

UNIVERSIDADE ESTADUAL DE GOIÁS
CAMPUS ITABERAÍ

DANILLO CARDOSO DE AGUAS
LUIS HENRIQUE PESSOA DA SILVA

**Análise e Desenvolvimento de um Sistema Web para Gerenciamento de Locação de
Artigos para Festas.**

ITABERAÍ

2015

DANILLO CARDOSO DE AGUAS
LUIS HENRIQUE PESSOA DA SILVA

**Análise e Desenvolvimento de um Sistema Web para Gerenciamento de Locação de
Artigos para Festas.**

Trabalho apresentado a Universidade Estadual de
Goiás – Campus de Itaberaí - como Trabalho de
Conclusão de Curso do ano letivo de 2015, sob a
orientação do professor Alisson Filgueiras.

Itaberaí
2015

**Danillo Cardoso de Aguas
Luis Henrique Pessoa da Silva**

**Análise e Desenvolvimento de um Sistema Web para Gerenciamento de
Locação de Artigos para Festas.**

Aprovada em ___ / ___ / ___

Banca examinadora:

Orientador Prof. Alisson Filgueiras

Leitor Prof. Juliana Vasconcelos Braga

Leitor Prof. Nádio Carlo

Dedicamos esse projeto aos nossos pais, que nos deram muito apoio nas dificuldades no decorrer do curso, que incentivaram e acreditaram que seríamos capazes de chegar até aqui.

Agradecemos primeiramente a Deus, aos nossos familiares pelo apoio que sempre nos deram, aos colegas que estivemos juntos nesse período e que colaboraram com nosso projeto, e ao nosso orientador Alisson Filgueiras.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes”. (Marthin Luther King).

RESUMO

O presente trabalho de conclusão de curso tem o objetivo de apresentar os documentos para a criação de um Sistema Web para Gerenciamento de Locações de Artigos para Festas, desenvolvido a partir de uma necessidade real. O sistema facilitará os processos da empresa, agilizando a locação e o controle dos produtos, e aprimorando a interação do cliente com a empresa. Serão apresentados a documentação de desenvolvimento de software, análise orientada a objetos, levantamento de requisitos, diagramas da UML, prototipação, entrevista. A metodologia utilizada será qualitativa, visando o contato direto com o cliente, buscando entender as necessidades da empresa, para desenvolvermos melhor o projeto e atender essas necessidades. Algumas referências usadas foram: Sommerville, Pressman, Bezerra, Guedes.

Palavras-chave: sistema, gerenciamento, locações, interação, software.

ABSTRACT

This course conclusion work aims to present the documents for the creation of a Web system for Articles locations Management for Parties, developed from a real need. The system will facilitate the company's processes, streamlining the lease and control of products, and improving customer interaction with the company. Will be presented the software development documentation, object-oriented analysis, requirements gathering, UML diagrams, prototyping, interview. The methodology used is qualitative, aimed at direct contact with the client, seeking to understand the company's needs, to better develop the project and meet those needs. Some references were used: Sommerville, Pressman, Bezerra, Guedes.

Keywords: system, management, leasing, interaction, software.

LISTA DE FIGURAS

Figura 1 Exemplo: Diagrama de caso de uso. Fonte: Gilleanes (2011).	27
Figura 2 Exemplo: Diagrama de Sequência Fonte: Gilleanes (2011).	28
Figura 3 Exemplo: Diagrama de Classes: Gilleanes (2011).	29
Figura 4 Exemplo: Dependência Fonte: Gilleanes (2011).	29
Figura 5 Exemplo: Especialização/Generalização Fonte: Gilleanes (2009).	30
Figura 6 Exemplo: Associação Fonte: Gilleanes (2011).	31
Figura 7 Exemplo: Agregação Fonte: Gilleanes (2011).	31
Figura 8 Exemplo: Composição Fonte: Gilleanes (2011).	32
Figura 9 Diagrama de Caso de Uso	45
Figura 10 Diagrama de Classes	54
Figura 11 MER	55
Figura 12 MRN.....	56
Figura 13 Tela Inicial.....	57
Figura 14 Tela de Login.....	58
Figura 15 Tela de Cadastro do Cliente	59
Figura 16 Tela de Produtos	60
Figura 17 Tela de um Produto Especifico.....	61
Figura 18 Tela de Gerenciar Locação	62
Figura 19 Tela do Carrinho.....	63
Figura 20 Tela de Decoração	64

LISTA DE TABELAS

Tabela 1 Lista de Requisitos	44
Tabela 2 Documentação de Caso de Uso: UC 01 - Gerenciar Produtos.....	46
Tabela 3 Documentação de Caso de Uso: UC 02 – Gerenciar Clientes	47
Tabela 4 Documentação de Caso de Uso: UC 03 - Gerenciar Locação	49
Tabela 5 Documentação de Caso de Uso: UC 04 – Efetuar Login.....	50
Tabela 6 Documentação de Caso de Uso: UC 05 – Realizar Orçamento.....	51
Tabela 7 Documentação de Caso de Uso: UC 06 – Efetuar Pagamento	52
Tabela 8 Documentação de Caso de Uso: UC 07 – Gerar Relatório	53
Tabela 9 Glossário de Mensagens	70
Tabela 10 Cronograma.....	71

LISTA DE SIGLAS E ABREVIATURAS

UML	Linguagem de Modelagem Unificada
AOO	Análise Orientada a Objeto
MRN	Modelo Relacional Normalizado
MER	Modelo Entidade Relacionamento
IDE	Integrated Development Environment (Ambiente Integrado de Desenvolvimento)
RE	Requirements Engineering (Engenharia de Requisitos)
OMG	Object Management Group (Grupo de Gerenciamento de Objetos)
B2B	Business-To-Business
B2C	Business-To-Consumer
B2A	Business-To-Administration
C2A	Consumer-To-Administration
C2C	Consumer-To-Consumer
C2B	Consumer-To-Business
PHP	Hypertext Preprocessor (Pré-processador de Hipertexto)
HTML	HiperText Markup Language (Linguagem de marcação de hipertexto).
HTTP	Hypertext Transfer Protocol

SUMÁRIO

INTRODUÇÃO	14
1 ENGENHARIA DE REQUISITOS	16
1.1 Levantamento de Requisitos	16
1.2 Requisitos Funcionais	17
1.3 Requisitos Não-funcionais	17
1.3.1 Requisitos de Domínio	17
1.3.2 Requisitos de Usuário	17
1.3.1 Requisitos de Sistema	18
2. ENGENHARIA DE SOFTWARE	19
2.1 Etapas de engenharia de software	19
2.2 Modelos de Processos de Software	20
2.2.1 Modelo Cascata	20
2.2.2 Desenvolvimento Evolucionário	21
2.2.3 Engenharia de software baseada em componentes	21
2.3 Requisitos de Sistema	21
2.3.1 A especificação de requisitos de software	22
2.4 Gerenciamentos de Projetos	23
2.4.1 Cronograma do projeto	24
3 ANÁLISE ORIENTADA A OBJETOS	25
4 UML	26
4.1 Diagrama de Caso de Uso	26
4.2 Diagrama de Sequência	27
4.3 Diagrama de Classes	28
4.3.1 Dependência	29
4.3.2 Generalização/Especialização	30
4.3.3 Associação	30
4.3.4 Agregação	31
4.3.5 Composição	31
5 E-COMMERCE	33
5.1 Business-To-Business - B2B (Empresa-Empresa)	35
5.2 Business-To-Consumer - B2C (Empresa-Consumidor)	35
5.3 Business-To-Administration - B2A (Empresa-Administração)	36
5.4 Consumer-To-Administration - C2A (Consumidor-Administração)	36
5.5 Outros Tipos De Comércio Eletrônico:	36
5.5.1 Consumer-To-Consumer - C2C (Consumidor-consumidor):	36

5.5.2	Consumer-To-Business - C2B (Consumidor-empresa):	36
6	BANCO DE DADOS	37
6.1	Normalização	38
7	PHP	39
7.1	Vantagens do PHP	40
7.2	O que pode ser feito com o PHP?	40
7.3	PHP com HTML	41
8	ESTUDO DE CASO	42
8.1	Modelagem do Negócio	42
8.1.1	Tema	42
8.1.2	Apresentação do tema	42
8.1.3	Objetivos	42
8.1.4	Justificativa	42
8.1.5	Metodologia e ferramentas utilizadas	42
8.2	Apresentação da Empresa	43
8.3	Problema	43
8.4	Proposta de Solução	43
8.5	Lista de Requisitos	44
8.6	Diagrama de Caso de Uso	45
8.7	Documentação de Caso de Uso	46
8.7.1	Documentação de Caso de Uso: UC 01 - Gerenciar Produtos	46
8.7.2	Documentação de Caso de Uso: UC 02 – Gerenciar Clientes	47
8.7.3	Documentação de Caso de Uso: UC 03 - Gerenciar Locação	48
8.7.4	Documentação de Caso de Uso: UC 04 – Efetuar Login	50
8.7.5	Documentação de Caso de Uso: UC 05 – Realizar Orçamento	51
8.7.6	Documentação de Caso de Uso: UC 06 – Efetuar Pagamento	52
8.7.7	Documentação de Caso de Uso: UC 07 – Gerar Relatório	53
8.8	Diagrama de Classe	54
8.9	Modelo Entidade Relacionamento – MER	55
8.10	Modelo Relacional Normalizado – MRN	56
8.11	Prototipação	57
8.11.1	Tela Inicial	57
8.11.2	Tela de Login	58
8.11.3	Tela de Cadastro de Cliente	59
8.11.4	Tela de Produtos	60
8.11.5	Tela de um Produto Especifico	61
8.11.6	Tela de Gerenciar Locação	62

8.11.7	Tela do Carrinho	63
8.11.8	Tela da Decoração	64
CONSIDERAÇÕES FINAIS		65
REFERÊNCIAS BIBLIOGRÁFICAS		66
APÊNDICE		67
Apêndice A - Entrevista		67
Apêndice B - Glossário de Mensagens		69
Apêndice C - Cronograma		71

INTRODUÇÃO

Este projeto de desenvolvimento de software, busca a implementação de um sistema na empresa “Decoração Chuva de Ouro” situada na cidade Itapuranga no estado de Goiás. A principal meta é aumentar e agilizar a interação entre o cliente e a empresa, visando manter o controle de locações, e modernizar a forma em que a empresa trabalha se adaptando aos novos conceitos de negócio.

O sistema tem como objetivo agilizar os processos e serviços, visando trazer mais facilidade para os clientes realizarem suas locações, verificando se a quantidade que ele precisa e a data estão disponíveis, e também contratarem a decoração de sua festa, verificando também o dia disponível.

A Metodologia utilizada será qualitativa, essa metodologia está ligada ao contato direto com o cliente. Através deles, é possível saber como a empresa utiliza seus métodos de trabalho, e com isto, será feita uma entrevista com cliente de software para entender o que realmente o ele precisa.

