

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO DE PÓS-GRADUAÇÃO  
ESPECIALIZAÇÃO EM DESENVOLVIMENTO WEB**

**RENIVALDO PEREIRA DA COSTA**

**APLICANDO A ENGENHARIA DE REQUISITOS COM O PROCESSO  
UNIFICADO - UM ESTUDO DE CASO PARA SISTEMAS WEB**

**MONOGRAFIA DE ESPECIALIZAÇÃO**

**LONDRINA**

**2013**

**RENIVALDO PEREIRA DA COSTA**

**APLICANDO A ENGENHARIA DE REQUISITOS COM O PROCESSO  
UNIFICADO - UM ESTUDO DE CASO PARA SISTEMAS WEB**

Monografia apresentada ao Programa de Pós-Graduação em Desenvolvimento Web, da Universidade Tecnológica Federal do Paraná, campus Londrina, como requisito parcial para obtenção do grau de Especialista em Desenvolvimento Web. Área de Concentração: Sistemas de Informação.

Orientador: Prof. Dr. Elias Canhadas Genvigir

Coorientador: Profa. Me. Letícia Jovelina Storto

**LONDRINA**

**2013**

Folha destinada à inclusão da **Ficha Catalográfica** (elemento obrigatório somente para as dissertações) a ser solicitada ao Departamento de Biblioteca do Campus UTFPR (prazo: 3 dias) e posteriormente impressa no verso da Folha de Rosto (folha anterior).



## TERMO DE APROVAÇÃO

Título da Monografia

**APLICANDO A ENGENHARIA DE REQUISITOS COM O PROCESSO UNIFICADO  
- UM ESTUDO DE CASO PARA SISTEMAS WEB**

por

**RENIVALDO PEREIRA DA COSTA**

Esta monografia foi apresentada às 16h00 do dia **01** de **fevereiro** de 2013 como requisito parcial para a obtenção do título de ESPECIALISTA EM DESENVOLVIMENTO WEB. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho \_\_\_\_\_.

(aprovado, aprovado com restrições ou reprovado)

---

Prof. Elias Canhadas Genvigir  
(UTFPR)

---

Prof. André Luis dos Santos  
Domingues  
(UTFPR)

---

Prof. Guilherme Luiz Frufrek  
(UTFPR)

Visto da coordenação:

---

Prof. Thiago Prado de Campos  
Coordenador da esp. em Desenvolvimento Web

---

Prof. Walmir Eno Pottker  
Coordenador de Pós-Graduação Lato Senso

À minha família!

## **AGRADECIMENTOS**

Agradeço a Deus pela força para vencer as barreiras que apareceram durante a caminhada até a concretização de mais esta etapa em minha vida.

Ao Professor Dr. Elias Canhadas Genvigir pela dedicação e pela orientação deste trabalho. Por meio dele, reporto-me a toda a comunidade da Universidade Tecnológica Federal do Paraná (UTFPR) pelo apoio incondicional.

À Professora Me. Letícia Jovelina Storto pela orientação na estruturação desta pesquisa e pelos momentos de aprendizado.

Aos pesquisadores e professores da banca examinadora pela atenção e contribuição dedicadas a este estudo.

Gostaria de deixar registrado também o meu reconhecimento à minha família, pois acredito que sem o seu apoio seria muito difícil vencer este desafio.

Por último, mas não menos importante, agradeço a minha querida esposa pelo carinho, pelo amor e pela compreensão.

A todos, muito obrigado.

“Mil poderão cair ao teu lado, e dez mil à tua direita; mas tu não serás atingido. Somente com os teus olhos contemplarás, e verás a recompensa dos ímpios. Porquanto fizeste do Senhor o teu refúgio, e do Altíssimo a tua habitação, nenhum mal te sucederá, nem praga alguma chegará à tua tenda.”

(Salmo: 91, 7-10).

## RESUMO

COSTA, Renivaldo P. **Aplicando a engenharia de requisitos com o processo unificado - um estudo de caso para sistemas web**. 2013. 47 f. Dissertação (Especialização em Desenvolvimento Web) – Universidade Tecnológica Federal do Paraná. Londrina, 2013.

Este estudo é voltado para a apresentação de boas práticas da engenharia de software, utilizando-se uma das suas disciplinas, a engenharia de requisitos, como forma de melhorar a qualidade das aplicações web, no que tange o levantamento de requisitos.

**Palavras-chave:** Engenharia de Requisitos. Análise de Requisitos. Engenharia de Software. Requisitos de Sistema.

## ABSTRACT

COSTA, Renivaldo P. **Applying the requirements engineering with the unified process - a case study for web systems**. 2013. 47 f. Dissertation (Specialization) – Universidade Tecnológica Federal do Paraná. Londrina, 2013.

This study is focused on the presentation of good practices of software engineering, using one of their disciplines, requirements engineering, as a way to improve the quality of web applications, regarding requirements elicitation.

**Keywords:** Requirements Engineering. Requirements Analysis. Software Engineering. System Requirements.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Processo .....	20
Figura 2 – Processo de Software .....	20
Figura 3 – Ciclo de vida de um projeto .....	21
Figura 4 – Desenvolver Visão Técnica .....	24
Figura 5 – Modelo de Domínio .....	32
Figura 6 – Modelo de Negócio .....	33
Quadro 1 – Caso de Uso Logar no Sistema.....	34
Quadro 2 – Caso de Uso Cadastrar Usuário.....	35
Quadro 3 – Caso de Uso Alterar Usuário .....	35
Quadro 4 – Caso de Uso Excluir Usuário.....	36
Quadro 5 – Caso de Uso Incluir Chamado.....	37
Quadro 6 – Caso de Uso Alterar Chamado.....	37
Quadro 7 – Caso de Uso Consultar Chamado .....	38
Quadro 8 – Caso de Uso Excluir Chamado .....	39
Quadro 9 – Caso de Uso Finalizar Chamado.....	39

## LISTA DE TABELAS

Tabela 1 - Cronograma de Execução.....	44
--	----

## LISTA DE SIGLAS

HTTPS	<i>HyperText Transfer Protocol Secure</i>
MVC	<i>Model View Controller</i>
PU	Processo Unificado
SIAC	Sistema de Abertura de Chamado
SGBD	Sistema Gerenciador de Banco de Dados
UML	<i>Unified Modeling Language</i>
UI	<i>User Interface</i>
UTFPR	Universidade Tecnológica Federal do Paraná

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
<b>2 OBJETIVOS</b> .....	<b>15</b>
2.1 OBJETIVO GERAL .....	15
2.2 OBJETIVOS ESPECÍFICOS .....	15
<b>3 METODOLOGIA</b> .....	<b>16</b>
<b>4 REFERENCIAL TEÓRICO</b> .....	<b>17</b>
4.1 ENGENHARIA DE REQUISITOS PARA APLICAÇÕES WEB.....	17
4.2 PROCESSO.....	19
4.3 PROCESSO UNIFICADO .....	21
4.3.1 Concepção.....	22
4.3.2 Elaboração.....	22
4.3.3 Construção.....	22
4.3.4 Transição .....	22
4.4 ATIVIDADES DE REQUISITOS NO PU .....	23
4.4.1 Artefatos gerados na atividade de requisitos .....	25
4.4.2 Trabalhadores ligados a atividades de requisitos .....	28
<b>5 ESTUDO DE CASO</b> .....	<b>30</b>
5.1 DESCRIÇÃO DO SISTEMA.....	30
5.2 MODELO DE DOMÍNIO .....	31
5.3 MODELO DE NEGOCIO.....	32
5.4 CASO DE USO (DESCRIÇÃO) .....	33
5.5 REQUISITOS SUPLEMENTARES .....	40
5.5.1 Requisitos Organizacionais .....	40
5.5.2 Outros Requisitos .....	40
<b>6 DISCUSSÕES E RESULTADOS</b> .....	<b>41</b>
<b>7 CONCLUSÃO</b> .....	<b>43</b>
<b>8 CRONOGRAMA DE EXECUÇÃO</b> .....	<b>44</b>
<b>9 CONTRIBUIÇÕES ESPERADAS</b> .....	<b>45</b>
<b>REFERÊNCIAS</b> .....	<b>46</b>

## 1 INTRODUÇÃO

Este capítulo apresenta uma introdução sobre a importância da atividade de coleção de requisitos na engenharia de software.

