>2. O sistema mostra os subeventos disponíveis</li> <li>3. O usuário escolhe o subevento</li> <li>4. O usuário mantém o subevento</li> </ol>			

<b>REF002</b>	<b>O sistema deve manter os dados de subevento</b>
<b>Extensões</b>	
3a. O usuário deseja incluir um novo evento	
2. Pula para o passo 4	



<b>REF003</b>	<b>O sistema deve manter os dados de área</b>		
<b>Função</b>	Inserir, alterar e apagar área.		
<b>Entradas</b>	Dados de área.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais da área mantidos.		
<b>Destino das saídas</b>	Cadastro de áreas.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de áreas do evento que coordena e realiza a manutenção de uma área.		
<b>Pré-condições</b>	O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Dados cadastrais da área mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b> Uma área é um agrupamento por afinidade de conteúdo que é aplicado a eventos e subeventos deste.			
<b>Caso de Uso</b>	<b>Manter área</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
Cenários Implícitos			

<b>REF004</b>	<b>O sistema deve manter os dados de instituição</b>		
<b>Função</b>	Inserir, alterar e apagar instituição.		
<b>Entradas</b>	Dados de instituição.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais da instituição mantidos.		
<b>Destino das saídas</b>	Cadastro de instituições.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de instituições do evento que coordena, ou um usuário do tipo administrador dirige-se até a opção de manutenção de instituições, e realiza a manutenção de uma instituição.		
<b>Pré-condições</b>	Nenhuma.		
<b>Pós-condições</b>	Dados cadastrais da instituição mantidos.		
<b>Efeitos colaterais</b>	A incorreta informação dos dados cadastrais de uma instituição pode acarretar em problemas legais.		
<b>Responsáveis</b>	Interessado inicial.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Uma instituição é qualquer entidade que se deseja anexar a um evento, podendo ser a organizadora do evento ou apenas uma parceira na realização deste mesmo evento.			
<b>Caso de Uso</b>	<b>Manter instituição</b>		
<b>Ator Principal</b>	<b>Coordenador do evento, Administrador</b>		
Cenários Implícitos			

<b>REF005</b>	<b>O sistema deve manter os dados de local de realização</b>		
<b>Função</b>	Inserir, alterar e apagar local de realização.		
<b>Entradas</b>	Dados de local de realização.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais do local de realização mantidos.		
<b>Destino das saídas</b>	Cadastro de locais de realização.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de locais de realização do evento que coordena e realiza a manutenção de um local de realização, tornando este local público a todos os eventos de uma mesma instituição.		
<b>Pré-condições</b>	O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Dados cadastrais do local de realização mantidos.		
<b>Efeitos colaterais</b>	A incorreta informação dos dados cadastrais de um local de realização pode fazer com que participantes do evento não se dirijam ou não encontrem o local correto.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Um local de realização é o local físico onde ocorrerá a realização de um ou mais subeventos de um evento.			
<b>Caso de Uso</b>	<b>Manter local de realização</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
Cenários Implícitos			

<b>REF006</b>	<b>O sistema deve manter os dados de sala</b>		
<b>Função</b>	Inserir, alterar e apagar sala.		
<b>Entradas</b>	Dados de sala.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais da sala mantidos.		
<b>Destino das saídas</b>	Cadastro de salas.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de salas de locais de realização de evento que coordena e realiza a manutenção de uma sala, tornando esta sala pública a todos os eventos de uma mesma instituição.		
<b>Pré-condições</b>	O local de realização deve estar cadastrado.		
<b>Pós-condições</b>	Dados cadastrais da sala mantidos.		
<b>Efeitos colaterais</b>	A incorreta informação dos dados cadastrais de uma sala pode fazer com que participantes do evento não se dirijam ou não encontrem o local correto.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Uma sala é um local físico interno a um local de realização.			
<b>Caso de Uso</b>	<b>Manter sala</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
Cenários Implícitos			

<b>REF007</b>	<b>O sistema deve manter os dados de palestrante</b>		
<b>Função</b>	Inserir, alterar e apagar palestrante.		
<b>Entradas</b>	Dados de palestrante.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais do palestrante mantidos.		
<b>Destino das saídas</b>	Cadastro de palestrantes.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de palestrantes de evento que coordena e realiza a manutenção de um palestrante, tornando este palestrante público a todos os eventos de uma mesma instituição.		
<b>Pré-condições</b>	O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Dados cadastrais do palestrante mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b> Um palestrante é qualquer pessoa que atuará como disseminador de conhecimento em determinado subevento de qualquer natureza (e não necessariamente somente uma palestra), podendo haver mais de um por subevento.			
<b>Caso de Uso</b>	<b>Manter palestrante</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
Cenários Implícitos			

<b>REF008</b>	<b>O sistema deve manter os dados de usuário</b>		
<b>Função</b>	Inserir, alterar e apagar usuário.		
<b>Entradas</b>	Dados de usuário.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais do usuário mantidos.		
<b>Destino das saídas</b>	Cadastro de usuários.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de usuários que estão ligados de evento que coordena, ou um usuário do tipo administrador dirige-se até a opção de manutenção de usuários do sistema, ou um participante dirige-se até a opção de manutenção de seus próprios dados cadastrais, e realiza a manutenção dos dados cadastrais de usuário.		
<b>Requisitos</b>	O usuário que vai alterar deve conhecer os tipos de usuários e as limitações a ele impostas.		
<b>Pré-condições</b>	Nenhuma.		
<b>Pós-condições</b>	Dados cadastrais do usuário mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Um usuário é qualquer pessoa que venha a estar cadastrado e ter acesso ao sistema, podendo ser de diversos tipos, sendo os inicialmente conhecidos: <ul style="list-style-type: none"> <li>• Participante</li> <li>• Coordenador de evento</li> <li>• Financeiro</li> <li>• Administrador</li> <li>• Comissão organizadora</li> </ul>			
<b>Caso de Uso</b>	<b>Manter usuário</b>		
<b>Ator Principal</b>	<b>Coordenador do evento, Administrador, Usuário</b>		
Cenários Implícitos			