A entrevista é realizada com o intuito de conhecer as regras do negócio para realizar o levantamento de requisitos, onde será exposto todas as necessidades que o sistema deverá atender para que se possa ser útil para a empresa. Desse modo, serão definidos os requisitos não funcionais, a lista de requisitos, seguindo para os diagramas de caso de uso, diagramas de classe, junto com o MER (Modelo Entidade Relacionamento) e o MRN (Modelo Relacional Normalizado).

Nosso projeto possui um embasamento teórico, apresentando a análise de requisitos, onde é exposto as formas e as técnicas usadas para a modelagem de sistema, apresentando também uma série de exemplos e regras para que se possa gerar um projeto modelado seguindo os padrões da engenharia de software. Para o embasamento teórico utilizamos alguns livros de autores com destaque nessas áreas, por exemplo, livros de Sommerville, Pressman, Bezerra, Guedes, e outros.

Outro passo, é o estudo de caso, que é elaborado para identificar o problema, analisar as evidências, avaliar e propor soluções. Conforme são coletadas as informações do cliente, permite conhecer um pouco mais do usuário, os objetivos e como o sistema irá resolver o seu problema.

A prototipação apresenta uma demonstração de como ficará as telas do sistema, e o que ele terá e como poderá ser usado pelo usuário.

O trabalho está dividido em oito capítulos, onde o 1º trata sobre o Engenharia de Requisitos, o 2ª sobre Engenharia de Software, o 3º sobre Analise Orientada a Objetos, o 4º sobre UML, o 5º sobre E-Commerce, o 6º sobre Banco de Dados, o 7º sobre a Linguagem de Programação PHP e o 8ª é o Estudo de Caso.

1 ENGENHARIA DE REQUISITOS

Conforme Sommerville (2007), a Engenharia de Requisitos é uma de várias etapas no desenvolvimento de sistemas, considerada a parte mais difícil, pois está relacionada com a definição do que o sistema deverá fazer e decidir o que construir no sistema, portanto Engenharia de Requisitos nada mais é do que um processo de desenvolvimento de sistemas que tem prioridades e restrições que são essenciais, tais como, operação do sistema, comunicação entre os clientes e os usuários do sistema tanto quanto os respectivos desenvolvedores do próprio sistema.

Os requisitos de um sistema são descrições fornecidas pelo sistema e as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema, por exemplo, controlar um dispositivo, enviar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de engenharia de requisitos (RE – Requirements Engineering). O termo requisito não é usado pela indústria de sistema de maneira consistente. Em alguns casos, um requisito é simplesmente uma declaração abstrata de alto nível de um serviço que o sistema deve fornecer ou uma restrição do sistema. (SOMMERVILLE, 2007).

1.1 Levantamento de Requisitos

Uma das primeiras fases de um processo de desenvolvimento de software consiste no Levantamento de Requisitos. A etapa de levantamento de requisitos trabalha com o domínio do problema e tenta determinar “o que” o software deve fazer e se é realmente possível desenvolver o software solicitado. Nessa etapa, o engenheiro de software busca compreender as necessidades do usuário e o que ele deseja que o sistema a ser desenvolvido realize. Isso é feito sobretudo por meio de entrevistas, nas quais o engenheiro tenta compreender como funciona atualmente o processo a ser informatizado e quais serviços o cliente precisa que o software forneça.

Devem ser realizadas tantas entrevistas quantas forem necessárias para que as necessidades do usuário sejam bem compreendidas. Durante as entrevistas, o engenheiro deve auxiliar o cliente a definir quais informações deverão ser produzidas, quais deverão ser fornecidas e qual o nível de desempenho exigido do software.

Um dos principais problemas enfrentados na fase de levantamento de requisitos é o de comunicação. A comunicação constitui-se em um dos maiores desafios da engenharia de software, caracterizando-se pela dificuldade em conseguir compreender um conjunto de conceitos vagos, abstratos e difusos que representam as necessidades e os desejos dos clientes e transformá-los em conceitos concretos e inteligíveis. (GUEDES, 2011).

1.2 Requisitos Funcionais

É uma interação entre o sistema e seu ambiente que declaram os serviços que deveram ser prestados pelo sistema, como o sistema deve se comportar a entradas específicas e ainda os requisitos funcionais também podem em alguns casos descrever o que o sistema não deve fazer.

Os requisitos funcionais de um sistema descrevem o que o sistema deve fazer. Esses requisitos dependem do tipo de software em que está sendo desenvolvido, dos usuários a que o software se destina e da abordagem geral considerada pela organização ao redigir os requisitos. Quando expressos como requisitos de usuário, eles são geralmente descritos de forma bastante abstrata. No entanto os requisitos funcionais descrevem a função do sistema detalhadamente, suas entradas e saídas, exceções, etc. (SOMMERVILLE, 2007).

1.3 Requisitos Não-funcionais

Dizem respeito ao sistema como um todo, define as propriedades do sistema e suas restrições, sendo elas, confiabilidade, tempo de resposta e espaço de armazenamento e as representações de dados usados no sistema. Os requisitos não funcionais estão raramente associados às características individuais do sistema.

Conforme Sommerville (2007), esses requisitos especificam ou restringem as propriedades emergentes do sistema. Portanto, podem especificar desempenho, proteção, disponibilidade e outras propriedades emergentes do sistema. Isso significa que eles geralmente são mais importantes do que os requisitos funcionais individuais.

Os requisitos não funcionais surgem devido às necessidades do usuário, às restrições de orçamento, às políticas organizacionais, à necessidade de interoperabilidade com outros sistemas de software ou hardware ou a fatores externos, como regulamentos de segurança ou legislação a respeito da privacidade. (SOMMERVILLE, 2011).

1.3.1 Requisitos de Domínio

São derivados do domínio de aplicação do sistema, podendo ser novos requisitos funcionais, podem restringir os requisitos funcionais existentes, ou estabelecer como realizar cálculos específicos. Se não forem satisfeitos, poderá ser impossível fazer com que o sistema funcione satisfatoriamente.

1.3.2 Requisitos de Usuário

Especificam os requisitos funcionais e não funcionais de modo compreensível pelos usuários do sistema, ainda que não tenha conhecimentos técnicos. Descrevem o comportamento externo do sistema sob a perspectiva do usuário e são definidos por linguagem.

1.3.1 Requisitos de Sistema

São descrições mais detalhadas dos requisitos de usuário, são a base para um contrato destinado à implementação do sistema, sendo considerado o ponto de partida para o projeto do sistema.

2. ENGENHARIA DE SOFTWARE

Segundo Sommerville (2007), a engenharia de software é uma área que busca métodos e processos de desenvolvimento de sistemas, já que desenvolver sistemas pode se tornar uma tarefa mais complexa do que o esperado e muito difícil de ser compreendido, assim sendo necessário a utilização da engenharia de software para gerenciar os processos e fazer com que o projeto de um sistema possa acontecer dentro do esperado, ou seja, cumprindo com todos os requisitos determinados. A engenharia de software visa melhorar a qualidade e aumentar a produtividade na construção dos sistemas, pois é de fundamental importância que o desenvolvimento do sistema tenha custos adequados e resultados de alta qualidade. Sendo assim a engenharia de software proporciona boas formas de trabalho, como trabalhar em equipe utilizando os aspectos de processos de desenvolvimento de sistemas, ferramentas e técnicas apropriadas, sempre visando a solução do problema, levando em consideração as possíveis restrições organizacionais e financeiras.

A Engenharia de Software auxilia nos aspectos técnicos do processo de desenvolvimento de software, também contribui para o gerenciamento de projetos e o desenvolvimento de ferramentas que auxiliam todo um processo de software. A engenharia de software tem uma organização em seu trabalho, que foca na produção de software de alta qualidade e desempenho, procurando assim o método mais apropriado para cada caso. (SOMMERVILLE, 2007).

A Engenharia de Software se concentra nos aspectos práticos da produção de um sistema de software, enquanto a ciência da computação estuda os fundamentos teóricos dos aspectos computacionais.

2.1 Etapas de engenharia de software

Conforme Sommerville (2007), a engenharia de software dentro do projeto de sistema ajuda a entender como o processo é criado. As várias fases dentro da engenharia de software que são de extrema importância dentro do projeto. Será mostrado alguns modelos que são mais usados para estrutura um projeto de software. “A Engenharia de software é uma disciplina de engenharia relacionada com todos os aspectos da população de software, desde os estágios iniciais de especificação do sistema até sua manutenção, depois de este entrar em operação”.

Um processo de software é um conjunto de atividades que levam a produção de um produto de software. Essas atividades podem envolver o desenvolvimento de software propriamente dito, usando uma linguagem de programação como Java ou C. (SOMMERVILLE, 2007).

2.2 Modelos de Processos de Software

Segundo Sommerville (2007), a engenharia de software é uma disciplina relacionada com os aspectos da produção de software até sua manutenção e até depois de entrar em operação. As principais etapas dos modelos de processo de software são: Modelo Cascata, Desenvolvimento Evolucionário, Engenharia de software baseada em componentes. Dentro do processo de software existe também a iteração de processo e atividades do processo. No projeto a iteração de processo é menos utilizada pela equipe de desenvolvimento a forma de trabalhar com esses modelos seja mais demorada e complexa.

Um modelo de processo de software é uma representação abstrata de um processo de software. Cada modelo de processo representa um processo sob determinada perspectiva e, dessa forma, fornece somente informação parciais sobre esse processo. (SOMMERVILLE, 2007).

2.2.1 Modelo Cascata

O Modelo Cascata é conhecido também como ciclo de vida de software, este modelo como o nome diz é o ciclo de vida do sistema, até o momento que tiver algum membro da equipe disposto a mexer no projeto ele tem um ciclo. Caso algum processo não seja bem elaborado os outros processos pode ser afetados. O Modelo apresenta algumas desvantagens, caso precise fazer uma correção no projeto terá que fazer todo o projeto primeiro para depois ir voltando corrigindo, ou seja, caso a equipe desenvolvedora precise acrescenta ou corrigir falhas encontradas no andamento do projeto não será possível, neste modo o custo para a empresa acaba mais caro e atrasa o projeto.

As vantagens do modelo em cascata consistem na documentação produzida em cada fase e sua aderência a outros modelos de engenharia. Seu maior problema é a divisão inflexível do projeto em estágio distintos. Os compromissos devem ser assumidos no estágio inicial do processo, o que torna difícil reagir as mudanças de requisitos do usuário. (SOMMERVILLE, 2007).

2.2.2 Desenvolvimento Evolucionário

O modelo Evolucionário não é um modelo mais convencional a ser utilizado em projetos, nele o usuário participa do desenvolvimento no início até o fim, expondo sua opinião o que precisa ser mudado ou não. O projeto torna demorado e é perigoso passar do tempo planejado.