A engenharia de software é uma área da computação voltada à especificação, desenvolvimento e manutenção de sistemas de software, comumente aplicada através de tecnologias e práticas de gerência de projetos, gerencia de requisitos, gerencia de testes, gerencia de processos, entre outras gerencias que aqui são denominadas disciplinas. A somatória destas disciplinas nos levam a obtenção de processos mais organizados, produtividade e qualidade.

A engenharia de software não está apenas relacionada à análise, hoje ele se envolve nas ferramentas de desenvolvimento, banco de dados, ferramentas complementares, plataformas, bibliotecas, padrões, processos e a qualidade de software.

A engenharia de software utiliza-se de modelos abstratos que permitem especificar, projetar, implementar, avaliar, qualificar e manter sistemas de software. Fora isso a engenharia de software oferece mecanismos de planejamento e gerenciamento de processos de desenvolvimento.

Neste estudo abordaremos apenas umas das tantas disciplinas de engenharia de software, a engenharia de requisitos voltada a aplicações web.

A engenharia de requisitos é um processo que engloba todas as atividades que contribuem para a produção de documentos de requisitos. Os requisitos estão presentes em quase todo o ciclo de vida do software e pode variar de acordo com o escopo ou a necessidade dos envolvidos.

A identificação dos requisitos é um processo lento e minucioso e de suma importância, onde é dividido em algumas atividades:

- Identificação;
- Análise e negociação;
- Especificação e documentação;
- Validação.

Mesmo após a produção da documentação se faz necessário a correção dos requisitos, mesmo porque as necessidades, escopo e os envolvidos estão em constante mudança.

## 2 OBJETIVOS

Este capítulo visa apresentar os objetivos levantados para desenvolver o trabalho proposto. A subseção 2.1 expõe o objetivo geral e a subseção 2.2 elicit os objetivos específicos.

### 2.1 OBJETIVO GERAL

Este estudo tem como principal objetivo mostrar a disciplina de engenharia de requisitos através da notação UML utilizando o processo unificado (PU), para desenvolver uma documentação de maior qualidade e facilitar o entendimento dos processos fornecidos pelos *stakeholders* no desenvolvimento da aplicação web.

### 2.2 OBJETIVOS ESPECÍFICOS

- Demonstrar todo o processo de criação de um software web em uma empresa que usa apenas alguns diagramas do PU.
- Mostrar que o uso de alguns recursos de requisitos web pode agregar ainda mais valor no produto da empresa.
- Nesse processo é mostrado que usando a descrição do caso de uso, o diagrama de caso de uso e o diagrama de classe é possível desenvolver um sistema com qualidade.

### 3 METODOLOGIA

O presente capítulo visa exibir a metodologia adotada para elaboração do trabalho.

Para o estudo será adotado o Modelo Incremental, que divide o projeto em partes gerenciáveis, de forma a incrementar as funcionalidades continuamente até o final da construção do produto, seguindo o paradigma Orientado a Objetos. Será utilizada a Unified Modeling Language (UML) (BOOCH, 2000) para visualizar, especificar, construir e documentar as funcionalidades da disciplina de requisitos. Para ilustrar a especificação das técnicas de modelagem será utilizada a ferramenta CASE Rational Rose.

A fim de aplicar a pesquisa realizada, foi realizado um estudo de caso numa empresa de desenvolvimento de software. O estudo de casos é de caráter exploratório. Seu objetivo é estabelecer uma plataforma para ajudar o discernimento na recepção de intuições sobre a terceirização de um dado problema. Um estudo de caso, segundo Fachin (2001), objetiva principalmente ser a experiência prática refinando a maneira de pensar empírico-indutivo para a construção de teorias.

## 4 REFERENCIAL TEÓRICO

O quarto capítulo apresenta um levantamento bibliográfico acerca do tema abordado neste trabalho. A subseção 4.1 define a aplicação da engenharia de requisitos em ambientes web. A subseção 4.2 teoriza o conceito de processo no âmbito da engenharia de requisitos. A subseção 4.3 descreve o processo unificado, bem como as fases que o compõem. Por fim, a subseção 4.4 trata das atividades de coleção de requisitos do processo unificado, bem como os artefatos e recursos envolvidos nas mesmas.

### 4.1 ENGENHARIA DE REQUISITOS PARA APLICAÇÕES WEB

Requisito é definido como um serviço ou uma limitação.

De acordo com Boehm (1981), a engenharia de requisitos de projeto responde por cerca de 6% dos gastos totais do projeto. Esses números foram confirmados por Chatzoglou e Macaulay (1996) em estudos realizados com uma amostragem de 107 projetos. A pesquisa mostrou que o custo da fase de engenharia de requisitos consome entre 5% a 15% do custo total do projeto.

Para aplicações de sistemas voltados para a web, são necessários alguns ajustes e adaptações nos processos regulares da engenharia de requisitos. A principal consideração a se fazer é quão rápida a aplicação deve responder as necessidades do utilizador, então devemos nos preocupar com o dimensionamento do ambiente alocado nos formulários e recursos que serão dispostos ao utilizador do sistema.

Cada projeto de software tem suas características, e são particulares em cada sistema, podemos dizer então que cada aplicação é uma pequena quebra de paradigma; mas, no desenvolvimento de software voltado a web, algumas características são comuns a todos, como a infraestrutura de rede (servidores, *backup*, *links*, *backbones*), segurança, acessibilidade e o meio de acesso à aplicação.

Um grande problema quando se desenvolve uma aplicação para a web é a infraestrutura de rede, pois nem todo o usuário tem um acesso de qualidade, isso implica na forma como aplicação se apresenta ao usuário, nesse ponto que o layout

e o designer de nossa aplicação ajuda a evitar problemas de uso e lentidão do sistema.

A segurança de uma aplicação web apesar de basear em protocolos de seguros de rede como o HTTPS, e garantir a criptografia dos dados transmitidos entre o cliente e o servidor, não é suficiente para garantir a segurança total das informações que podem ser capturadas através de softwares espíões ou até mesmo em visitas a sites maliciosos onde as informações dos *cookies* dos usuários são capturadas. Portanto, a eliminação dessas vulnerabilidades se faz necessária para diminuir os riscos de roubo de informações nas aplicações web.

Além disso, podemos levar em consideração recomendações como:

- Utilização de um valor pequeno para o tempo de expiração do *cookie*.
- Evitar utilizar *cookies* persistentes, impedindo assim que um atacante o roube, caso tenha acesso físico ao computador do usuário.
- Separar os cookies de autenticação dos cookies de personalização, que são utilizados para armazenar as preferências dos usuários.