<b>REF009</b>	<b>O sistema deve manter os dados de patrocinador</b>		
<b>Função</b>	Inserir, alterar e apagar patrocinador.		
<b>Entradas</b>	Dados de patrocinador.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais do patrocinador mantidos.		
<b>Destino das saídas</b>	Cadastro de patrocinadores.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de patrocinadores de evento que coordena e realiza a manutenção de um patrocinador.		
<b>Pré-condições</b>	O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Dados cadastrais do patrocinador mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b> Um patrocinador é uma pessoa, empresa ou entidade que está disposta a colaborar de alguma forma (normalmente financeira) com a realização do evento.			
<b>Caso de Uso</b>	<b>Manter patrocinador</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
Cenários Implícitos			

<b>REF010</b>	<b>O sistema deve manter os dados de atividade</b>		
<b>Função</b>	Inserir, alterar e apagar atividade.		
<b>Entradas</b>	Dados de atividade.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Dados cadastrais da atividade mantidos.		
<b>Destino das saídas</b>	Cadastro de atividades.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de atividades de evento que coordena e realiza a manutenção de uma atividade.		
<b>Requisitos</b>	Nenhum.		
<b>Pré-condições</b>	O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Dados cadastrais da atividade mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b> Uma atividade é um serviço ligado ao evento que o coordenador de um evento faz ou designa alguém para fazer.			
<b>Caso de Uso</b>	<b>Manter atividade</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
Cenários Implícitos			



<b>REF011</b>	<b>O sistema deve manter os dados de receita/despesa de evento</b>		
<b>Função</b>	Inserir, alterar e apagar receita ou despesa de evento.		
<b>Entradas</b>	Dados de receita ou despesa de evento.		
<b>Fonte das entradas</b>	Informação por parte do usuário. Geração automatizada por outras rotinas do sistema.		
<b>Saídas</b>	Dados cadastrais da receita ou despesa de evento mantidos.		
<b>Destino das saídas</b>	Cadastro de receitas ou despesas de evento.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de manutenção de receitas ou despesas de evento que coordena e realiza a manutenção de uma receita ou despesa de evento ou há uma geração automatizada de receita ou despesa de evento através da realização de outra rotina do sistema.		
<b>Pré-condições</b>	O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Dados cadastrais da receita ou despesa de evento mantidos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Média	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Uma receita ou despesa de evento é um valor financeiro que, respectivamente, decresce ou acresce o saldo final do projeto. Uma receita ou despesa de evento pode ser diretamente mantida em tela específica para tal ou pode ser gerada automaticamente por outras rotinas do sistema, o que acabará por impossibilitar a citada manutenção manual direta desta última receita ou despesa de evento gerada de forma automatizada. Exemplos de formas de geração automatizada de receita ou despesa de evento são: <ul style="list-style-type: none"> <li>• Pagamento de valor por parte de patrocinador, que gerará uma receita;</li> <li>• Pagamento de despesa com palestrante, que gerará uma despesa;</li> <li>• Pagamento de inscrição por parte de participante, que gerará uma receita.</li> </ul>			
<b>Caso de Uso</b>	<b>Manter receita/despesa de evento</b>		
<b>Ator Principal</b>	<b>Coordenador do evento, Sistema</b>		
Cenários Implícitos			

<b>REF012</b>	<b>O sistema deve permitir gerenciar grupos de usuários e permissões</b>		
<b>Função</b>	Permitir ao administrador do sistema e ao coordenador de evento manipular as informações de tipos de usuários e permissões destes.		
<b>Entradas</b>	Usuário a ser manipulado.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Usuário com permissões gerenciadas.		
<b>Destino das saídas</b>	Cadastro de usuários.		
<b>Ação</b>	Um usuário do tipo coordenador de evento dirige-se até a opção de gerenciar grupos de usuários e permissões de evento que coordena, ou um usuário do tipo administrador dirige-se até a opção de gerenciar grupos de usuários e permissões, e realiza a gerência dos grupos, manipulando as permissões do usuário para que se adeque às suas reais responsabilidades.		
<b>Requisitos</b>	O usuário que vai alterar deve conhecer os tipos de usuários e as limitações a ele impostas.		
<b>Pré-condições</b>	O usuário deve estar cadastrado. O evento deve estar cadastrado. O evento deve possuir um coordenador.		
<b>Pós-condições</b>	Permissões de usuário gerenciadas.		
<b>Efeitos colaterais</b>	Uma permissão mal gerenciada pode acarretar em atribuição incorreta de responsabilidades ao usuário.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Média	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Neste processo o coordenador do evento manipula os usuários, de qualquer tipo, ligados ao mesmo, aplicando a um ou mais usuários, um ou mais privilégio de acesso (responsabilidade) dentro do evento. Os privilégios inicialmente conhecidos são os ligados a tipos de usuários, porém com nível de privilégios inferior ou igual ao do próprio coordenador: <ul style="list-style-type: none"> <li>• Participante</li> <li>• Coordenador de evento</li> <li>• Comissão organizadora</li> </ul>			
<b>Caso de Uso</b>	<b>Realizar inscrição em evento</b>		
<b>Ator Principal</b>	<b>Visitante</b>		
<b>Cenário principal</b>			
8. O usuário acessa a opção de gerenciamento de grupos de usuários e permissões			