“O desenvolvimento evolucionário baseia-se na ideia de desenvolvimento de uma implementação inicial, expondo o resultado aos comentários do usuário e refinando esse resultado por meio de várias versões até que seja desenvolvido um sistema adequado”. (SOMMERVILLE, 2007).

Esse processo acaba não sendo viável porque a cada etapa que ele for concluído é mostrada ao usuário final, e o usuário sempre terá motivos para não continuar o processo, sempre mostrando ideias novas, desde modo atrasando o termino do projeto.

2.2.3 Engenharia de software baseada em componentes

A engenharia baseada em componentes, ela é mais simples, porém esse processo não dá para voltar caso precise voltar para rever alguns requisitos, ele não oferece revisar. Em projetos ele não é muito usado. Essa engenharia está voltada para projetos menores onde ele não precise passar por correções futuras.

Conforme Sommerville (2007), Engenharia de software baseada em componentes tem a vantagem óbvia de reduzir a quantidade de software a ser desenvolvida, e dessa maneira, reduzir os custos e riscos.

2.3 Requisitos de Sistema

Os requisitos dentro do projeto é fundamental porque são neles que descobre o que terá dentro do sistema. Os requisitos podem ser funcionais e não funcionais. Sem os requisitos não existe sistema, pois os requisitos é a “cabeça” do sistema.

Para Sommerville (2007), as definições dos requisitos do sistema definem o que deve fazer. Com as análises dos requisitos de software envolve fazer entrevistas com os clientes e usuários que usaram o software direta ou indiretamente.

As definições de requisitos de sistema especificam o que o sistema deve fazer (suas funções) e suas propriedades essenciais e desejáveis. Como na análise de requisitos de software, a criação das definições dos requisitos de sistema envolve consultas aos clientes e usuários finais do sistema. (SOMMERVILLE 2007).

Segundo Pressman (1995), os softwares são criados por vários mecanismos de conversões que mapeiam o que o cliente precisa para o código executável em máquina.

Conforme Pressman (1995), Os componentes de software são criados por meio de uma série de conversões que mapeiam as exigências do cliente para código executável em máquina.

Os componentes de softwares são mecanismos que são usados na programação para facilitar o desenvolvimento de uma aplicação. Um exemplo bem simples é a criação de relatórios dentro do sistema, para fazer os relatórios são necessários os componentes que auxilia no desenvolvimento do software.

O projeto de software é convertido numa forma de linguagem que especifica a estrutura de dados do software, os atributos procedimentais e os requisitos relacionados. A forma de Linguagem é processada por um tradutor que o converte em instruções executáveis em máquina. (PRESSMAN, 1995).

Para desenvolver um relatório são usados os componentes que ajuda a equipe a ter melhor desempenho no resultado para o cliente, desse modo, a equipe dá o máximo de esforço possível pelo sistema para criar um sistema completo onde atenda a necessidade do cliente, seja simples de mexer, no entanto qualquer um que não tenha conhecimentos técnicos na área consiga mexer sem problema.

Dentro do projeto de software existe a parte da programação onde a equipe de desenvolvimento da vida ao sistema, ou seja, se cria as rotinas que o usuário fará para resolver seu problema. Essa tradução que foi citada a cima refere a transformação de uma linguagem de programação para um sistema que o usuário entenda e consiga operar sem um conhecimento técnico da linguagem em si.

2.3.1 A especificação de requisitos de software

A especificação de requisitos de software serve para filtrar o desenvolvimento que são atribuídos ao sistema, para estabelecer uma descrição completa da informação dos requisitos e restrições, também ajuda nos critérios de validação e outros dados que venha a persistir aos requisitos.

A especificação de Requisitos de software é produzida no auge da tarefa de análise. A função e o desempenho atribuídos ao software como parte da engenharia do sistema são refinados ao se estabelecer uma descrição completa da informação, uma indicação dos requisitos de desempenho e restrições de projeto, critérios de validação apropriados e outros dados pertinentes aos requisitos. (PRESSMAN, 1995).

2.4 Gerenciamentos de Projetos

O gerenciamento do projeto é uma parte fundamental da engenharia de software, pois é dele que dependerá o andamento do projeto, ou seja, caso o projeto for bem gerenciado a probabilidade do projeto ser mal sucedido é menor do que se for mal gerenciado. Quando o projeto é mal gerenciado pode ocorrer várias falhas como: O atraso da entrega do software, o custo do desenvolvimento do software poderá sair do previsto. O gerenciamento de projeto então é primordial, dentro do projeto pois é ele que fica responsável pelo fracasso ou sucesso do projeto.

Os gerentes de software são responsáveis pelo desenvolvimento de planos e cronogramas do projeto. Eles supervisionam o trabalho para assegurar que ele esteja sendo realizado dentro dos padrões exigidos e monitoram o progresso para verificar se o desenvolvimento está no prazo e dentro do orçamento. (SOMMERVILLE, 2007).

O gerente possui uma das tarefas mais complicadas dentro do projeto, pois ele precisa entender de todas as áreas técnicas dentro do projeto e mais algumas outras como: Administração, finanças. Para poder administrar o projeto e que nada possa dar errado no decorrer de seu desenvolvimento. O gerente no início do projeto já terá que saber tudo sobre o projeto que está sendo criado, para quando estiver algum problema futuro consiga resolver de imediato.

No início do planejamento de projeto deve ser feito uma análise completa do problema, pois é nela que o gerente terá que prever futuros problemas e já ter algumas soluções para que mais na frente não tenha surpresa.

O gerenciamento de um projeto de software depende de um planejamento minucioso do progresso do projeto. Os gerentes devem prever os problemas que podem ocorrer e preparar soluções experimentais para esses problemas. Um plano elaborado no início de um projeto deve ser usado como guia. Esse plano inicial deve ser o melhor possível em face das informações disponíveis. Ele deve evoluir à medida que o projeto progride e melhores informações se tornem disponíveis. (SOMMERVILLE, 2007).

Todo projeto precisa de um plano a ser seguido, esse plano é elaborado antes do projeto ter início é onde a equipe senta e analisa qual projeto será desenvolvido, e a partir do momento que começa a ser desenvolvido vai surgindo novas ideias para melhora. E o plano que deu origem ao projeto terá que sofrer alterações dependendo do seu tamanho. Esse plano é todas as informações coletadas através do cliente, devido isso deverá ser modificado no decorrer do projeto.

2.4.1 Cronograma do projeto

No desenvolvimento do projeto o cronograma é uma das partes mais difíceis de elaborar, pois é nele que está o tempo que o projeto irá demorar para ser finalizado. Existe uma pessoa responsável só para criar esta ferramenta, que é o gerente de projeto, ele fica designado para estimar o tempo de um projeto de acordo com as atividades que devem ser executadas.

O desenvolvimento do cronograma do projeto é um dos trabalhos mais difíceis para um gerente de projeto. Os gerentes estimam o tempo e recursos necessários para concluir atividades, organizando-as em sequência coerente. A menos que o projeto cujo cronograma está sendo desenvolvido seja similar a um projeto anterior, as estimativas anteriores constituem uma base incerta para o desenvolvimento do cronograma do novo projeto. A estimativa de cronograma é mais complicada pelo fato de projetos diferentes podem usar métodos e linguagens de implementação diferentes. (SOMMERVILLE, 2007).

O desenvolvimento do projeto deve ser bem elaborado, pois se for planejado rápido, poderá haver falhas que os desenvolvedores não estavam esperando.

Dentro do cronograma existe a estrutura analítica, as dependências de atividades e as alocações de pessoal, ou seja, esses conjuntos são interligados no entanto cada equipe ficará responsável em executar a tarefa no tempo previsto de início e fim.

3 ANÁLISE ORIENTADA A OBJETOS

Análise orientada a objeto (AOO) tem o objetivo de encontrar os objetos que representem os problemas a serem resolvidos e a seguinte característica básica, a unificação de análise, projeto e programação de sistemas de informação baseados na composição de diversas unidades de programas chamados de objetos.

De acordo com Bezerra (2007), objeto pode ser formado de coisas do mundo real com responsabilidades específicas que possui atributos e métodos comuns. E classe é um conjunto de objetos que possui atributos semelhantes, comportamentos comuns, os mesmos tipos de relacionamentos com os outros objetos e que herdaram atributos de outras classes.

A orientação a objeto é como uma técnica na modelagem de sistemas que diminui a diferença semântica entre a realidade da modelagem em andamento e também os modelos já construídos. Assim a orientação a objeto é uma forma diferente de abordar um problema, pois ela estabelece modelos de acordo com tais problemas enfrentados. Pois o seu principal princípio na construção de um sistema é que agentes autônomos possam interagir entre si para que o objetivo seja alcançado.

O paradigma da orientação a objeto visualiza um sistema como uma coleção de agentes interconectados chamados objetos. Cada objeto é responsável por realizar tarefas específicas. É pela interação entre objetos que uma tarefa computacional é realizada. (BEZERRA, 2007).

4 UML

Conforme Guedes (2009) “a UML surgiu de três métodos de modelagem: o método de Booch, o método OMT (Object Software Technique), de Jacobson, e o método OOSE (Object-Oriented Software Engineering) de Rumbaugh”. Na década de 1990 eram esses os métodos de modelagem orientada a objetos mais usados popularmente entre os profissionais da área de desenvolvimento de software. Com a união desses métodos, graças ao trabalho de Booch, Jacobson, Rumbaugh, conhecidos como “Os Três Amigos” resultou-se no lançamento, em 1996, da primeira versão da UML propriamente dita.

A UML foi tão bem aceita que em 1997, apenas um ano depois de sua primeira versão foi adotada pela OMG (Object Management Group ou Grupo de Gerenciamento de Objetos), como uma linguagem padrão de modelagem. De acordo com Guedes (2011), a versão 2.0 da linguagem UML foi lançada no ano de 2005, atualmente encontra-se na versão 2.4beta.

A UML é uma linguagem visual utilizada para modelar sistemas computacionais por meio da Orientação a Objetos. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem de sistemas adotada internacionalmente. Com características marcantes, onde se pode modelar qualquer sistema, tornando-se possível ser modelado corretamente, com consistência, fácil de comunicação com as outras aplicações, simples de ser atualizado e compreensível. A UML é uma linguagem de modelagem e não linguagem de programação.

A UML é uma linguagem visual para modelar sistemas orientados a objetos. Isso quer dizer que a UML é uma linguagem que define elementos gráficos (visuais), que podem ser utilizados na modelagem de sistemas. Esses elementos permitem representar os conceitos do paradigma da orientação a objetos. Através dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas de um sistema. (BEZERRA, 2007).

4.1 Diagrama de Caso de Uso

De acordo com Bezerra (2007), a modelagem de um diagrama de caso de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema, de forma a demonstrar a linguagem entre o autor e o sistema para melhor entendimento de quem observa externamente o sistema. Os atores representam papéis que podem ser usados por usuários do sistema. Casos de uso são descrições de interações entre alguns atores exteriores e o sistema computacional. Assim, diagramas de casos de uso são descrições gráficas dos relacionamentos

entre atores e casos de uso e entre um caso de uso e outro caso de uso.