O Navegador é outro item que se deve levar em consideração, pois não a uma compatibilidade definida de como uma aplicação será visualizada em cada navegador ou como os recursos irão se apresentar em cada navegador e ainda temos que pensar na retrocompatibilidade, assim a aplicação como ele se comportara nos navegadores mais antigos e um mais atual deve-se criar maneiras para que o aplicativo seja visto com qualidade em várias plataformas.

A principal função da disciplina de requisitos é realizar o levantamento das informações do negócio para o qual a aplicação está sendo desenvolvido. É preciso também entender as necessidades dos envolvidos no processo, os *stakeholders*, para compreender o contexto no qual a aplicação será inserida. A participação dos *stakeholders* nesta fase é fator primordial para o sucesso do projeto e impacta diretamente no custo do mesmo.

As aplicações desenvolvidas para a web têm alguns pontos muito importantes como a navegabilidade e o design. Um sistema voltado para a web tem que ter uma navegabilidade bem definida para que não traga aos *stakeholders* uma experiência desagradável. O levantamento das informações deve ser o mais detalhado possível, a fim de minimizar as possibilidades de erro. O design, por sua vez é a interface do sistema. Se as informações levantadas estiverem bem

distribuídas em um formulário, o *stakeholder* terá uma boa navegação e domínio da aplicação.

As aplicações web estão mais susceptíveis a falhas de segurança por estarem em uma rede aberta, o estudo de caso irá tratar de uma aplicação dentro de uma intranet onde a incidência de falhas na segurança não impacta na utilização da aplicação.

A disciplina de requisitos está separada em requisitos funcionais e requisitos não funcionais (ESCALONA; KOCH, 2004). Segundo os autores, os requisitos não funcionais estão focados em definir as fronteiras do sistema e as suas restrições. Já os requisitos funcionais podem ser divididos em outras categorias como:

- Requisitos de dados: têm como função estabelecer como as informações serão armazenadas e administradas pelos sistemas web.
- Requisitos de interfaces: estabelecem como o usuário interagirá com o sistema.
- Requisitos de navegação: representam a forma com a qual o usuário navegará por meio do sistema na web.
- Requisitos de personalização: são usados para definir a adaptação do usuário com o sistema.
- Requisitos transacionais: têm como função definir o que o sistema executará sem considerar aspectos da interface e da interação.

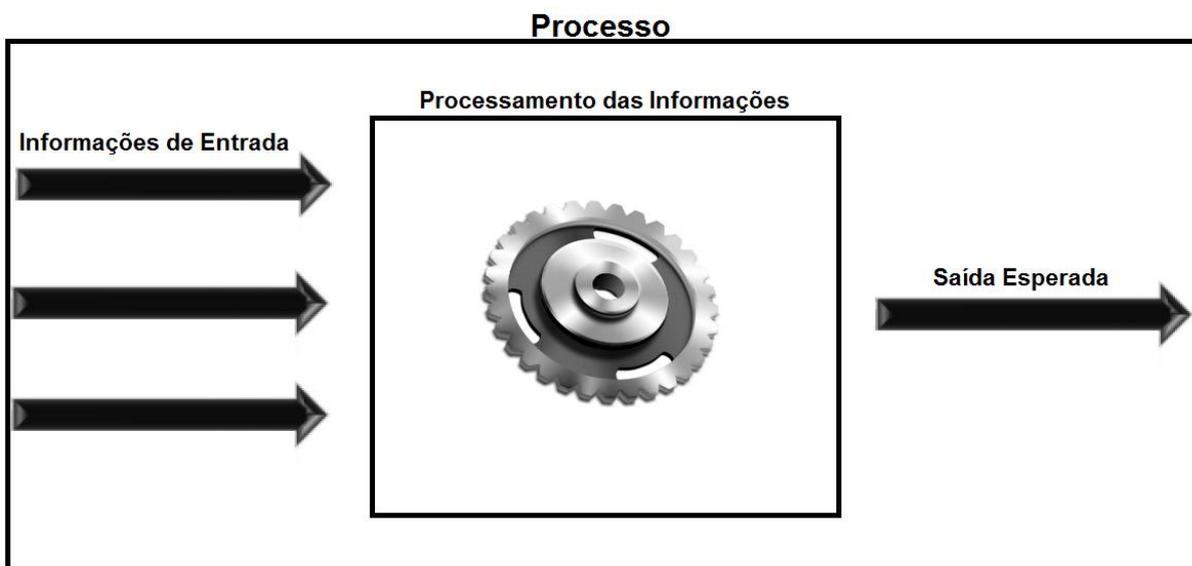
Uma aplicação web é feita para que possa ser acessada através de navegadores em uma rede interna ou na internet (ANICETO, 2009, p. 2).

## 4.2 PROCESSO

Os processos na engenharia de requisitos são usados para descobrir, analisar e validar requisitos do sistema e podem descrever atividades como, técnicas de obtenção e análise de requisitos, validação de requisito, revisões de requisitos, gerenciamento de requisitos.

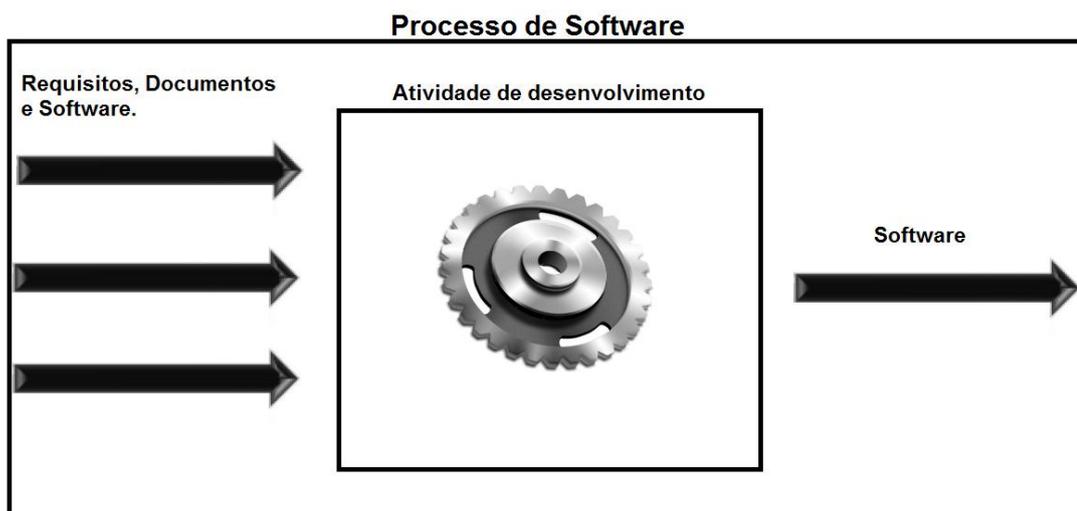
O processo usado na engenharia de requisitos varia bastante, dependendo do domínio da aplicação, os *stakeholders* envolvidos e a organização e podem ser definido como um conjunto de atividades organizadas que transformam uma

determinada informação de entrada em uma informação de saída desejada como mostrado na figura 1.



**Figura 1 – Processo**  
Fonte: Autoria própria.

Na figura 2 é mostrado o processo de desenvolvimento de um software, que segundo Booch (2000) é um conjunto de resultados e atividades que, associados, geram como resultado um produto de software.