<b>REF012</b>	<b>O sistema deve permitir gerenciar grupos de usuários e permissões</b>
9. O sistema mostra os usuários disponíveis 10. O usuário escolhe o usuário 11. O sistema mostra as opções de permissões que podem ser para um usuário 12. O usuário escolhe as permissões e confirma 13. O sistema grava as permissões para o usuário	
<b>Extensões</b>	

<b>REF013</b>	<b>O sistema deve permitir realizar inscrição em evento</b>		
<b>Função</b>	Permitir a um visitante do site personalizado de um evento cadastrar-se, informando seus dados pessoais, com o objetivo de vincular seu nome aos participantes deste evento.		
<b>Entradas</b>	Dados cadastrais do participante.		
<b>Fonte das entradas</b>	Informação por parte do usuário visitante.		
<b>Saídas</b>	Participante inscrito no evento, mas ainda não confirmado. Valor da inscrição-base calculado.		
<b>Destino das saídas</b>	Cadastro de usuários. Inscrições do evento.		
<b>Ação</b>	Um usuário do tipo visitante pode acessar a opção fornecida para a realização da inscrição no evento. Preenche, então, seus dados pessoais e outros dados cabíveis em um formulário onde o campo identificador é o CPF da pessoa, possibilitando o reaproveitamento de dados pessoais de uma inscrição realizada para outro evento.		
<b>Pré-condições</b>	O usuário deve ter acessado o site personalizado do evento. O usuário deve conhecer os dados pessoais que serão dele cobrados.		
<b>Pós-condições</b>	O usuário passa a ser conhecido como participante. O usuário está cadastrado e inscrito no evento, mas ainda não está confirmado. O usuário pode acessar o painel do participante e realizar as ações a ele permitidas.		
<b>Efeitos colaterais</b>	A inscrição não confirmada até data especificada acarreta na perda da vaga no evento.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Alto
<b>Observações:</b> Este requisito possui estabilidade baixa, pois possui alta volatilidade, estando diretamente ligado aos subeventos, que são objetos mutáveis e não totalmente conhecidos inicialmente. Mesmo com estabilidade alta, o risco deste requisito pode ser considerado alto, pois depende da manipulação do sistema por parte de um usuário não treinado para esta função, o que pode acarretar na invalidação de sua inscrição.			
<b>Caso de Uso</b>	<b>Realizar inscrição em evento</b>		
<b>Ator Principal</b>	<b>Visitante</b>		
<b>Cenário principal</b>			
14. O visitante seleciona a opção de inscrição			

<b>REF013</b>	<b>O sistema deve permitir realizar inscrição em evento</b>
15. O sistema solicita o CPF do visitante 16. O visitante informa o CPF 17. O sistema solicita os dados cadastrais do visitante 18. O visitante informa seus dados cadastrais 19. O visitante grava a inserção dos dados cadastrais 20. O usuário passa a ser participante não confirmado do evento	
<b>Extensões</b>	
3a. Já existe um usuário cadastrado com este CPF 4. Sistema solicita a senha do usuário 5. Usuário informa a senha Senha inválida Cancela o caso de uso 6. Pular para o passo 7	

<b>REF014</b>	<b>O sistema deve permitir escolher participação em subevento</b>		
<b>Função</b>	Permitir a participantes inscritos em eventos realizar inscrição em subevento que deseja participar.		
<b>Entradas</b>	Subevento que o participante deseja participar.		
<b>Fonte das entradas</b>	Informação por parte do usuário participante.		
<b>Saídas</b>	Participante inscrito no subevento, mas ainda não confirmado. Valor total da inscrição no evento recalculado.		
<b>Destino das saídas</b>	Participação em subevento.		
<b>Ação</b>	Um usuário participante de um evento, com inscrição ainda não confirmada, seleciona um subevento do evento em questão que deseja participar, incluindo sua participação no mesmo.		
<b>Pré-condições</b>	O usuário deve estar inscrito no evento. O participante não pode estar com sua inscrição confirmada.		
<b>Pós-condições</b>	O usuário participante tem seu cadastro incluído nos participantes do subevento, mas ainda não está confirmado.		
<b>Efeitos colaterais</b>	A inscrição não confirmada até data especificada acarreta na perda da vaga no subevento.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Baixa	<b>Situação</b> Proposto	<b>Risco</b> Alto
<b>Observações:</b> Este requisito possui estabilidade baixa, pois possui alta volatilidade, estando diretamente ligado aos subeventos, que são objetos mutáveis e não totalmente conhecidos inicialmente. Seu risco é alto, pois depende da manipulação do sistema por parte de um usuário não treinado para esta função, o que pode acarretar na invalidação de sua participação.			
<b>Caso de Uso</b>	<b>Escolher subevento</b>		
<b>Ator Principal</b>	<b>Participante</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O participante seleciona a opção de escolha de subeventos</li> <li>2. O sistema mostra a listagem dos subeventos que possibilitam inscrição e ainda possuem vagas</li> <li>3. O participante escolhe um subevento para participar</li> <li>4. O sistema grava a escolha do participante</li> </ol>			
<b>Extensões</b>			
3a. Subevento realizado ao mesmo tempo em que outro já selecionado			

<b>REF014</b>	<b>O sistema deve permitir escolher participação em subevento</b>
1. Cancela a execução do caso de uso	