Os diagramas de casos de uso são importantes para visualizar, especificar e documentar o comportamento de um elemento. Esses diagramas fazem com que sistemas, subsistemas e classes fiquem acessíveis e compreensíveis, por apresentarem uma visão externa sobre como esses elementos podem ser utilizados no contexto. Os diagramas de caso de uso também são importantes para testar sistemas executáveis por meio de engenharia de produção e para compreendê-los por meio de engenharia reversa. (BOOCH, RUMBAUGH, JACOBSON, 2006).

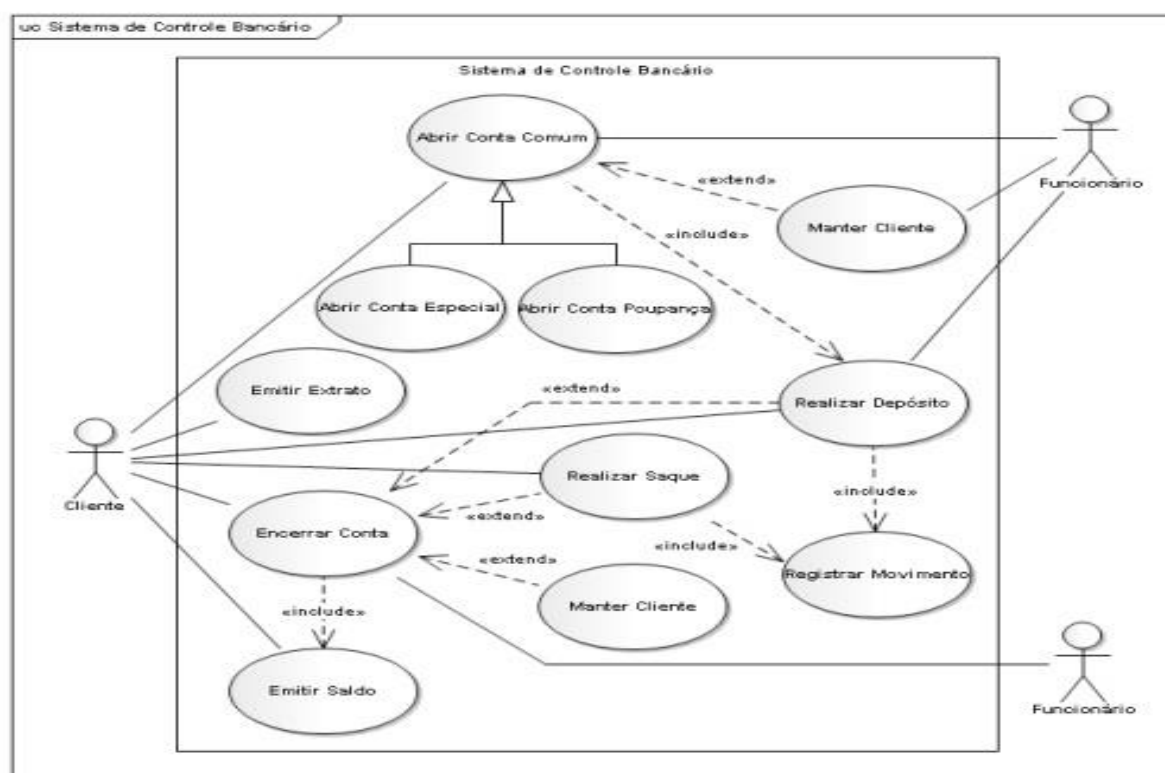


Figura 1 Exemplo: Diagrama de caso de uso. Fonte: Gilleanes (2011).

4.2 Diagrama de Sequência

Diagrama de sequência é um diagrama usado em UML, representando a sequência de processos (mais especificamente, de mensagens passadas entre objetos) num programa de computador. Como um projeto pode ter uma grande quantidade de métodos em classes diferentes, pode ser difícil determinar a sequência global do comportamento. O diagrama de sequência representa essa informação de uma forma simples e lógica.

Um diagrama de sequência descreve a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo. Ele registra o comportamento de um único caso de uso e exibe os objetos e as mensagens passadas entre esses objetos no caso de uso.

O diagrama de sequência é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso definido pelo diagrama de mesmo nome e apoia-se no diagrama de classes para determinar os objetos das classes envolvidas em um processo. Um diagrama de sequência costuma identificar o evento gerador do processo modelado, bem como o ator responsável pelo evento, e determina como o processo deve se desenrolar e ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos. (GUEDES, 2011).

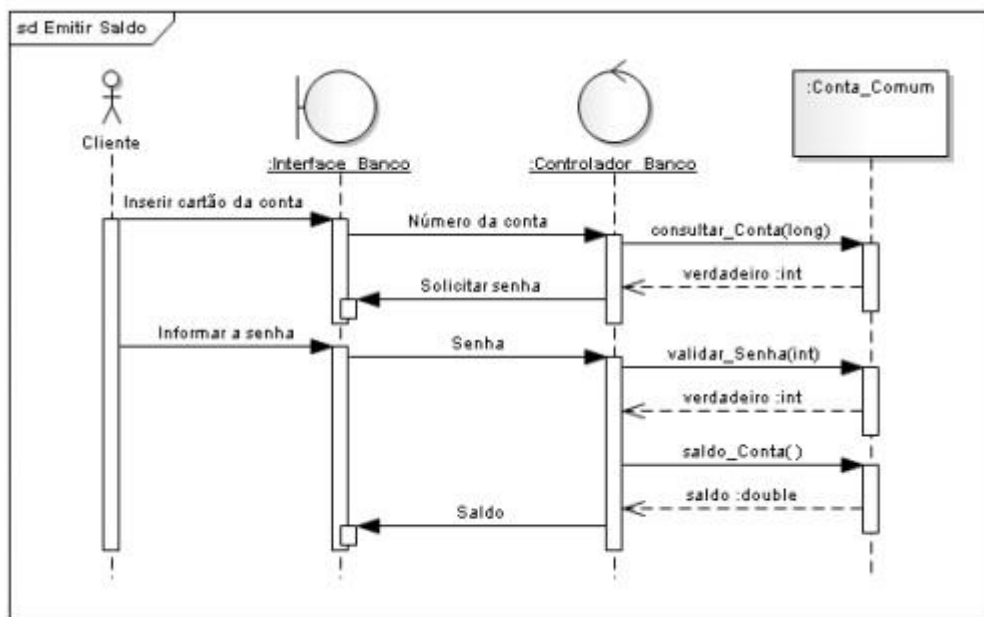


Figura 2 Exemplo: Diagrama de Sequência Fonte: Gilleanes (2011).

4.3 Diagrama de Classes

O diagrama de classe é um dos diagramas mais encontrados na modelagem de sistemas orientados a objetos e na UML. É utilizado na definição e visualização das classes que pertenceram ao sistema determinando os atributos e métodos que cada classe contém e ainda demonstra como as classes se relacionam. Tem por finalidade auxiliar o entendimento da estrutura do sistema, através de um conjunto de classes que atenderá as funcionalidades do sistema.

Os diagramas de classes são importantes não só para a visualização, a especificação e a documentação de modelos estruturais, mas também para a construção de sistemas executáveis por intermédio de engenharia de produção e reversa. (BOOCH, RUMBAUGH, JACOBSON, 2006).

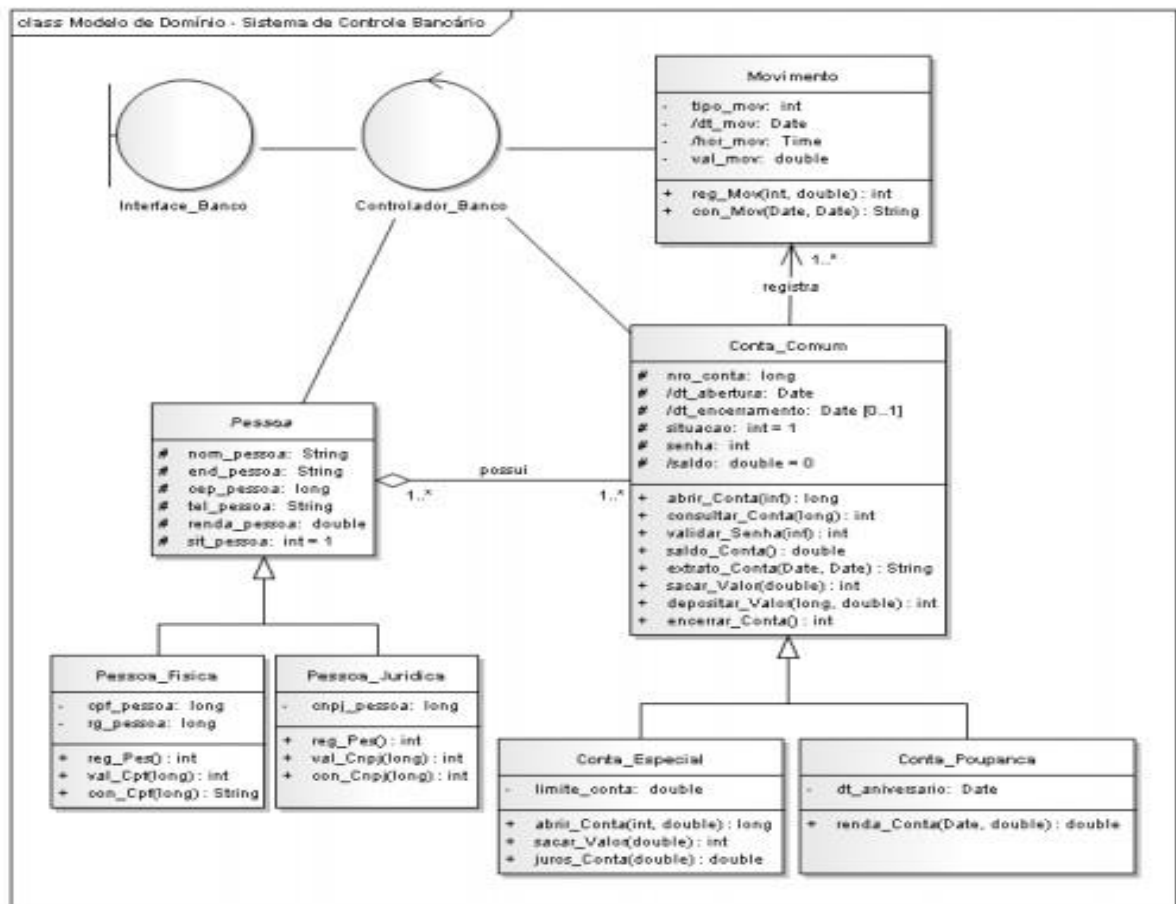


Figura 3 Exemplo: Diagrama de Classes: Gilleanes (2011).