**Figura 2 – Processo de Software**  
Fonte: Autoria própria.

Um dos processos mais utilizados é o Processo Unificado, o qual segundo Larman (2004) se divide em quatro fases: i) Iniciação ou Concepção, nesta fase é feito o levantamento dos requisitos; ii) Elaboração, nela se realiza a análise dos requisitos e o projeto do software; iii) Construção, nesta fase o software é implementado e testado; iv) Transição, nela o software é implantado no sistema do cliente no item abaixo detalharemos um pouco mais sobre cada um desses itens.

### 4.3 PROCESSO UNIFICADO

O processo unificado surgiu para realizar o desenvolvimento de software visando construção de sistemas orientados a objetos. É um processo iterativo e adaptativo.

Ainda segundo Larman (2004) principal ideia do PU (Processo Unificado) é o desenvolvimento iterativo e incremental que consiste na repetição de uma serie de ciclos no desenvolvimento de um sistema. Os ciclos são divididos em: i) Iniciação ou Concepção; ii) Elaboração; iii) Construção e; iv) Transição, e estas fases são subdivididas em: i) Requisitos; ii) Análise; iii) Projeto; iv) Implementação e; v) Teste, como é mostrado na figura abaixo.

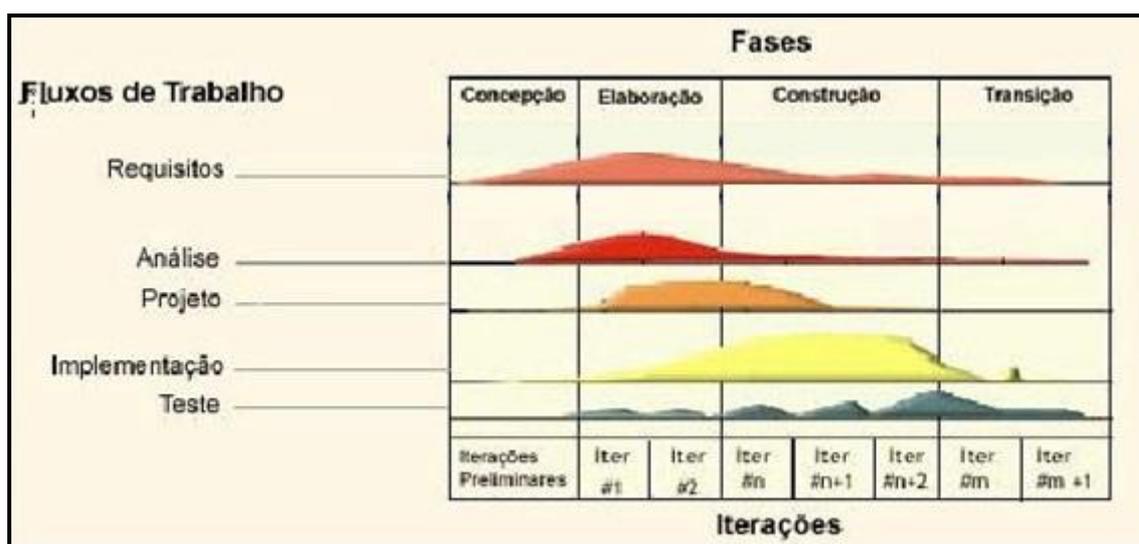


Figura 3 – Ciclo de vida de um projeto

Fonte: <http://www.adonai.eti.br/wordpress/2011/06/processo-unificado-pu-unified-process/>

#### 4.3.1 Concepção

A etapa de Concepção trabalha com o domínio do problema e tenta, assim, determinar o que o sistema deve fazer. Além do mais, nessa etapa busca-se o desenvolvimento do software. Para tanto, devem ser realizadas diversas entrevistas com o cliente até que o escopo do problema esteja bem definido. Nessas entrevistas, o engenheiro deve auxiliar o cliente nas definições dos requisitos, de modo a ser conhecer o desempenho esperado do sistema, a prioridade nessa fase é a ênfase no escopo do sistema.

#### 4.3.2 Elaboração

Na etapa de Elaboração, também é conhecida como análise de requisitos, todas as necessidades informadas pelo cliente são analisadas. Nela o engenheiro examina os requisitos informados pelos usuários e as reais necessidades do sistema de informação são determinadas, a prioridade nessa fase é a ênfase na arquitetura.

#### 4.3.3 Construção

Na etapa de Construção começa a ser desenvolvido do produto, programação e testes do software, o trabalho inicia com base no que foi definido na fase de Elaboração, são detalhados os casos de uso remanentes e a descrição da arquitetura é modificada se for necessário, a prioridade nessa fase é a ênfase no desenvolvimento.

#### 4.3.4 Transição

A etapa de Construção tem como objetivo disponibilizar um produto para o cliente em seu ambiente, a partir desta versão procura-se por possíveis falhas mínimas que passaram despercebidas pela fase de Construção, a prioridade nessa fase é a ênfase na implantação do sistema.

Cumprido ressaltar que o processo unificado é um processo iterativo e incremental, em que cada iteração resulta em um incremento, o qual se torna uma

nova versão do software, esse já aperfeiçoado em comparação com a versão anterior.

#### 4.4 ATIVIDADES DE REQUISITOS NO PU

Os requisitos do sistema são especificados através da identificação das necessidades de usuários e clientes, estes requisitos são transformados em casos de uso através do modelo de casos de uso. Os casos de uso são representados através da notação UML, onde cada caso de uso é composto pelos diagramas de casos de uso que compõem o sistema.

De acordo com Scott (2003) durante a fase de Concepção, os requisitos mais importantes são identificados, delimitando assim o domínio do sistema.

Na fase de Elaboração, os requisitos remanescentes são analisados, permitindo a equipe de desenvolvedores identificarem o real tamanho do sistema. Ao final da fase de Elaboração, grande maioria dos requisitos do sistema já está descrito, no entanto, apenas uma pequena parte destes requisitos terá sua implementação nesta fase. Os requisitos remanescentes serão identificados e implementados durante a fase de Construção.

A fase de Transição praticamente não possui requisitos a serem identificados, a menos que ocorram mudanças nos requisitos descritos anteriormente ou no domínio do projeto.

A fase de trabalho de requisitos, aqui chamada de disciplina, reúne todas as tarefas relativas ao processo de análise e especificação de requisitos e análise de negócios do PU, com uma maior ênfase na fase de construção do PU.

A disciplina de requisitos possui quatro tarefas básicas (OPEN UP, 2013), são elas: i) Desenvolver visão técnica; ii) Identificar e esboçar os requisitos; iii) Detalhar os cenários de casos de uso e; iv) Especificar os requisitos não funcionais. Essas disciplinas são executadas sequencialmente, como apresentado na figura 4.

- Desenvolver visão técnica.
- Identificar e esboçar os requisitos.
- Detalhar os cenários de casos de uso.
- Especificar os requisitos não funcionais.

A seguir, as mesmas serão descritas com mais detalhes.



**Figura 4 – Desenvolver Visão Técnica**

Fonte: Open UP (2013).

### **Desenvolver visão técnica**

A tarefa de desenvolver visão técnica é, dado um determinado problema de negócio, independentemente do tipo da corporação, e propõe uma solução que seja viável e tenha a concordância de todos os envolvidos no processo. Os *stakeholders* auxiliam a equipe de desenvolvimento para identificar e documentar seus problemas, necessidades e potenciais características que o sistema possa vir a ter.

O principal resultado dessa fase é o documento de visão que apresenta a análise do ambiente de negócio como um todo.