<b>REF015</b>	<b>O sistema deve permitir gerar boleto de inscrição em evento</b>		
<b>Função</b>	Permitir a participantes inscritos, mas não confirmados, a geração do boleto bancário, de valor sumarizado de acordo com sua inscrição.		
<b>Entradas</b>	Nenhuma.		
<b>Fonte das entradas</b>	Nenhuma.		
<b>Saídas</b>	Boleto.		
<b>Destino das saídas</b>	Tela.		
<b>Ação</b>	Um usuário participante de um evento, com inscrição ainda não confirmada e subeventos já selecionados, seleciona a opção de geração de boleto para pagamento da inscrição no evento.		
<b>Pré-condições</b>	O usuário deve estar inscrito no evento. O usuário deve ter selecionado os subeventos que deseja participar (caso haja necessidade e possibilidade).		
<b>Pós-condições</b>	O usuário participante tem seu cadastro incluído nos participantes do subevento, mas ainda não está confirmado.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Este requisito possui risco médio, pois pode haver dependência de outros sistemas ou processos externos ao ambiente do sistema, como, por exemplo, os pertencerem ao sistema bancário.			
<b>Caso de Uso</b>	<b>Gerar boleto</b>		
<b>Ator Principal</b>	<b>Participante</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O participante seleciona a opção de geração de boletos</li> <li>2. O sistema mostra o boleto com o valor calculado de acordo com o evento e subeventos selecionados</li> </ol>			
<b>Extensões</b>			



<b>REF016</b>	<b>O sistema deve permitir confirmar inscrição</b>		
<b>Função</b>	Permitir ao usuário financeiro ou ao coordenador do evento confirmar a inscrição de participante.		
<b>Entradas</b>	Usuário a ter inscrição confirmada.		
<b>Fonte das entradas</b>	Informação por parte do usuário que está confirmando.		
<b>Saídas</b>	Participante com inscrição confirmada no evento. Participante com participação garantida em subeventos.		
<b>Destino das saídas</b>	Inscrições do evento. Participação em subevento.		
<b>Ação</b>	O usuário financeiro garante a confirmação do pagamento de inscrição por um participante, ou o coordenador do evento não exige pré-requisitos, e informa que o participante está com inscrição confirmada no evento.		
<b>Pré-condições</b>	O usuário deve ter pago sua inscrição no evento ou o coordenador do evento pode, por qualquer motivo, dar o participante como apto a ser confirmado.		
<b>Pós-condições</b>	O usuário participante tem sua inscrição confirmada no evento. O usuário participante tem sua participação garantida em subeventos.		
<b>Efeitos colaterais</b>	A demora na confirmação da aptidão de confirmação pode acarretar na perda da inscrição no evento ou vaga em subevento.		
<b>Responsáveis</b>	Interessado financeiro, Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Este requisito possui risco médio, pois pode haver dependência de outros sistemas ou processos externos ao ambiente do sistema.			
<b>Caso de Uso</b>	<b>Confirmar inscrição</b>		
<b>Ator Principal</b>	<b>Financeiro, Coordenador de evento</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção de confirmação de inscrições</li> <li>2. O sistema mostra a listagem de participantes não confirmados</li> <li>3. O usuário seleciona um participante</li> <li>4. O sistema pede se o participante pagou ou teve isenção de inscrição</li> <li>5. O usuário confirma a inscrição do participante</li> <li>6. O sistema confirma a inscrição no evento</li> <li>7. O sistema confirma a participação em subevento</li> </ol>			

<b>REF016</b>	<b>O sistema deve permitir confirmar inscrição</b>
<b>Extensões</b>	
6a. O número de vagas confirmadas já foi excedido	
1. Agendar reembolso do valor pago	
2. Cancelar caso de uso	
6a. O número de vagas confirmadas já foi excedido	
1. Agendar reembolso da diferença de valor	

<b>REF017</b>	<b>O sistema deve permitir realizar aprovação financeira do evento</b>		
<b>Função</b>	Permitir ao usuário financeiro informar que o evento foi aprovado após análise dos dados financeiros deste.		
<b>Entradas</b>	Projeto do evento.		
<b>Fonte das entradas</b>	Entregue pelo coordenador do evento.		
<b>Saídas</b>	Evento com aprovação financeira.		
<b>Destino das saídas</b>	Cadastro de evento.		
<b>Ação</b>	O usuário financeiro analisa o projeto do evento entregue a ele pelo coordenador do evento, verifica a viabilidade financeira do evento e o define como aprovado para execução.		
<b>Pré-condições</b>	O usuário coordenador do evento deve ter gerado o projeto do evento. O coordenador do evento deve ter entregue o projeto do evento para o financeiro.		
<b>Pós-condições</b>	O evento com aprovação financeira obtida.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado financeiro.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b>			
<b>Caso de Uso</b>	<b>Realizar aprovação financeira do evento</b>		
<b>Ator Principal</b>	<b>Financeiro</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O financeiro acessa a opção de aprovação financeiro do evento</li> <li>2. O financeiro avalia o projeto do evento recebido</li> <li>3. O sistema solicita a informação de dados sobre a aprovação</li> <li>4. O financeiro confirma a aprovação financeira do evento</li> </ol>			
<b>Extensões</b>			
<p>2a. O financeiro reprova o orçamento do evento</p> <ol style="list-style-type: none"> <li>1. O sistema solicita o preenchimento do motivo</li> <li>2. A execução se encerra</li> </ol>			