4.3.1 Dependência

Segundo Gilleanes (2011), Dependência é um relacionamento que identifica a dependência de uma classe em relação à outra. A dependência é representada por uma linha tracejada entre duas classes, contendo uma seta apontando para a classe da qual a classe posicionada na outra extremidade do relacionamento é dependente.

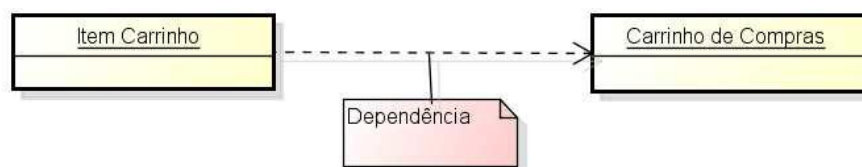


Figura 4 Exemplo: Dependência Fonte: Gilleanes (2011).

4.3.2 Generalização/Especialização

Segundo Gilleanes (2011), Generalização/Especialização é um tipo de relacionamento especial, tem por objetivo representar a ocorrência de herança entre as classes, identificar as classes mães, chamadas de gerais e classes filhas, chamadas especializadas, demonstrando a hierarquia entre as classes e possivelmente métodos polifórmicos nas classes especializadas. Este tipo de relacionamento ocorre quando existem duas ou mais classes semelhantes, para evitar redundância de atributos e métodos, e cria-se uma nova classe genérica que declare os atributos e métodos que existam em todas as classes envolvidos no processo.

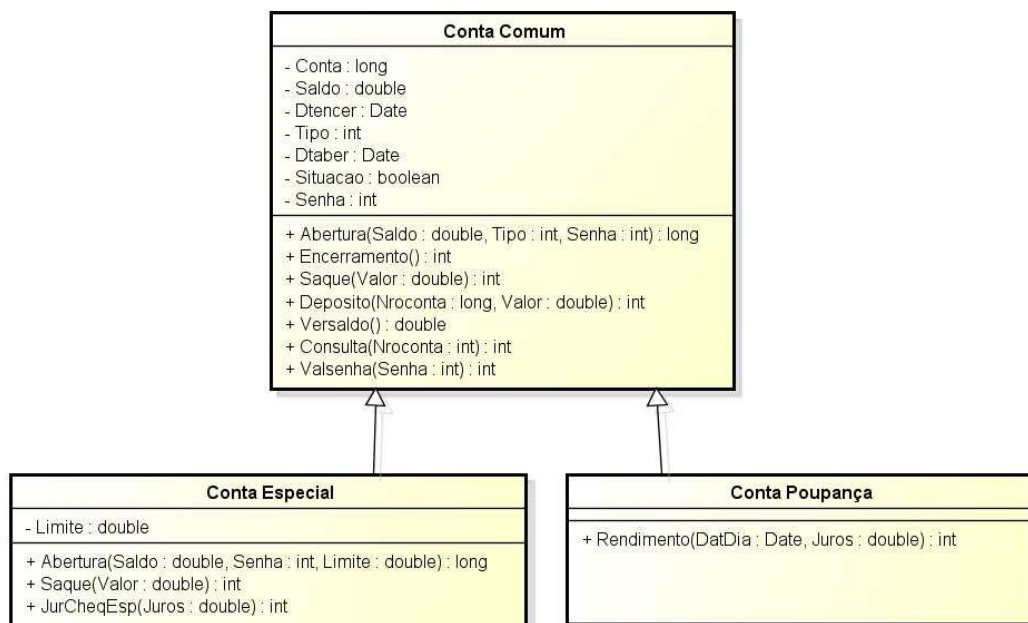


Figura 5 Exemplo: Especialização/Generalização Fonte: Gilleanes (2009).

4.3.3 Associação

Associação é um tipo de relacionamento entre classes, que servem para compartilhar informações e métodos, podendo identificar algum nível de dependência entre as classes.

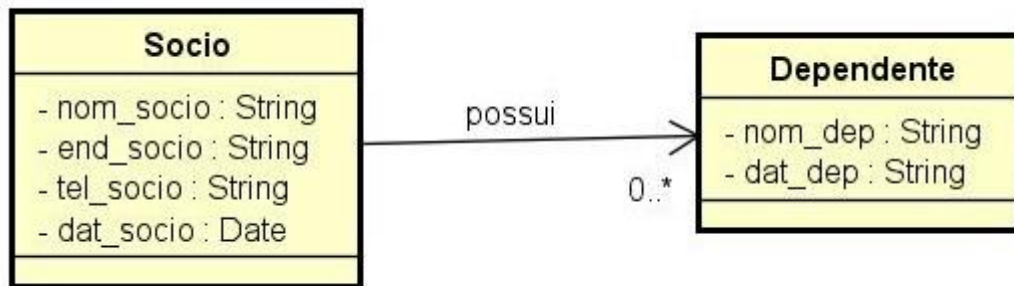


Figura 6 Exemplo: Associação Fonte: Gilleanes (2011).

4.3.4 Agregação

Conforme Gilleanes (2011) agregação é um tipo especial de associação em que um objeto é parte de outro, de tal forma que a parte pode existir sem o todo. Uma agregação consiste de objeto contendo referências para outros objetos, de tal forma que o primeiro seja o todo, e que os objetos referenciados sejam as partes do todo.

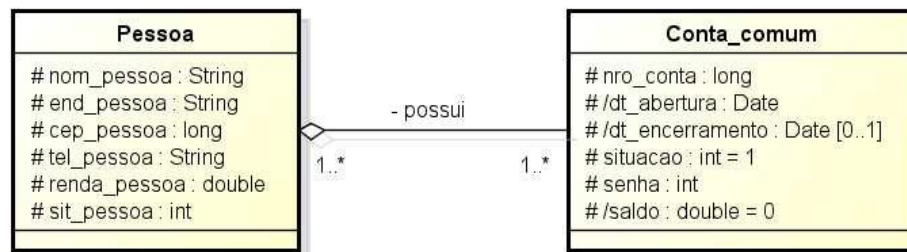


Figura 7 Exemplo: Agregação Fonte: Gilleanes (2011).

4.3.5 Composição

De acordo com Gilleanes (2011), composição é mais um tipo de associação que constitui uma variação da agregação, onde o objeto todo e o objeto parte possuem vínculos, ou seja, eles precisam estar associados.

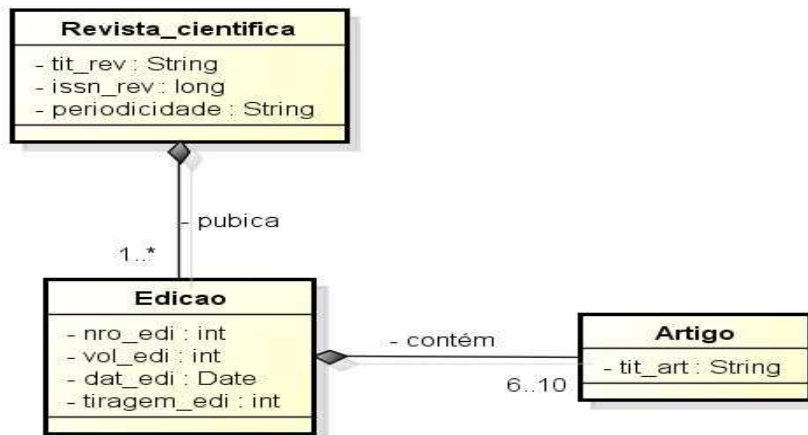


Figura 8 Exemplo: Composição Fonte: Gilleanes (2011).

5 E-COMMERCE

O e-commerce, que em português significa comércio eletrônico, é uma modalidade de comércio que realiza suas transações financeiras por meio de dispositivos e plataformas eletrônicas, como computadores e celulares. Um exemplo deste tipo de comércio é comprar ou vender produtos em lojas virtuais.

No início, o e-commerce era utilizado basicamente para vender bens tangíveis com valores modestos, como: livros e CDs. Hoje, ele é utilizado para comercializar desde produtos que custam milhões, como: iates, carros de luxo e mansões, até produtos que há pouco tempo eram inimagináveis pela sua incompatibilidade com este tipo de comércio, como roupas, perfumes e alimentos.

Do ponto de vista do empresário, E-Commerce ou Comércio Eletrônico, significa simplificação das operações e procedimentos administrativos, controle de entrada e saída de produtos, redução nos custos de pessoal e de recursos físicos, e muitos outros benefícios que vão depender do modelo adotado. Do ponto de vista do consumidor este tipo de comércio visa sobretudo reduzir custos, na medida em que o cliente pode adquirir produtos que deseja e/ou necessita a preços mais reduzidos, sem ter que gastar dinheiro com deslocamentos, trânsito, filas e tudo o que está envolvido numa compra tradicional e ainda algo bastante importante, o fato de poder adquirir algo numa parte distante do globo.

De acordo com Laudon (2007), podemos classificar os três principais tipos de categorias, de acordo com as transações de comércio eletrônico, levando em conta a natureza das participações da transação”.

Sob essa perspectiva, as três principais categorias de comércio eletrônico são: empresa-consumidor, empresa-empresa e consumidor-consumidor.

Abrir as páginas dos jornais e sites de notícia de economia está um pouco desesperador nos últimos meses em nosso país. Alta dos juros, quase nenhuma previsão de crescimento – são dados que levam empreendedores a pensar duas vezes antes de realizar qualquer tipo de investimento. Apesar da crise financeira que vive o Brasil, existem setores que estão em alta e um deles é o e-commerce. Segundo o E-Bit, o varejo online brasileiro deve faturar neste ano R\$ 43 milhões. Como diz o ditado: “quando a esmola é demais, o santo desconfia”. Contudo, não há motivos para desconfiar do comércio eletrônico brasileiro. Existem razões que explicam por que o setor consegue driblar a crise e trazer lucros para quem é da área.

Alves (2002) que classifica, por exemplo: empresa-governo, como modelo de

comércio eletrônico, em virtude das transações que envolvem empresas e entidades governamentais na atualidade.

O e-commerce está em ascensão em todo o mundo, mas aqui no Brasil o cenário é mais especial. Levamos em consideração, em primeiro lugar, que é cada vez maior o número de brasileiros com acesso à internet por meios de desktops ou de dispositivos móveis. Mas apenas isso não explica o sucesso do setor. O principal motivo para o crescimento do varejo online no país é a criatividade dos empreendedores brasileiros. Diferente do que acontece na maioria dos outros países, os empresários brasileiros não se contentam com o básico e conseguem encontrar soluções e estratégias que encantam o consumidor, o que reverte em vendas. Isso pode acontecer em diversas etapas da operação de e-commerce: em uma campanha de marketing, na forma de pagamento, no e-commerce de nicho, etc.