### **Identificar e esboçar os requisitos**

O trabalho de identificação e esboço dos requisitos é delinear como identificar e esboçar os requisitos para os requisitos da solução proposta, de modo a determinar a finalidade do sistema.

Como principal objetivo dessa etapa tem-se a identificação e a captura dos requisitos funcionais e não funcionais para a solução proposta. Estes requisitos constituem a base para o entendimento e a aceitação entre a equipe de desenvolvimento e os *stakeholders* sobre quais são as necessidades para a solução proposta atender ao negócio analisado.

O foco é a compreensão dos requisitos em alto nível de maneira que o escopo inicial do trabalho possa ser determinado. Depois, uma análise minuciosa deve ser feita para refinar estes requisitos antes do processo de implementação.

### **Detalhar os cenários de casos de uso**

Nesta tarefa o principal objetivo é nortear os analistas no processo refinamento das especificações e no processo de detalhamento dos cenários de caso de uso.

### **Especificar os requisitos não funcionais**

O propósito desta etapa é apresentar os diversos requisitos não funcionais, esmiuçados o suficiente para permitir a compreensão e a validação dos mesmos por parte dos desenvolvedores e dos *stakeholders*, permitindo a evolução da solução proposta.

De acordo com Scott (2003) os requisitos não funcionais estão relacionados a questões como desempenho, segurança, escalabilidade e confiabilidade.

#### 4.4.1 Artefatos gerados na atividade de requisitos

Os artefatos que podem ser gerados na atividade de requisitos segundo Scott (2003) são: Modelo de Domínio, Modelo de Negócio, Glossário, Ator, Caso de Uso, Protótipo da Interface com Usuário, Modelo de Casos de Uso, Descrição da Arquitetura e Requisitos Suplementares, abaixo detalharemos um pouco sobre cada um deles.

#### **Modelo de Domínio**

Modelo de domínio tem como objetivo capturar as entidades e os conceitos importantes do mundo real pertencente ao problema, problema este que o novo sistema deve solucionar.

Essas entidades são mapeadas para classe, no modelo de domínio é gerado o diagrama de classe da UML, porem esse diagrama não contem muitos detalhes.

#### **Modelo de Negócio**

O modelo de negócio se refere a dois modelos, o de objetos de negócio e o de casos de uso de negócio.

No modelo de objetos de negócio descreve como os casos de uso de negócio são realizados por um conjunto de *stakeholders*. Esse modelo também descreve as entidades de negócio.

No modelo de casos de uso de negócio é composto pelos casos de uso de negócios que são responsáveis por descrever processos de negócios, atores de negócios, que por sua vez representam os clientes e parceiros.

### **Glossário**

O componente glossário contém vários termos peculiares para o sistema, termos esse que podem descrever fatos concretos como entidades.

O objetivo do glossário é facilitar a aceitação de terminologias usadas pelos participantes do projeto. O glossário deriva geralmente do modelo de domínio, do modelo de negócio ou de ambos.

### **Ator**

Um ator pode ser um usuário que interaja com o sistema ou um outro sistema, como um banco de dados que reside fora do sistema.

O nome de um ator não deve ter um nome de uma pessoa em particular e sim deve identificar um papel ou um conjunto de papéis do sistema, o ator pode estar ligado a um ou mais casos de uso.

### **Casos de Uso**

O caso de uso pode ser definido como uma sequência de atividades executadas pelos atores para atingir um objetivo particular.

Um caso de uso descreve uma atividade que o sistema precisa executar sem especificar como isso será feito. O caso de uso é geralmente representado como uma elipse com um nome curto como Cadastrar Cliente, note que o nome é composto por um verbo no infinitivo e um substantivo, o nome do caso de uso pode se deixado dentro da elipse ou abaixo dela.

### **Protótipo da Interface com o Usuário**

O protótipo da interface com o usuário (UI) auxiliam os desenvolvedores a identificar a forma com que o usuário irá interagir com o sistema.

No processo unificado um protótipo de interface pode ser desenvolvido de qualquer forma, desde um esboço em folha de papel até protótipos interativos completamente operacionais, de toda forma esses protótipos geram uma fonte excelente de material para os casos de uso.

### **Modelo de Casos de Uso**

Um modelo de caso de uso é basicamente um pacote de pacotes de casos de uso, onde cada qual contendo atores e casos de uso.

Em um diagrama UML o pacote é representado como uma pasta com etiqueta ou aba, tendo duas variações o nome do pacote aparecendo dentro da aba e seu conteúdo listado no corpo da pasta ou apenas o nome do pacote aparecendo no corpo da pasta.

### **Descrição da Arquitetura**

Parte da arquitetura é uma visão do modelo de casos de uso que contem os casos de uso arquitetonicamente importantes para o sistema, por sua vez esse casos de uso são os que descrevem funcionalidades importantes do sistema, se refere a requisitos especialmente significativos para o sistema.

### **Requisitos Suplementares**

Os requisitos suplementares são requisitos não funcionais que tratam questões como desempenho, segurança e cópias de segurança ou restrições impostas por agências reguladoras. Um requisito suplementar dificilmente é mapeado a um único caso de uso, no entanto pode ser abordado em vários casos de uso.

#### 4.4.2 Trabalhadores ligados a atividades de requisitos

Os trabalhadores envolvidos nas atividades de requisitos ainda segundo Scott (2003) são: Analista de Sistema, Especificador de Casos de Uso, Projetista de Interface com o Usuário e Arquiteto, abaixo falaremos um pouco mais sobre cada um deles.

##### **Analista de Sistemas**

O analista de sistemas tem como objetivo principal capturar os requisitos relativos aos casos de uso.

Deve conduzir o seu esforço em estabelecer os modelos de domínio e de negócio, extraindo os atores e os casos de uso assegurando assim a completeza e consistência do modelo de casos de uso como um todo, geralmente é também o principal responsável pela geração do glossário.

##### **Especificador de Casos de Uso**

Esse profissional tem com grande responsabilidade escrever as descrições detalhadas dos fluxos principal e excepcional de eventos para um ou mais casos de uso.

O responsável pela especificação de casos de uso também pode ser responsável por um pacote de casos de uso e por sustentar a integridade desse pacote. É recomendável que o especificador de casos de uso seja responsável por um pacote de casos de uso também seja responsável pelos casos de uso e atores contidos no pacote.

##### **Projetista de Interfaces com o Usuário**

O projetista de interfaces com o usuário tem como seu foco projetar uma interface que determine de forma visual os elementos de interface que um ou mais usuários irão usar.

Seja um protótipo interativo operacional ou um esboço em papel esse protótipo vai definir como o usuário irá interagir com o sistema.

Além disso, o papel projetista de interfaces com o usuário pode ser responsável por um ou mais pacotes de design ou subsistemas de protótipos, incluindo todas as classes pertencentes aos pacotes ou subsistemas.

### **Arquiteto**

O arquiteto prioriza o conjunto de casos de uso dado e inclui os casos de uso arquitetonicamente de maneira significativa na descrição da arquitetura.

A responsabilidade de profissional é muito grande, pois se a priorização dos casos de uso não for bem estruturada pode se ter alguns gastos extras no projeto.

## 5 ESTUDO DE CASO

Yin (2005) define estudo de caso como uma lógica de planejamento de pesquisa empírica que investiga um fenômeno contemporâneo em seu contexto real, especialmente quando os limites entre fenômeno e contexto não são claramente evidentes. Nesse delineamento, o pesquisador precisa mergulhar numa situação real, acompanhando-a e analisando-a, a fim de aplicar e validar seu trabalho.