<b>REF018</b>	<b>O sistema deve permitir imprimir crachás</b>		
<b>Função</b>	Permitir ao usuário da comissão organizadora com designação para tal função imprimir os crachás de identificação de participantes do evento.		
<b>Entradas</b>	Nenhuma.		
<b>Fonte das entradas</b>	Nenhuma.		
<b>Saídas</b>	Dados dos crachás de identificação.		
<b>Destino das saídas</b>	Impressora.		
<b>Ação</b>	O usuário da comissão organizadora, que possui permissão para impressão de crachás para o evento, acessa a opção de impressão de crachás no sistema e imprime.		
<b>Pré-condições</b>	O prazo para as inscrições já deve ser finalizado. O usuário da comissão organizadora deve possuir permissão para impressão de crachás.		
<b>Pós-condições</b>	Crachás de identificação de participantes impressos.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Média	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b>			
No crachá deve haver, dentre outras informações, um código de barras, ou recurso semelhante, que possibilite a leitura automatizada do mesmo.			
Estabilidade classificada como média, pois pode haver a necessidade de adequação da impressão para casos inicialmente não previstos, como outros formatos de crachá ou modelos de impressora.			
Risco classificado como médio, pois para ter sucesso depende de ações externas ao sistema realizadas por usuários possivelmente mais leigos e destreinados que os demais.			
<b>Caso de Uso</b>	<b>Imprimir crachás</b>		
<b>Ator Principal</b>	<b>Comissão organizadora</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O usuário da comissão organizadora acessa a opção de impressão de crachás</li> <li>2. O sistema mostra uma prévia dos crachás</li> <li>3. O usuário da comissão organizadora imprime os crachás</li> </ol>			
<b>Extensões</b>			

<b>REF019</b>	<b>O sistema deve permitir lançar presença</b>		
<b>Função</b>	Permitir ao usuário da comissão organizadora com designação para tal função informar ao sistema que um participante esteve presente em um subevento.		
<b>Entradas</b>	Código de barras do crachá.		
<b>Fonte das entradas</b>	Leitor de código de barras. Informação manual do número do código de barras do crachá por parte do usuário da comissão organizadora.		
<b>Saídas</b>	Participante com presença em subevento confirmada.		
<b>Destino das saídas</b>	Participação em subevento.		
<b>Ação</b>	O usuário da comissão organizadora, que possui permissão para lançamento de presenças para o evento, acessa a opção de lançamento de presenças no sistema e utiliza o leitor de código de barras, ou digita o número do código de barras manualmente, para informar ao sistema que o participante cujo crachá está sendo informado está presente no subevento.		
<b>Pré-condições</b>	O participante está configurado como não presente no subevento. O usuário da comissão organizadora deve possuir permissão para lançamento de presenças.		
<b>Pós-condições</b>	O participante está configurado como presente no subevento.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Média	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Estabilidade classificada como média, pois pode haver a necessidade de adequação a diferentes formatos ou equipamentos de leitura dos crachás de identificação dos participantes. Risco classificada como médio, pois para ter sucesso depende de ações externas ao sistema realizadas por usuários possivelmente mais leigos e destreinados que os demais.			
<b>Caso de Uso</b>	<b>Lançar presença</b>		
<b>Ator Principal</b>	<b>Comissão organizadora</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O usuário da comissão organizadora acessa a opção de lançamento de presença</li> <li>2. O sistema mostra os subeventos liberados para lançamento de presença</li> <li>3. O usuário da comissão organizadora escolhe o subevento desejado</li> <li>4. O sistema permite a leitura de crachás utilizando leitor de código de barras</li> <li>5. O usuário faz a leitura com o leitor de código de barras</li> </ol>			

<b>REF019</b>	<b>O sistema deve permitir lançar presença</b>
6. O sistema confirma a presença do participante	
<b>Extensões</b>	
5a. Sem leitor de código de barras disponível	
1. O sistema permite a digitação manual do código de barras	
ba. O subevento necessitava confirmação de participação e o participante não a fez	
1. O sistema mostra uma mensagem de participante não inscrito	
2. Cancela o passo	
3. Volta para o item 4	

<b>REF020</b>	<b>O sistema deve permitir visualizar site de evento</b>		
<b>Função</b>	Permitir ao usuário visitante visualizar os dados públicos do evento na página personalizada do evento.		
<b>Entradas</b>	Nenhuma.		
<b>Fonte das entradas</b>	Nenhuma.		
<b>Saídas</b>	Nenhuma.		
<b>Destino das saídas</b>	Nenhum.		
<b>Ação</b>	O usuário visitante digita o endereço do site personalizado do evento no navegador, entra no site e pode visualizar os dados públicos do evento.		
<b>Requisitos</b>	O visitante precisa possuir internet e um navegador.		
<b>Pré-condições</b>	Nenhuma.		
<b>Pós-condições</b>	Site personalizado do evento visualizado.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b>			
<b>Caso de Uso</b>	<b>Visualizar site de evento</b>		
<b>Ator Principal</b>	<b>Visitante</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O visitante digita o endereço da página do evento em um navegador web</li> <li>2. O navegador web mostra o a página do evento para o usuário</li> </ol>			
<b>Extensões</b>			

<b>REF021</b>	<b>O sistema deve permitir montar programação</b>		
<b>Função</b>	Permitir ao coordenador do evento montar, de forma gráfica, a grade de programação do evento, contendo todos os subeventos e prevenindo a inserção de inconsistências.		
<b>Entradas</b>	Subeventos.		
<b>Fonte das entradas</b>	Sistema.		
<b>Saídas</b>	Grade de programação do evento.		
<b>Destino das saídas</b>	Cadastro de subeventos.		
<b>Ação</b>	O usuário coordenador de evento dirige-se até a opção de montagem da programação do evento, consegue visualizar todos os subeventos e possui uma forma gráfica de montar e visualizar a grade de programação do evento que não permite a inserção de inconsistências.		
<b>Pré-condições</b>	Ter todos os subeventos que se deseja alocar cadastrados. Os subeventos devem estar com suas durações definidas.		
<b>Pós-condições</b>	Grade de programação do evento criada.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Média	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b> Estabilidade classificada como média por não ter seu funcionamento completamente entendido pela equipe.			
<b>Caso de Uso</b>	<b>Montar programação</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O coordenador do evento acessa a opção de montagem da programação do evento</li> <li>2. O sistema mostra uma grade de horários e salas para o coordenador distribuir os subeventos</li> <li>3. O sistema mostra uma listagem com os eventos não alocados</li> <li>4. O coordenador do sistema arrasta um subevento para um espaço não alocado</li> <li>5. O sistema confirma a alocação da sala e horário para aquele subevento</li> </ol>			
<b>Extensões</b>			