Desde o aparecimento das primeiras redes informáticas até ao lançamento do correio eletrônico em 1971, da web em 1991, dos browsers em 1993 e dos populares serviços online (em linha) em 1995, a Internet alterou a forma como milhares de pessoas passaram a lidar com as suas vidas pessoais e profissionais.

Podemos observar também que o varejo online, em tempos de crise, é mais competitivo do que o tradicional. Com o orçamento mais apertado, é natural que o consumidor pesquise muito mais antes de realizar uma compra. Neste contexto, o comércio eletrônico é muito mais vantajoso, pois oferece inúmeras possibilidades de comparação de preços (e de maneira muito mais ágil e prática). Além disso, o varejista consegue desenvolver campanhas especiais para atrair o público que entra em sua loja, pesquisa, mas não efetua a compra. Nessas ações, a loja virtual relaciona-se diretamente com o cliente, a conquista e é capaz de finalizar a venda.

Outra vantagem do e-commerce que sempre o coloca à frente do varejo tradicional é a capacidade de conhecer o que o público alvo deseja. Campanhas de monitoramento do público servem para fornecer informações do comportamento e do desejo do consumidor. Com esses dados em mãos, é possível oferecer para um público específico exatamente o que ele deseja no momento exato. Oferecer um modelo de compra personalizado de acordo com a necessidade do usuário aumenta as possibilidades de venda, afinal, a loja virtual joga com a real necessidade ou vontade do usuário.

Sempre enfatizo que não basta apenas estar no e-commerce para lucrar. É preciso estar atento ao mercado, conhecer suas especificidades, oportunidades e desafios e ter a capacidade de se remodelar, de planejar novas estratégias. Ganha quem mantiver sempre esta mentalidade

e souber aproveitar o que o setor oferece.

5.1 Business-To-Business - B2B (Empresa-Empresa)

Esta é uma categoria bastante importante do E-Commerce, podemos dizer que é nas transações eletrônicas entre empresas que se gera o maior volume de negócios deste tipo de comércio, visto que também tem maiores oportunidades para melhores relacionamentos com o cliente, por meio de marketing.

Segundo Alves (2002), os serviços do B2B têm como ponto central a introdução de facilidades e de recursos que permitem a realização de transações de negócios on-line entre as companhias.

Pode representar uma grande oportunidade de redução de custos e ganhar vantagens competitivas pelo compartilhamento da economia de escala da cadeia de suprimentos, com a integração de fornecedores de produtos e serviços, provedores de redes de valor agregado e da Internet.

5.2 Business-To-Consumer - B2C (Empresa-Consumidor)

Este tipo de comércio tem vindo a desenvolver-se de forma acentuada, existindo já várias lojas e centros comerciais virtuais na World Wide Web, tendo como principal característica a de permitir às empresas ofertarem produtos e serviços diretamente aos consumidores por meios eletrônicos, desta forma, qualquer bem de consumo ou serviço pode ser vendido pela internet, está agindo como um elemento que facilita as atividades de pré ou pós-venda, associando grande valor ao processo de venda.

Segundo Cunningham (2000), a evolução do mercado B2C tem sido rápido e feroz, enquanto o estado atual deste mercado é de crescimento e confusão.

As empresas que esperaram para ver que a Internet era simplesmente uma moda passageira fecharam, enquanto outras despertaram para ver novos desafios trazidos pela velocidade das informações. Os estudos continuam a confirmar que os consumidores utilizam a web como uma ferramenta de pesquisa, mas muitos ainda estão comprando com frequência nas lojas tradicionais.

5.3 Business-To-Administration - B2A (Empresa-Administração)

É um segmento do comércio eletrônico que apesar de se encontrar ainda numa fase inicial de expansão, tem tendência a aumentar rapidamente, nomeadamente com a promoção do comércio eletrônico na administração Pública e com os mais recentes investimentos.

5.4 Consumer-To-Administration - C2A (Consumidor-Administração)

Atinge todas as transações eletrônicas efetuadas entre os indivíduos e a Administração Pública. Sendo várias as áreas de aplicação: Segurança Social (através da divulgação de informação e realização de pagamentos), Saúde (marcação de consultas, informação sobre doenças e pagamento de serviços), Educação (divulgação de informação e formação à distância), Impostos (entrega das declarações e pagamentos).

5.5 Outros Tipos De Comércio Eletrônico:

5.5.1 Consumer-To-Consumer - C2C (Consumidor-consumidor):

Os consumidores negociam entre si, como por exemplo os leilões, onde os consumidores disputam um produto.

5.5.2 Consumer-To-Business - C2B (Consumidor-empresa):

Quando o consumidor envia sua proposta para várias empresas que entram em concorrência para conquistá-lo. Refere, portanto, ao e-commerce em que a iniciativa parte do consumidor.

O Projeto de lei nº 979/07 elaborado pelo deputado federal Chico Alencar PSOL/RJ visa acrescentar artigo à Lei nº 8.078, de 11 de setembro de 1990 – Código de Defesa do Consumidor, “para obrigar os fornecedores que ofertam ou comercializam produtos ou serviços pela rede mundial de computadores a informar seu endereço para fins de citação, bem como o número de telefone e endereço eletrônico utilizáveis para atendimento de reclamações de consumidores”.

6 BANCO DE DADOS

O banco de dados é fundamental dentro de um sistema seja ele desktop ou web, são neles que ficam armazenados os dados que serão usados em outro momento. Se o aplicativo não usa banco de dados quando for fechado a aplicação, a empresa poderá os dados quando o sistema finaliza.

Segundo Toby (2006), a tecnologia de banco de dados tem evoluído rapidamente nas últimas três décadas, desde a ascensão e o conseqüente domínio dos sistemas de bancos de dados relacionais.

O Modelo Entidade Relacionamento básico possui três classes de Objetos que são denominados:

- Entidades: onde os principais dados são coletados podendo representar pessoas lugares, ou algum evento que possa surgir interesse.
- Relacionamentos: Representa associações do mundo real, são as ligações entre as entidades, funciona como o conector para que duas ou mais entidades possa compartilhar informações de dados.
- Atributos: São os valores que a entidade terá, por exemplo: id_cliente, nome_cliente, etc, ou seja, são as características da entidade.

De acordo com Toby (2006), entidades são os principais objetos de dados sobre os quais informações devem ser coletados; elas normalmente representam uma pessoa, lugar, coisa ou evento de interesse informativo. Uma ocorrência específica de uma entidade é chamado de instancia da entidade, ou, às vezes, ocorrência da entidade.

Os Relacionamentos se tratando de bancos de dados é a ligação das tabelas: cliente com produto. Tornando uma associação do mundo real com o mundo digital.

Segundo Toby (2006), relacionamentos representam associações do mundo real entre um ou mais entidades e dessa forma, não possuem existência física ou conceitual além de sua dependência das entidades associadas.

Os Atributos dentro do banco de dados está relacionado aos campos que o sistema poderá receber valores que o usuário irá digitar, desde modo esses dados, futuramente virará uma informação necessária ou não para o funcionário da empresa.

Atributos são características de entidades que oferecem detalhes descritivos sobre elas. Uma ocorrência em particular de um atributo dentro de uma entidade relacionamento é chamada de valor de um atributo. Os atributos de uma entidade como o da entidade Funcionário podem incluir id_func, nomefunc. (Toby, 2006).

Os atributos são classificados em dois tipos que são: Identificadores e descritores. O identificador pode ser chamado também como chave primária, essa chave é responsável pelo restante dos atributos. Os descritores são as não chaves para especificar uma característica instância da entidade.

Existem dos tipos de atributos: identificadores e descritores. Um identificador (ou chave) é usado para determinar exclusivamente uma instância de uma entidade; um descritor (ou atributo não-chave) é usado para especificar uma característica não – exclusiva de determinada instância da entidade. (Toby, 2006).

Os identificadores são chaves principais, são encarregados de “cuidar” dos descritores um exemplo de identificador é o código_cliente nele são possíveis se ter um controle do resto dos descritores, o descritor fica sendo nome_cliente ,cpf_cliente.

6.1 Normalização

A normalização existe devido os bancos de dados do projeto serem muitos extensos, e conterem muitas repetições. A normalização ajuda a tirar as redundâncias que o banco de dados tiver, facilitando a manipulação dos dados dentro da tabela e até na hora das consultas. A normalização possui três formas:

Segundo Toby (2006), a 1º forma diz: “Uma tabela estará na primeira forma normal se, e somente se, todas as colunas tiverem apenas valores atômicos, ou seja, se cada coluna só puder ter um valor para cada linha na tabela.”.

Segundo Toby (2006), a 2º forma diz: “Uma tabela está na segunda forma normal se, e somente se, ela estiver na 1º forma normal e os atributos não chaves forem totalmente dependentes da chave primaria. Um Atributo será totalmente dependente da chave primaria se estiver no lado direito de um DF que tem no lado esquerdo a própria chave primaria ou algo que possa ser derivado da chave primaria usando a transitividade das DFs”.

Segundo Toby (2006), a 3º forma diz: “Uma tabela está na terceira forma normal se, e somente se para cada dependência funcional não trivial $X \rightarrow A$, onde X e A são atributos simples ou compostos, uma das duas condições precisam ser mantidas ou o atributo X é uma super chave, ou o atributo A é membro de uma chance candidata. Se o atributo A é uma chave candidata, A é chamada de atributo primo”.

7 PHP

PHP significa “Hypertext Preprocessor”, e é uma linguagem interpretada principalmente utilizada no desenvolvimento WEB. Sua Sintaxe lembra um pouco a sintaxe do C e do Perl e é uma linguagem bem fácil de aprender.

Uma linguagem de programação é considerada compilada, quando ela precisa de um programa de computador chamado compilador, que transforma o código escrito pelo programador num arquivo que pode ser executado pelo usuário. A maioria dos programas que você possui no seu computador provavelmente foi escrito usando uma linguagem compilada.

O PHP é um conhecido exemplo de linguagem server side script, ao passo que o JavaScript é um exemplo de client side script. Ambas podem ser inseridas em páginas HTML.

As linguagens de programação de script são amplamente usadas, hoje, para realizar tarefas ou construir softwares complexos;

Este sistema de licença não traz lucro aos desenvolvedores, pois estes disponibilizam tudo para o público gratuitamente, e o público, por sua vez, ajuda reportando erros e ajudando a modificar o código fonte. Muitas empresas apoiam os desenvolvedores do PHP, pois estes não visam lucro ao criarem e desenvolverem o programa. O PHP é muito usado com o Linux e o MySQL, dois outros programas Open Source. O PHP tem código fonte disponível para qualquer pessoa, sem custos. A licença de uso e edição é Open Source, ou seja, ninguém pode comercializar qualquer versão modificada do PHP, e qualquer modificação deve continuar com o código fonte aberto para os usuários explorarem e modificarem. O PHP se diferencia de outros scripts CGI porque ao invés de se escrever um monte de comandos para imprimir os HTML, é escrito um arquivo HTML com os códigos PHP embutidos entre o HTML delimitado por tags de início e fim.