A fim de demonstrar a importância da atividade de requisitos na produção de um sistema, foi realizado o estudo de casos, realizando o levantamento dos requisitos de um sistema de abertura de chamado (SIAC) que deverá ser desenvolvido para ambiente web, pela empresa Evolut®.

A empresa Evolut® pode ser classificada como empresa de pequeno porte, com um quadro de funcionários pequeno, onde cada um tem tempo restrito para realização de suas atividades. Dessa forma, devido à redução de custos e tempo, a empresa não faz uso de todos os artefatos indicados pelo Processo Unificado. Os artefatos que são usados na empresa são os de Modelo de Domínio, Modelo de Negócio, Caso de Uso e Requisitos Suplementares, por serem considerados essenciais para a elaboração eficiente de um sistema.

O capítulo 5 delinea o estudo de caso realizado, apresentando os casos de uso do sistema, bem como os elementos que o compõem. A subseção 5.1 exibe uma descrição do que o sistema deve fazer e os artefatos gerados para o sistema proposto. A subseção 5.2 apresenta a forma de uso do modelo de domínio. A subseção 5.3 expõe o modelo de negócios e sua usabilidade por parte da empresa. O detalhamento dos casos de uso na forma como são utilizados pelas empresas é apresentado na subseção 5.4. Finalmente, a subseção 5.5 levanta os demais requisitos do sistema.

### 5.1 DESCRIÇÃO DO SISTEMA

A empresa Evolut® atua no segmento comercial de desenvolvimento de softwares. O sistema SIAC proporcionará o controle de chamados para prestação de serviços.

O sistema tem como objetivo auxiliar no gerenciamento dos chamados abertos pelos usuários, como: excluir, incluir, consultar e alterar.

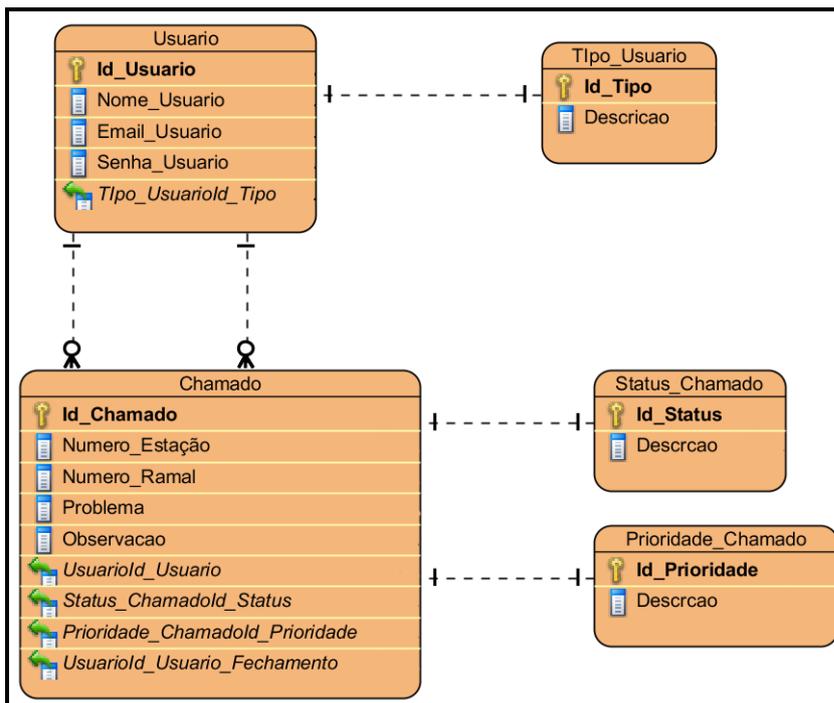
O sistema devera funcionar em todas as estações de trabalho da empresa. Os usuários deverão estar registrados no sistema para terem permissão de incluir, alterar, excluir e consultar chamados e somente o administrador do sistema poderá incluir, alterar e excluir usuários.

Para a abertura do chamado o usuário informa usuário e senha do sistema e com isso tem acesso à tela de abertura de chamado, onde pode informar o número de sua estação de trabalho, numero do ramal, prioridade do chamado e informar o problema que está ocorrendo e clicar no botão com função de salvar. Se o usuário não informar os dados como estação de trabalho, prioridade do chamado e defeito uma mensagem é mostrada informando que os dados são obrigatórios.

Após o funcionário responsável executar o atendimento do chamado, ele entra no sistema e muda o status do chamando para finalizado e preenche a observação.

## 5.2 MODELO DE DOMÍNIO

Abaixo é mostrado como o modelo de domínio é usado pela a empresa, nesse item a empresa procura extrair todas as entidades que serão usadas no sistema proposto para o cliente.

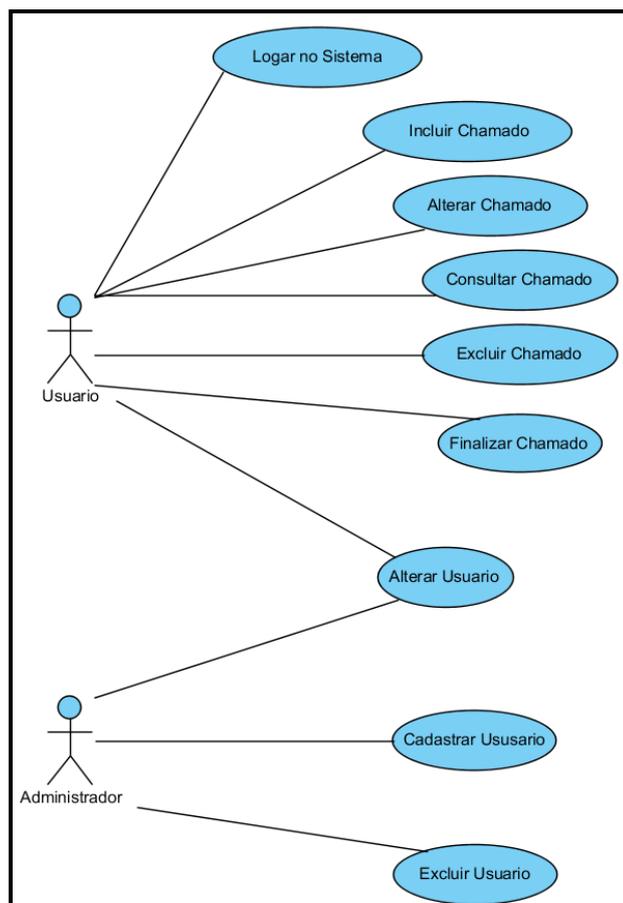


**Figura 5 – Modelo de Domínio**

**Fonte: Autoria própria.**

### 5.3 MODELO DE NEGOCIO

Abaixo é mostrado como o modelo de negócio é usado pela a empresa na tarefa de requisito, nesse item a empresa procura extrair todos os casos de uso que serão usados no sistema proposto para o cliente.



**Figura 6 – Modelo de Negócio**  
**Fonte: Autoria própria.**

#### 5.4 CASO DE USO (DESCRIÇÃO)

Abaixo é mostrado como os casos de uso são usados pela a empresa na tarefa de requisitos. Nesse item a empresa procura detalhar todos os casos de uso que serão usados no sistema proposto para o cliente.