<b>REF022</b>	<b>O sistema deve permitir gerar projeto de evento</b>		
<b>Função</b>	Permitir ao coordenador do evento manter as informações referentes ao projeto do evento e gerar a versão impressa do mesmo.		
<b>Entradas</b>	Dados do projeto.		
<b>Fonte das entradas</b>	Informação por parte do usuário coordenador do evento.		
<b>Saídas</b>	Projeto do evento.		
<b>Destino das saídas</b>	Impressora.		
<b>Ação</b>	O usuário coordenador de evento dirige-se até a opção de geração do projeto do evento do evento que coordena e mantém as informações faltantes. Com as informações cadastradas, requisita a geração impressa do projeto, que contém os dados cadastrados para ele, adicionados de dados sumarizados e agrupados obtidos das outras etapas realizadas.		
<b>Pré-condições</b>	Informações iniciais do evento cadastradas.		
<b>Pós-condições</b>	Projeto do evento cadastrado e gerado.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Média	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b> Estabilidade classificada como média por não ter seu funcionamento completamente entendido pela equipe.			
<b>Caso de Uso</b>	<b>Gerar projeto de evento</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O coordenador do evento acessa a opção de geração do projeto do evento</li> <li>2. O sistema requisita a informação de dados referentes ao projeto</li> <li>3. O coordenador do evento informa os dados do projeto</li> <li>4. O coordenador do evento confirma a geração do projeto do evento</li> <li>5. O sistema grava o projeto gerado</li> <li>6. O sistema imprime o projeto do evento</li> </ol>			
<b>Extensões</b>			

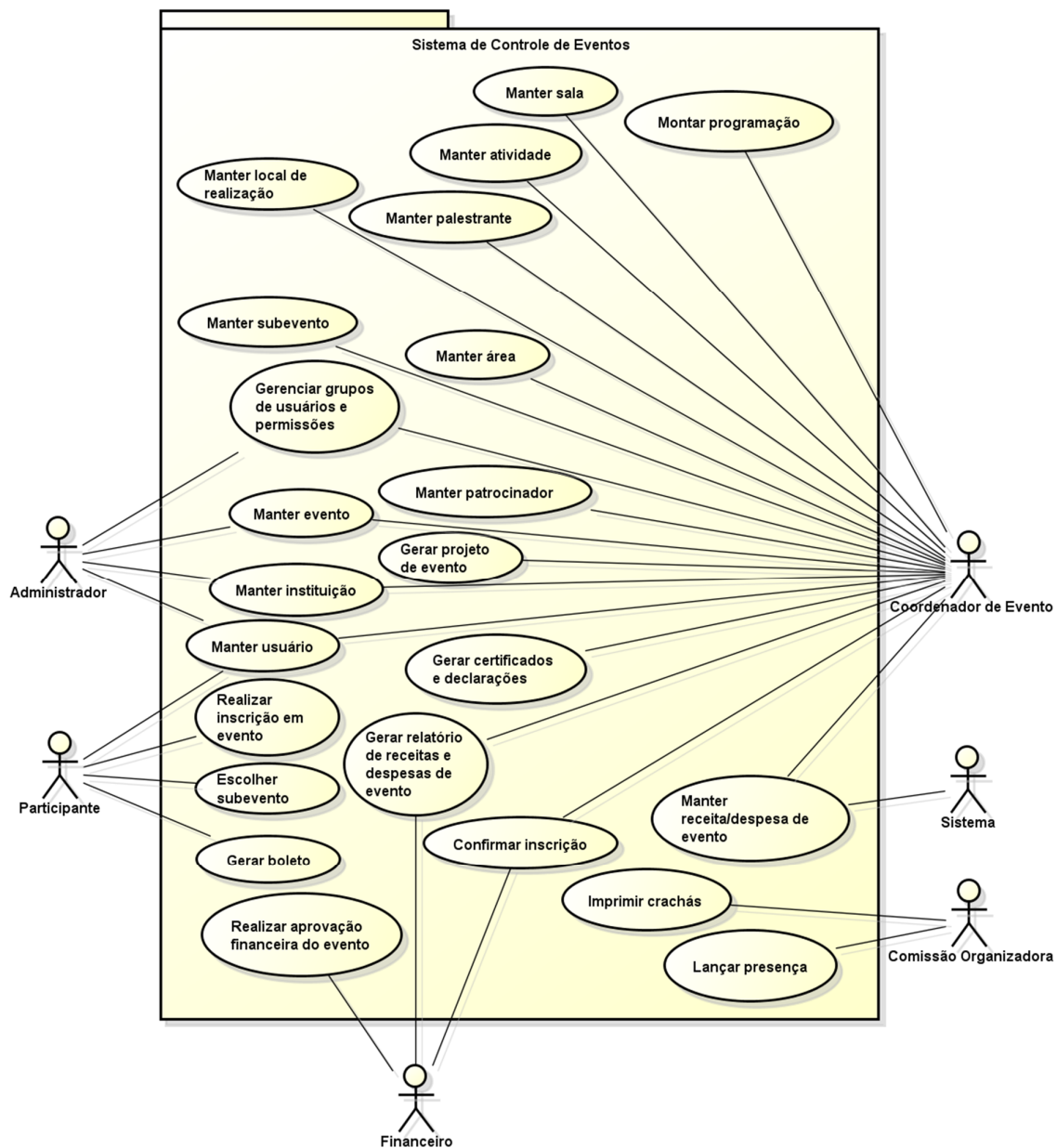
<b>REF023</b>	<b>O sistema deve permitir gerar relatório de receitas e despesas de evento</b>		
<b>Função</b>	Permitir ao coordenador do evento e ao financeiro emitir uma relação de todas as receitas e despesas apontadas para o evento.		
<b>Entradas</b>	Evento.		
<b>Fonte das entradas</b>	Informação por parte do usuário coordenador do evento ou financeiro.		
<b>Saídas</b>	Relatório de receitas e despesas do evento.		
<b>Destino das saídas</b>	Tela ou impressora.		
<b>Ação</b>	O usuário coordenador de evento, ou financeiro, solicita a impressão do relatório de receitas e despesas do evento.		
<b>Pré-condições</b>	Nenhuma.		
<b>Pós-condições</b>	Relatório de receitas e despesas do evento gerado.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial, Interessada em eventos e Interessado financeiro.		
<b>Benefício</b> Importante	<b>Estabilidade</b> Alta	<b>Situação</b> Proposto	<b>Risco</b> Baixo
<b>Observações:</b>			
<b>Caso de Uso</b>	<b>Gerar relatório de receitas e despesas de evento</b>		
<b>Ator Principal</b>	<b>Coordenador do evento, Financeiro</b>		
<b>Cenário principal</b>			
<ol style="list-style-type: none"> <li>1. O usuário acessa a opção de geração do relatório de receitas e despesas de evento</li> <li>2. O sistema imprime o relatório de receitas e despesas de evento</li> </ol>			
<b>Extensões</b>			