Contudo, a grande quantidade de operações necessárias para executar-se cada um destes procedimentos fez com que o PHP apresentasse algumas limitações ao projeto, derivadas das características de linguagem de alto nível, onde ele se enquadra, tais como:

- Tamanho máximo do arquivo de imagem em torno de 200 kbytes;
- Velocidade de processamento;
- Possibilidade de utilização de imagens somente nos formatos GIF, JPEG, PNG e BMP.

O PHP é uma linguagem de programação server-side scripts (scripts executados no servidor). Um exemplo de server-side scripts são os sistemas de busca da internet (www.google.com.br, www.yahoo.com.br, etc...).

Por não se tratar de uma linguagem de "programação" propriamente dita (trata-se de uma linguagem descritiva que tem como objeto dar formato ao texto e às imagens que se pretende visualizar no navegador), o HTML necessita de linguagens que complementem e supram as suas limitações, como é o caso do PHP (BHON, 2004).

As linhas de código PHP são embutidas no código HTML (HiperText Markup Language). O HTML é a linguagem padrão para criação de páginas de Internet. Como o próprio significado da sigla informa, o HTML é capaz de proporcionar hiper-textos. Hiper-texto é um modo que proporciona ao usuário uma maior interação com textos de uma página web, onde informações são interligadas intuitiva e associativamente. Através de saltos - que marcam o movimento do hipertexto - o leitor assume um papel ativo, sendo ao mesmo tempo co-autor.

Quando o usuário acessa uma página PHP por meio de seu browser (navegador), todo o código PHP é executado no servidor, e os resultados são enviados para seu navegador. Portanto, o navegador exibe a página já processada, sem consumir recursos de seu computador. As linhas de programação PHP não podem ser vistas por ninguém, já que elas são executadas no próprio servidor, e o que retorna ao usuário é apenas o resultado do código executado.

7.1 Vantagens do PHP

O cliente recebe apenas o resultado dos scripts, que são interpretados no servidor, não tendo acesso ao código. Muito simples de se aprender e trabalhar, o PHP atende desde os mais experientes desenvolvedores quanto os iniciantes na área. O PHP é multiplataforma, podendo ser usado na maioria dos Sistemas Operacionais, OpenSource, e diferente de scripts como o JavaScript, ele roda no servidor, que aliás é suportado pela maioria dos servidores WEB que existem hoje no mercado como o Apache, IIS, PWS, etc.

7.2 O que pode ser feito com o PHP?

PHP tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente, HTTP. Ainda é possível abrir sockets e interagir com outros protocolos.

Basicamente, qualquer coisa que pode ser feita por algum programa CGI pode ser feita também com PHP, como coletar dados de um formulário, gerar páginas dinamicamente ou enviar e receber cookies. PHP também tem como uma das características mais importantes o

suporte a um grande número de bancos de dados, como dBase, Interbase, mSQL, mySQL, Oracle, Sybase, PostgreSQL e vários outros. Construir uma página baseada em um banco de dados torna-se uma tarefa extremamente simples com PHP.

7.3 PHP com HTML

Hoje em dia, a linguagem PHP vai além da criação de sites dinâmicos, também sendo usada para a criação de grandes aplicações que rodam 100% na internet, onde o usuário precisa apenas de um navegador para acessar e fazer uso dessas aplicações.

Como a linguagem PHP cria fragmentos de HTML, ou até páginas Web inteiras, ela pode criar páginas com conteúdo dinâmico, que é modificado sempre que o usuário acessa a página, ou quando o usuário solicita alguma ação da página, como uma pesquisa num site ou compra de um produto em sites de comércio eletrônico.

A linguagem HTML nasceu para criação de páginas Web estáticas, onde o conteúdo não é modificado. Dessa forma, toda vez que um usuário acessa uma página Web estática, ele vê sempre a mesma página com o mesmo conteúdo.

Podemos dizer que PHP é uma linguagem de script que é executada no lado servidor. Como ela foi projetada para ser usada com HTML, PHP fornece mais flexibilidade que a linguagem HTML sozinha. E por fim, a PHP possui uma sintaxe muito parecida com C/C++, Java ou Perl.

8 ESTUDO DE CASO

8.1 Modelagem do Negócio

8.1.1 Tema

Estudo de Caso: Desenvolvimento de um sistema web para gerenciamento de locação de artigos para festas.

8.1.2 Apresentação do tema

Este trabalho aborda um estudo de caso que é um documento baseado nas necessidades de uma empresa real que atua na área de locação de artigos para festas. A empresa necessita de um controle que venha facilitar a locação, agilizando os processos e serviços. O estudo de caso terá como resultado a documentação, modelagem e implementação de um sistema web visando satisfazer as necessidades do cliente.

8.1.3 Objetivos

Elaborar a modelagem e documentação do sistema com necessidades reais, e implementar suas funcionalidades. Assim, será possível satisfazer as necessidades do cliente, e ainda aplicar os conhecimentos adquiridos no curso, tais como: linguagens de programação, análise de sistemas, engenharia de software e banco de dados, entre outros.

8.1.4 Justificativa

A motivação para produção desse estudo de caso é atender aos objetivos e apresentar uma solução para os problemas com organização de todos os controles, pois tudo é feito de forma manual podendo ocasionar perdas de informações sem contar as dificuldades para fazer qualquer tipo de pesquisa. Sabe-se a importância de identificar, analisar, modelar e documentar as necessidades informadas, pois essas ações têm como objetivo facilitar o desenvolvimento do software, assim solucionando os problemas encontrados no momento das entrevistas com cliente.

8.1.5 Metodologia e ferramentas utilizadas

A Pesquisa é qualitativa, aplicada através de um estudo de caso para um problema real.

Para aquisição das informações da empresa foi desenvolvido uma entrevista com o cliente, a fim de adquirir o conhecimento sobre a empresa, seu funcionamento, e permitindo o entendimento das necessidades e do negócio.

Para documentar e modelar o sistema foram utilizadas técnicas de Análise Orientada a Objetos e os diagramas da UML 2.0. No processo de implementação será utilizado a IDE de desenvolvimento Netbeans IDE 8.0.1, por se tratar de um aplicativo de uso público com uma gama de recursos que possibilita a implementação visual, na codificação será utilizada linguagem de programação PHP que por suas características é totalmente orientada a objetos e o banco de dados SQL MySQL Server 5.5, mesmo sendo uma versão gratuita possui recursos suficientes para armazenagem dos dados.

8.2 Apresentação da Empresa

A empresa Decoração Chuva de Ouro está situada na Rua 07 Qd. 20 Lote 03, Setor Parque Alvorada, no município de Itapuranga-Go. A empresa tem em sua propriedade um galpão onde são guardados todos os artigos a serem locados, e uma sala onde os documentos e papéis ficam armazenados.

8.3 Problema

A empresa encontra muitas dificuldades para realizar todos os controles necessários da mesma, tais: como controle de produtos e de locações. Atualmente, todo controle é feito manual através de anotações, a fim de evitar que acabe qualquer item para a realização da locação.

8.4 Proposta de Solução

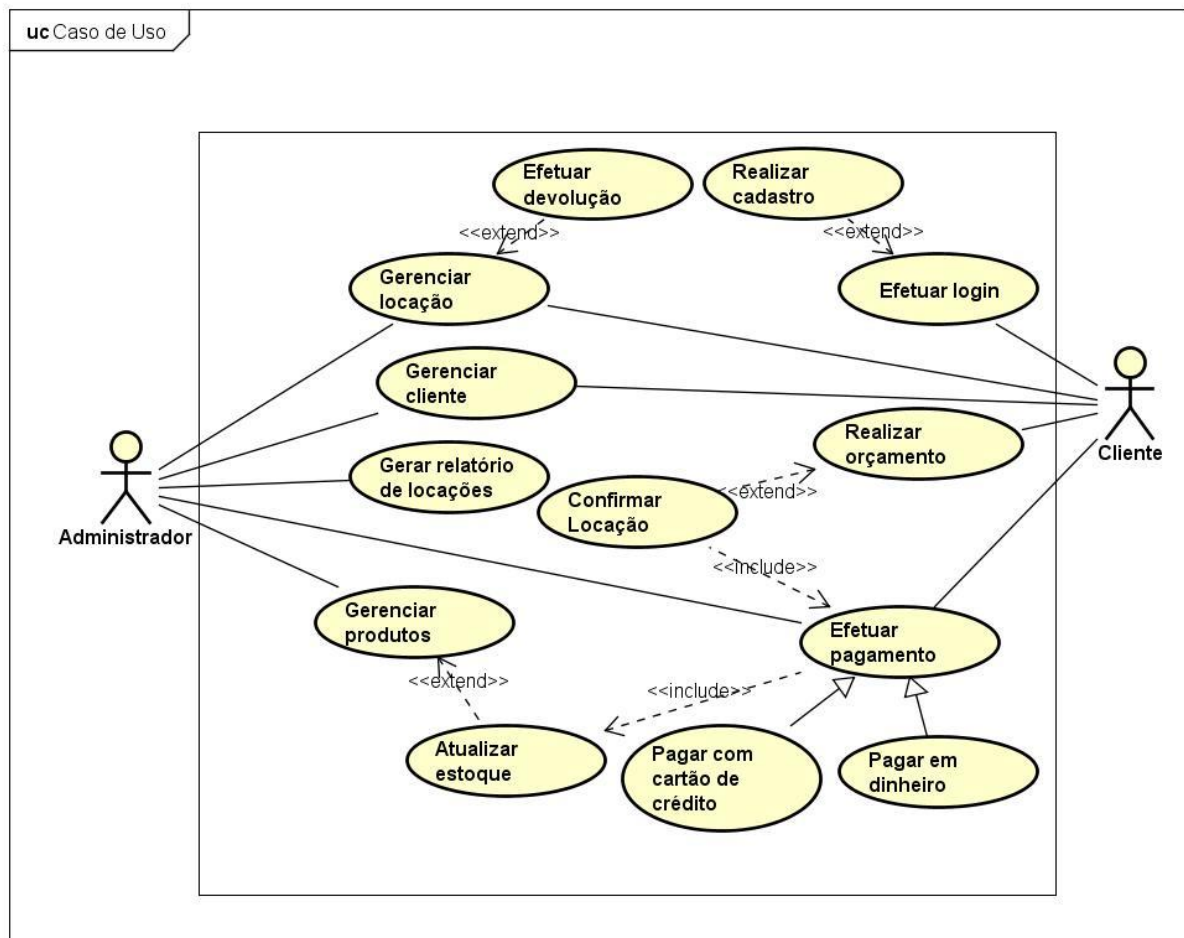
A empresa está no ramo de locação de artigos para festas, atendendo toda a região. Esta necessita de um sistema para aprimorar o processo de locação, agilizando esse processo de tal maneira que facilite a interação do cliente com a empresa. Além de facilitar essa interação com o cliente, o sistema permitirá a pessoa ver os dias disponíveis para o seu evento, podendo realizar apenas a locação de nossos itens, e também contratar nossos serviços de decoração. O sistema viabiliza o processo da locação, evitando inúmeras anotações e perda de informações.