Nome do Caso de Uso	Logar no Sistema
Caso de Uso Geral	<i>Logar no Sistema</i>
Ator Principal	Usuário
Atores Secundários	
Resumo	Usuário informa o usuário e senha para acessar o sistema.
Pré-Condições	O usuário deve estar cadastrado no sistema.
Pós-Condições	

<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informar o usuário e senha.	
	2. Sistema valida o usuário e a senha.
	3. Sistema exibe página de abertura de chamado.
<b>Fluxo Alternativo</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informar o usuário e senha.	
	2. Sistema valida o usuário e a senha.
	3. Sistema emite mensagem de usuário ou senha inválido.

**Quadro 1 – Caso de Uso Logar no Sistema**

Fonte: Autoria própria.

Nome do Caso de Uso	Cadastrar Usuário
Caso de Uso Geral	<i>Cadastrar Usuário</i>
Ator Principal	Administrador
Atores Secundários	
Resumo	O administrador do sistema acessa o cadastro de usuário, informa os dados do usuário e salva.
Pré-Condições	
Pós-Condições	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Administrador informar os dados do usuário.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema exibe mensagem de usuário cadastrado com sucesso.
<b>Fluxo Alternativo</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Administrador informar os dados do usuário.	
	2. Sistema valida se os dados estão corretos.

	3. Sistema emite mensagem de dados inconsistentes.
--	--

**Quadro 2 – Caso de Uso Cadastrar Usuário**

Fonte: Autoria própria.

Nome do Caso de Uso	Alterar Usuário
Caso de Uso Geral	<i>Alterar Usuário</i>
Ator Principal	Usuário
Atores Secundários	Administrador
Resumo	O administrador do sistema acessa o cadastro de usuário, informa os dados do usuário e salva.
Pré-Condições	
Pós-Condições	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informa os dados que quer alterar.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema exibe mensagem de usuário alterado com sucesso.
<b>Fluxo Alternativo</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informa os dados que quer alterar.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema emite mensagem de dados inconsistentes.

**Quadro 3 – Caso de Uso Alterar Usuário**

Fonte: Autoria própria.

Nome do Caso de Uso	Excluir Usuário
Caso de Uso Geral	<i>Excluir Usuário</i>
Ator Principal	Administrador
Atores Secundários	
Resumo	O administrador do sistema acessa a lista de usuário e exclui o usuário desejado.
Pré-Condições	
Pós-Condições	

<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Administrador seleciona o usuário que deseja excluir.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema exibe mensagem de usuário excluído com sucesso.
<b>Fluxo Alternativo</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Administrador seleciona o usuário que deseja excluir.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema emite mensagem de dados inconsistentes.

**Quadro 4 – Caso de Uso Excluir Usuário**

Fonte: Autoria própria.

Nome do Caso de Uso	Incluir Chamado
Caso de Uso Geral	<i>Incluir Chamado</i>
Ator Principal	Usuário
Atores Secundários	
Resumo	O usuário do sistema acessa a página de chamado e efetua a abertura do chamado.
Pré-Condições	
Pós-Condições	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informar os dados para abertura do chamado.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema exibe mensagem de chamado aberto com sucesso.
<b>Fluxo Alternativo</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informar os	

dados para abertura do chamado.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema emite mensagem de dados inconsistentes.

**Quadro 5 – Caso de Uso Incluir Chamado**

Fonte: Autoria própria.

Nome do Caso de Uso	Alterar Chamado
Caso de Uso Geral	<i>Alterar Chamado</i>
Ator Principal	Usuário
Atores Secundários	
Resumo	O usuário do sistema acessa a página de chamado e efetua a alteração do chamado.
Pré-Condições	
Pós-Condições	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informar os dados para alteração do chamado.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema exibe mensagem chamado alterado com sucesso.
<b>Fluxo Alternativo</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Usuário informar os dados para alteração do chamado.	
	2. Sistema valida se os dados estão corretos.
	3. Sistema emite mensagem de dados inconsistentes.

**Quadro 6 – Caso de Uso Alterar Chamado**

Fonte: Autoria própria.

Nome do Caso de Uso	Consultar Chamado
Caso de Uso Geral	<i>Consultar Chamado</i>
Ator Principal	Usuário
Atores Secundários	

Resumo	O usuário do sistema acessa a página de consulta de chamado e efetua a consulta.
Pré-Condições	
Pós-Condições	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. O sistema consulta se tem algum chamado aberto para o usuário.
	2. Sistema exibe a tela com os chamados aberto do usuário.
<b>Fluxo Alternativo</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. O sistema consulta se tem algum chamado aberto para o usuário.
	2. Sistema emite mensagem informando que não tem chamados abertos para o usuário.

**Quadro 7 – Caso de Uso Consultar Chamado**

Fonte: Autoria própria.

Nome do Caso de Uso	Excluir Chamado
Caso de Uso Geral	<i>Excluir Chamado</i>
Ator Principal	Usuário
Atores Secundários	
Resumo	O usuário do sistema acessa a página de consulta de chamado e efetua a exclusão do chamado.
Pré-Condições	
Pós-Condições	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. O sistema consulta se tem algum chamado aberto para o usuário.
	2. Sistema exibe a tela com os chamados abertos do usuário.
3. Usuário seleciona o chamado que deseja excluir e clica em excluir.	
	4. 2. Sistema emite mensagem informando que o chamado foi excluído com sucesso.

Fluxo Alternativo	
Ações do Ator	Ações do Sistema
	1. O sistema consulta se tem algum chamado aberto para o usuário.
	2. Sistema emite mensagem informando que não tem chamados abertos para o usuário.

**Quadro 8 – Caso de Uso Excluir Chamado**

Fonte: Autoria própria.

Nome do Caso de Uso	Finalizar Chamado
Caso de Uso Geral	<i>Finalizar Chamado</i>
Ator Principal	Usuário
Atores Secundários	
Resumo	O usuário do sistema acessa a página de consulta de chamado e efetua a finalização do chamado.
Pré-Condições	
Pós-Condições	
Fluxo Principal	
Ações do Ator	Ações do Sistema
	1. O sistema consulta se tem algum chamado aberto para o usuário.
	2. Sistema exibe a tela com os chamados abertos do usuário.
3. Usuário seleciona o chamado que deseja finalizar e muda o status para finalizado e clica em salvar.	
	4. 2. Sistema emite mensagem informando que o chamado foi finalizado com sucesso.
Fluxo Alternativo	
Ações do Ator	Ações do Sistema
	1. O sistema consulta se tem algum chamado aberto para o usuário.
	2. Sistema emite mensagem informando que não tem chamados abertos para o usuário.

**Quadro 9 – Caso de Uso Finalizar Chamado**

Fonte: Autoria própria.

## 5.5 REQUISITOS SUPLEMENTARES

A seguir, são descritos os requisitos organizacionais e os demais requisitos que compõem o sistema SIAC.

### 5.5.1 Requisitos Organizacionais

Os requisitos organizacionais que devem ser definidos para construção do sistema são: i) Software; ii) Linguagem de Programação e; iii) Hardware.

#### **Software**

O SGBD utilizado será o SQL Serve 2012, além da empresa já possuir a licença para esse SGBD é muito confiável.

#### **Linguagem de Programação**

O sistema será feito em ASP.NET MVC, que é a linguagem de programação utilizada no desenvolvimento de *websites* na empresa.

#### **Hardware**

A máquina servidora da base de dados será o servidor de aplicações da empresa onde também será publicada a aplicação.