<b>REF024</b>	<b>O sistema deve permitir gerar certificados e declarações</b>		
<b>Função</b>	Permitir ao coordenador do evento imprimir os certificados e declarações cabíveis para o evento.		
<b>Entradas</b>	Subevento. Evento.		
<b>Fonte das entradas</b>	Informação por parte do usuário.		
<b>Saídas</b>	Impressão de certificados e declarações.		
<b>Destino das saídas</b>	Impressora.		
<b>Ação</b>	O usuário coordenador de evento solicita a impressão de certificados e declarações para os participantes presentes no evento.		
<b>Requisitos</b>	O participante deve ter estado presente no item cujo certificado será impresso.		
<b>Pré-condições</b>	Nenhuma.		
<b>Pós-condições</b>	Certificados e declarações gerados.		
<b>Efeitos colaterais</b>	Nenhum.		
<b>Responsáveis</b>	Interessado inicial e Interessada em eventos.		
<b>Benefício</b> Crítico	<b>Estabilidade</b> Média	<b>Situação</b> Proposto	<b>Risco</b> Médio
<b>Observações:</b> Estabilidade classificada como média, por estar ligado diretamente ao funcionamento de subeventos. Risco classificada como médio, por ser uma atividade obrigatória que depende de fatores externos e por seu produto de saída possuir valor legal.			
<b>Caso de Uso</b>	<b>Gerar certificados e declarações</b>		
<b>Ator Principal</b>	<b>Coordenador do evento</b>		
<b>Cenário principal</b>			
5. O coordenador do evento acessa a opção de geração de certificados e declarações 6. O sistema pede se a geração é para o evento ou por subevento 7. O coordenador do evento escolhe o tipo desejado 8. O sistema imprime os certificados ou declarações			
<b>Extensões</b>			
3a. O coordenador do evento escolhe evento 1. Passa para o passo 4 normalmente 3b. O coordenador do evento escolhe por subevento			

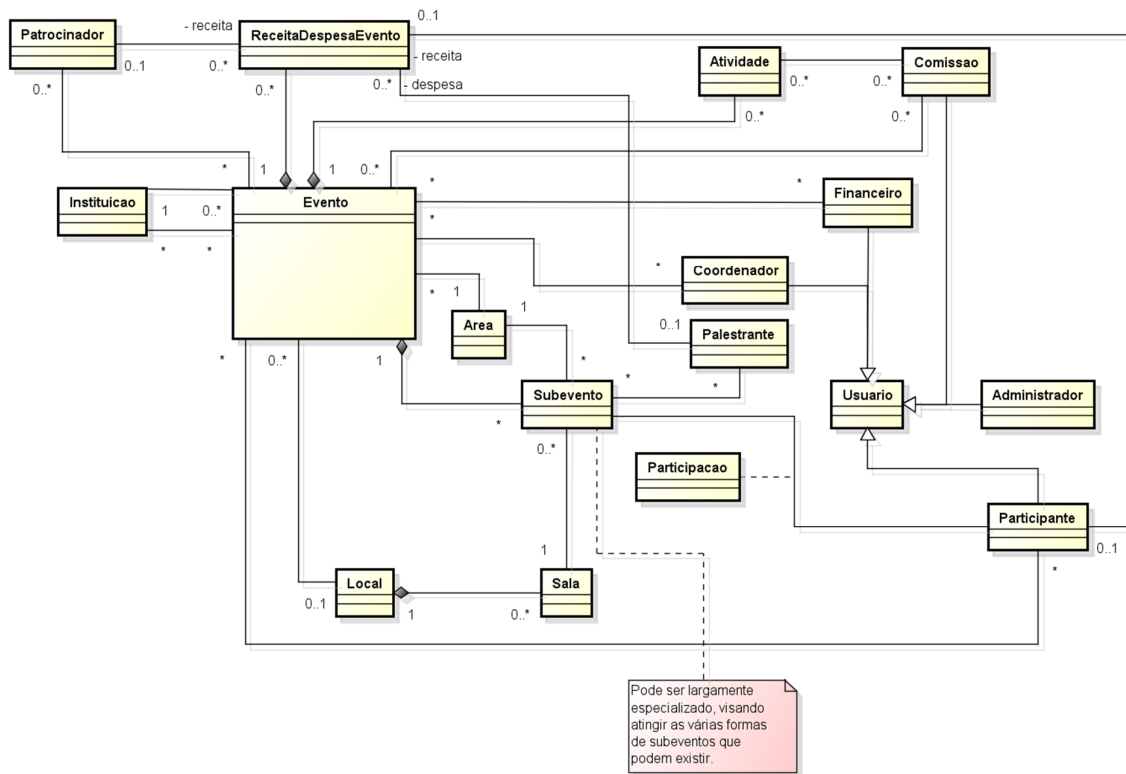
<b>REF024</b>	<b>O sistema deve permitir gerar certificados e declarações</b>
<ol style="list-style-type: none"><li>1. O sistema mostra os subeventos disponíveis</li><li>2. O coordenador do evento escolhe o subevento desejado</li><li>3. Passa para o passo 4 normalmente</li></ol>	

# DIAGRAMAS

## Casos de Uso



## Classes



## REQUISITOS NÃO FUNCIONAIS

	<b>Categoria</b>	<b>Ident.</b>	<b>Descrição</b>
Requisitos de produto	Usabilidade	RNU001	A funcionalidade de subeventos do sistema deve conseguir trabalhar no formato de plugin.
		RNU002	Cada evento deve possuir uma página personalizada.
		RNU003	O visitante da página do evento deve conseguir facilmente realizar as ações básicas dele esperadas.
	Proteção	RNP001	Um usuário pode ser de diversos tipos, podendo ter diversas atribuições dentro do sistema ou de um evento.
		RNP002	Todos os registros gravados devem conter um atributo apontando qual foi o último usuário que o manipulou.
		RNP003	Todas as manipulações de dados do sistema devem ser arquivadas em um sistema de log, apontando o usuário, data, hora, entidade e propriedade manipulada.
	Confiança	RNC001	O sistema deve ficar disposto 100% do tempo.
		RNC002	No caso de sair do ar e haja eventos em processo, o sistema deve ser restabelecido em, no máximo, 20 minutos.
	Espaço	RNE001	Todos os dados cadastrados devem possuir retenção ilimitada, possibilitando a eterna consulta de histórico.
	Requisitos organizacionais	Ambiental	RNA001
RNA002			O servidor de hospedagem deve oferecer possibilidade de envio de e-mail.
RNA003			O servidor de hospedagem deve permitir a instalação de qualquer ferramenta que for solicitada.
RNA004			O sistema deve conseguir ser instalado em mais de um ambiente operacional.
Desenvolvimento		RND001	O sistema deve utilizar linguagem de programação orientada a objetos.
		RND002	O sistema deve utilizar uma linguagem open-source ou gratuita.
		RND003	O sistema deve utilizar linguagem e ferramental multiplataforma.
Requisitos externos	Reguladores	RNR001	O sistema deve seguir a normativa para emissão de declarações e certificados.
	Éticos	RNT001	Os dados cadastrais dos usuários não podem ser divulgados sem a permissão dos mesmos.

## REQUISITOS DO USUÁRIO POR PERFIL

Este item mostra os perfis de usuários previstos para o sistema de controle de eventos, juntamente com as funcionalidades acessíveis a cada tipo.

Nome	Responsabilidades
Visitante	<ul style="list-style-type: none"> <li>• Visualizar dados dos eventos</li> <li>• Visualizar detalhes de subeventos</li> <li>• Realizar inscrição em evento</li> </ul>
Participante	<ul style="list-style-type: none"> <li>• Visualizar dados dos eventos</li> <li>• Visualizar detalhes de subeventos</li> <li>• Gerenciar informações de inscrição em evento</li> <li>• Pagar inscrição em evento</li> </ul>
Coordenador de Evento	<ul style="list-style-type: none"> <li>• Administrar informações cadastrais do evento</li> <li>• Administrar subeventos</li> <li>• Administrar participantes do evento</li> <li>• Administrar palestrantes do evento</li> <li>• Administrar datas e horários do evento</li> <li>• Gerar grade de subeventos</li> <li>• Administrar custos do evento</li> <li>• Criar projeto do evento</li> <li>• Visualizar resumos do evento</li> <li>• Imprimir certificados</li> <li>• Designar funções para o evento</li> <li>• Administrar patrocinadores do evento</li> <li>• Administrar receitas do evento</li> <li>• Administrar comissão organizadora do evento</li> <li>• Administrar locais de realização do evento</li> <li>• Administrar instituições do evento</li> <li>• Administrar atividades do evento</li> </ul>
Financeiro	<ul style="list-style-type: none"> <li>• Confirmar inscrições</li> <li>• Visualizar pagamentos de inscrições</li> </ul>
Administrador	<ul style="list-style-type: none"> <li>• Cadastrar novo evento</li> <li>• Administrar usuários</li> <li>• Designar funções e permissões a outros usuários</li> <li>• Administrar características gerais do sistema</li> <li>• Administrar tipos de subeventos</li> </ul>



<b>Nome</b>	<b>Responsabilidades</b>
Comissão Organizadora	<ul style="list-style-type: none"><li>• Emitir de crachás</li><li>• Conferir presenças</li></ul>