### 5.5.2 Outros Requisitos

O sistema funcionará somente na rede interna da empresa.

## 6 DISCUSSÕES E RESULTADOS

O objetivo do levantamento de requisitos é obter informações sobre o software que será construído. Ao levantar tais informações deve-se focar no objetivo almejado pelo cliente, a fim de poder identificar as necessidades e o contexto de desenvolvimento do projeto.

Ao tomar a Engenharia de Requisitos como ponto de partida para o desenvolvimento, o sucesso do projeto aproxima-se significativamente. Realizar com eficácia o levantamento de requisitos favorece a delimitação do escopo do projeto e a definição e interligação das demais atividades a serem executadas, além de minimizar os riscos de mudanças nas fases posteriores. Quanto melhor for a realização do levantamento de requisitos, menor a ocorrência de erros custosos durante o desenvolvimento, influenciando no custo total do projeto.

Pressman (2006) indica a Engenharia de Requisitos como a atividade responsável por sistematizar o processo de definição de requisitos. Contudo, diversos projetos não possuem um levantamento de requisitos eficiente, o que conduz a erros e sistemas inconsistentes. E a complexidade dos sistemas exige que sejam identificadas as necessidades para o desenvolvimento do projeto, tornando necessária esta sistematização. Para que ela seja efetiva, é necessário um esforço fundamental na tarefa de coleção de requisitos, a fim de melhor entender o contexto no qual o software irá funcionar e escolher os modelos que melhor se ajustem ao ambiente.

A inconsistência na definição de requisitos pode trazer consequências de impacto para o projeto, podendo resultar em um sistema que não atende aos usuários e/ou ao objetivo proposto, aumentar os custos e o retrabalho e prejudicar os desenvolvedores tanto na construção quanto na manutenção do sistema. De acordo com Santos (2008), a incapacidade de entender e identificar as necessidades dos usuários tem sido a principal causa de falha de softwares.

Cordeiro (2005) aponta que a identificação de erros ainda na fase de coleção de requisitos pode diminuir em até 200 vezes o custo relativo do projeto em comparação com a descoberta de erros na fase de documentação. A partir desta observação, evidencia-se que a melhoria contínua de processos e o aumento de

esforço no levantamento de requisitos podem ser cruciais para o custo final do projeto, além de contribuir para a qualidade do software desenvolvido.

A empresa analisada no estudo de casos fez uso de alguns dos artefatos do Processo Unificado. A opção por tais artefatos foi feita tendo em vista a necessidade de otimizar e dar confiabilidade a seu processo de desenvolvimento sem incluir um número grande de atividades (já que o contingente de funcionários é pequeno).

Tendo em vista as restrições e o porte da empresa, nem todos os artefatos do Processo Unificado foram confeccionados, primando pelos que oferecessem uma definição inicial do sistema a ser desenvolvido. Com a adoção parcial do PU a empresa chegou a produtos consistentes e satisfatórios para os usuários.

A adoção de um modelo de processo deve ser compatível com as condições e necessidades da empresa. Contudo, dadas as proporções e sendo aplicável um modelo de processos, quanto mais suas indicações forem seguidas, maior a chance de que o software corresponda às especificações propostas.

## **7 CONCLUSÃO**

Com esse trabalho conclui-se que com aplicação do mínimo de análise é possível desenvolver uma aplicação para a web com uma qualidade boa e com isso ter uma melhor aceitação pelos usuários do produto.

Nesse estudo, procurou-se mostrar como artefatos específicos, utilizados no desenvolvimento de projetos Web, podem ser usados durante a construção de um sistema baseado no Processo Unificado. Além disso, mostrou-se um estudo de casos prático indicando como o processo pode ser ajustado de acordo com o tamanho de um projeto.

## 8 CRONOGRAMA DE EXECUÇÃO

Tabela 1 - Cronograma de Execução

Etapa / Atividade	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out
Revisão Bibliográfica/ Estado da Arte	X	X	X	X	X						
Desenho/ Implementação/ Codificação			X	X	X	X					
Escrita da Monografia				X	X	X	X	X	X	X	X

Fonte: Autoria própria.

### Etapas Fixas:

- 15/Outubro a 05/Novembro de 2012 – Entrega da monografia para a banca.
- 16/Novembro a 30/Novembro de 2012 – Apresentação da monografia.
- 11/Dezembro de 2012 – Entrega da versão final da monografia.

## **9 CONTRIBUIÇÕES ESPERADAS**

Pretende-se com este trabalho demonstrar que a documentação detalhada de todas as regras de negócio do sistema facilite a manutenção e novas implementações de software.

Através da aplicação de algumas das técnicas do PU é possível desenvolver uma aplicação web com qualidade.

## REFERÊNCIAS

ANICETO, Jefferson. **Aplicações Web**. Apostila ASP.net. Escola Técnica da Univale (ETEIT). 2009.

BOOCH, Grady. et. al. **UML Guia do Usuário**. Rio de Janeiro: Campos, 2000.

BOEHM, Barry. W. **Software engineering economics**. Englewood Cliffs, N. J.: Prentice-Hall, 1981.

CHATZOGLU, Prodromos. D.; MACAULAY, Linda. A. **Requirements capture and analysis: a survey of current practice**. Requir. Eng., v. 1, n. 2, 1996.

CORDEIRO, Marco A. **Uma Ferramenta de Suporte ao Processamento de Gerenciamento de Requisitos**. Dissertação de Mestrado, Universidade Estadual de Campinas, pags 7,8, 2005.

ESCALONA, Jose; KOCH, Nora. **Requirements Engineering for Web Applications**. *Journal of Web Engineering*. V.2, n. 3, p. 193-212, 2004.

FACHIN, Odília. **Fundamentos de metodologia**. 3. ed. São Paulo: Saraiva, 2001.

GUEDES, Gilleans T. A. **UML – Uma Abordagem Prática**, São Paulo: Novatec, 2011.

**IBM Rational Unified Process**. Disponível em: [http://pt.wikipedia.org/wiki/IBM\\_Rational\\_Unified\\_Process](http://pt.wikipedia.org/wiki/IBM_Rational_Unified_Process). Acesso em 05 fev. 2013.

LARMAN, Craig. **Utilizando UML e Padrões: uma introdução à análise e ao projeto orientado a objetos e ao Processo Unificado**. Trad. Luiz Augusto Meirelles Salgado e João Tortello. 2 ed. Porto Alegre: Bookman, 2004.

MARCHIONINI, Gary. **Exploratory search: from finding to understanding**. CACM, 49(4) 41-46, 2006.

**OPEN UP**. Disponível em: <http://open2up.blogspot.com.br/p/5-tarefas-disciplina-de-requisitos.html>. Acesso em 06 fev. 2013.

PRESSMAN, Roger. S. **Engenharia de software**. São Paulo: McGraw Hill, 2006.

SANTOS, Gilberto. **A Necessidade e a Importância da Engenharia de Requisitos**. 61f. Monografia de Pós-Graduação, Universidade Estadual de Londrina, 2008.

SCOTT, Kendall. **O Processo Unificado Explicado**. Ed. Bookman, 2003.

SILVA FILHO, Antonio. M. **Engenharia de software**. Magazine, p. 4-12, 2008.

SOMMERVILLE, Ian; SAWYER, Pete. **Requirements engineering: a good practice guide**. New York: Wiley, 1997.

YIN, Robert K. **Estudo de caso: planejamento e métodos**. 3. Ed. Porto Alegre: Bookman, 2005. 212 p.