8.5 Lista de Requisitos

ID	FUNCIONALIDADE	DESCRIÇÃO	ATOR
R1	Gerenciar Produto	Permite consultar, alterar e visualizar os produtos.	Administrador
R2	Gerenciar Cliente	Permite cadastrar clientes, para facilitar futuras locações do mesmo cliente.	Cliente/ Administrador
R3	Gerenciar locação	Permite visualizar informações das datas das locações e realizar a devolução dos produtos locados.	Administrador
R4	Efetuar Login	Permite o cliente ou o administrador logar no sistema.	Cliente/ Administrador
R5	Realizar Orçamento	Permite fazer um orçamento do que se pretende locar.	Cliente
R6	Efetuar pagamento	Permite escolher a forma de pagamento e efetuar a mesma	Cliente
R7	Gerar Relatório	Permite gerar relatórios das locações.	Administrador

Tabela 1 Lista de Requisitos

8.6 Diagrama de Caso de Uso



powered by Astah

Figura 9 Diagrama de Caso de Uso

8.7 Documentação de Caso de Uso

8.7.1 Documentação de Caso de Uso: UC 01 - Gerenciar Produtos

Nome do caso de uso	Gerenciar Produto
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	Usuário
Resumo	Este caso de uso tem como objetivo, consultar, alterar e visualizar todos os produtos com seu todos seus atributos, e também permite a atualização do estoque.
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	Após o termino da função o sistema registra as informações do cliente no banco de dados.
Fluxo Principal	O fluxo principal inicia quando o administrador começa a introduzir as funcionalidades desejadas.
Ações do Ator	Ações do Sistema
1. O ator entra na tela de Produtos	
	2. O sistema retorna uma tela de contendo todos os itens do estoque.
3. O ator visualiza os produtos para atualização do estoque.	
	4. O sistema guarda as informações no banco de dados e retorna uma mensagem[MSG.001]
Restrições/Validações	

Tabela 2 Documentação de Caso de Uso: UC 01 - Gerenciar Produtos

8.7.2 Documentação de Caso de Uso: UC 02 – Gerenciar Clientes

Nome do caso de uso	Gerenciar clientes
Caso de Uso Geral	
Ator Principal	Cliente
Atores Secundários	Administrador
Resumo	Este caso de uso tem como função cadastrar clientes para facilitar futuras locações do mesmo cliente, e também permite que possa visualizar e alterar dados em seu cadastro
Pré-Condições	O ator principal deve entrar no sistema, para se cadastrar e executar as próximas tarefas.
Pós-Condições	Após o termino da função de cadastro o sistema registra as informações do cliente no banco de dados.
Fluxo Principal	O fluxo principal inicia quando o cliente entra no sistema e clica na opção Cadastre-se.
Ações do Ator	Ações do Sistema
1. O ator entra na tela de cadastro	
	2. O sistema disponibiliza uma tela com um formulário para preenchimento de dados. [MSG.002]
3. O ator preenche o formulário e clica na função salvar.	
	4. O sistema salva as informações no banco de dados, e retorna uma mensagem ao cliente. [MSG.003]
Fluxo Alternativo - Alteração de dados	
Ações do Ator	Ações do Sistema
1. O cliente solicita alteração de cadastro	
	2. O sistema retorna um formulário com suas informações já preenchidas
3. O administrador realiza sua alteração	
	4. O sistema automaticamente retorna uma mensagem. [MSG.004]
Restrições/Validações	

Tabela 3 Documentação de Caso de Uso: UC 02 – Gerenciar Clientes

8.7.3 Documentação de Caso de Uso: UC 03 - Gerenciar Locação

Nome do caso de uso	Gerenciar Locação
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	Usuário
Resumo	Este caso de uso permite que o administrador possa se cadastrar e assim ter acesso às áreas privilegiadas do sistema, além de alterar, excluir, visualizar ou consultar qualquer informações relacionada ao sistema, além de dar baixa em todas as devoluções efetuadas.
Pré-Condições	O ator deverá preencher um formulário com seus dados pessoais para se cadastrar no sistema e assim efetuar novas tarefas.
Pós-Condições	Após a confirmação de todas as informações será necessário logar no sistema.
Fluxo Principal	O fluxo principal inicia quando o administrador solicita ao sistema a funcionalidade cadastro para inicializar os processos.
Ações do Ator	Ações do Sistema
1. O ator solicita novo cadastro	
	2. O sistema disponibiliza um formulário com campos de preenchimento de dados. [MSG.005]
3. O ator preenche os campos, confirma suas informações e solicita a opção de salvar.	
	4. Após solicitar função salvar, o sistema guarda as informações no banco de dados.
	5. O sistema automaticamente ira retornar uma tela de login de usuário.
6. O ator informa login e senha cadastrados anteriormente.	
	7. O sistema retornara a uma tela de carregamento de login.
	8. Assim que carregado o sistema retorna uma tela informando que o usuário está logado. [MSG.006]
Fluxo de Exceção	
Ações do Ator	Ações do Sistema
	1. Caso o administrador não preencha os campos obrigatórios, o sistema retorna uma mensagem. [MSG.007]
	2. Se algum campo preenchido não corresponder, o sistema retorna uma mensagem. [MSG.008]
	3. Caso o E-mail do usuário seja inválido, retornar mensagem de erro. [MSG.009]
	4. Caso o CPF do usuário seja inválido, retornar mensagem de erro.

	[MSG.0010]
	5. Caso o usuário preencha os campos obrigatórios com dados já existentes o sistema retorna uma mensagem. [MSG.0011]
Fluxo Alternativo - Alteração de dados	
Ações do Ator	Ações do Sistema
1. O administrador solicita alteração de cadastro	
	2. O sistema retorna um formulário com suas informações já preenchidas
3. O administrador realiza sua alteração	
	4. O sistema automaticamente retorna uma mensagem. [MSG.0012]
Restrições/Validações	

Tabela 4 Documentação de Caso de Uso: UC 03 - Gerenciar Locação

8.7.4 Documentação de Caso de Uso: UC 04 – Efetuar Login

Nome do caso de uso	Efetuar Login
Caso de Uso Geral	
Ator Principal	Cliente
Atores Secundários	Administrador
Resumo	Este caso de uso permite ao cliente e ao administrador a mesma função de efetuar o login no sistema.
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	Depois de estar logado no sistema, podem executar qualquer uma das opções disponíveis.
Fluxo Principal	O fluxo principal inicia quando um dos dois estejam logados no sistema.
Ações do Ator	Ações do Sistema
1. O ator solicita tela de login	
	2. O sistema retorna uma tela pedindo e-mail e senha para se conectar. [MSG.0013]
3. O ator informa os dados necessários ao sistema.	
	4. O sistema verifica as informações no banco de dados e retorna uma mensagem ao usuário [MSG.0014]
	5. O sistema transfere o usuário para a tela principal do sistema.
Fluxo de Exceção	
Ações do Ator	Ações do Sistema
	1. Se algum campo preenchido não corresponder, o sistema retorna uma mensagem. [MSG.0015]
	2. Caso o E-mail do usuário seja inválido, retornar mensagem de erro. [MSG.0016]
	3. Caso o Senha do usuário seja inválido, retornar mensagem de erro. [MSG.0017]
Fluxo Alternativo	
Ações do Ator	Ações do Sistema
1. O ator entra na tela de login	
	2. O sistema retorna uma tela pedindo e-mail e senha para se conectar e retornara a [MSG.0013]
3. O ator informa os dados necessários ao sistema.	
	4. O sistema verifica no banco de dados e não encontra dados, então automaticamente retorna uma mensagem. [MSG.0018]
Restrições/Validações	

Tabela 5 Documentação de Caso de Uso: UC 04 – Efetuar Login

8.7.5 Documentação de Caso de Uso: UC 05 – Realizar Orçamento

Nome do caso de uso	Realizar Orçamento
Caso de Uso Geral	
Ator Principal	Cliente
Atores Secundários	Administrador
Resumo	Este caso de uso permite que o usuário possa gerenciar suas informações relacionadas ao orçamento. Isso inclui adicionar, alterar e remover item do carrinho de compra e também cancelar orçamento.
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	O orçamento só será aprovado após a realização do pagamento.
Fluxo Principal	O fluxo principal inicia quando o usuário adicionar algum item no carrinho de compra e finalizar seu orçamento.
Ações do Ator	Ações do Sistema
1. O ator solicita o orçamento	
	2. O sistema retorna uma tela de produtos contendo todos itens, preços e quantidades dos produtos. [MSG.0019]
3. O ator seleciona os produtos para o carrinho de compras.	
	4. O sistema grava as informações no banco de dados
	5. O sistema retorna uma tela para finalizar a locação[MSG.0020]
6. O ator seleciona o endereço, o dia e a forma de pagamento	
	7. O sistema verifica a disponibilidade de data disponível e retorna uma mensagem ao cliente. [MSG.0021]
8. O ator confirma a locação	
	9. O sistema retorna uma mensagem de sucesso e gera um comprovante. [MSG.0022]
Fluxo de Exceção	
Ações do Ator	Ações do Sistema
	1. Caso o orçamento e o pagamento já esteja aprovado, não há possibilidade de cancelamento fora do contrato.
Fluxo Alternativo	
Ações do Ator	Ações do Sistema
	1. Caso o cliente não marque uma forma de pagamento, o sistema retorna uma mensagem [MSG. 0023]
	2. Caso o cliente não marque uma local de entrega, o sistema retornara uma mensagem [MSG. 0024]
Restrições/Validações	

Tabela 6 Documentação de Caso de Uso: UC 05 – Realizar Orçamento

8.7.6 Documentação de Caso de Uso: UC 06 – Efetuar Pagamento

Nome do caso de uso	Efetuar Pagamento
Caso de Uso Geral	
Ator Principal	Cliente
Atores Secundários	Administrador
Resumo	Este caso de uso permite que o cliente possa escolher a forma de pagamento para realização da locação dos produtos.
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	Após o pagamento será feita confirmação da locação dos serviços da empresa.
Fluxo Principal	O fluxo principal inicia quando o cliente escolhe a forma de pagamento.
Ações do Ator	Ações do Sistema
1. O ator finaliza o orçamento	
	2. O sistema retorna uma tela com o valor do seu orçamento e a forma de pagamento que o cliente deseja. [MSG.0025]
3. O ator seleciona a forma de pagamento e finaliza o orçamento.	
	4. O sistema grava no banco de dados e retorna uma mensagem. [MSG.0026]
Restrições/Validações	

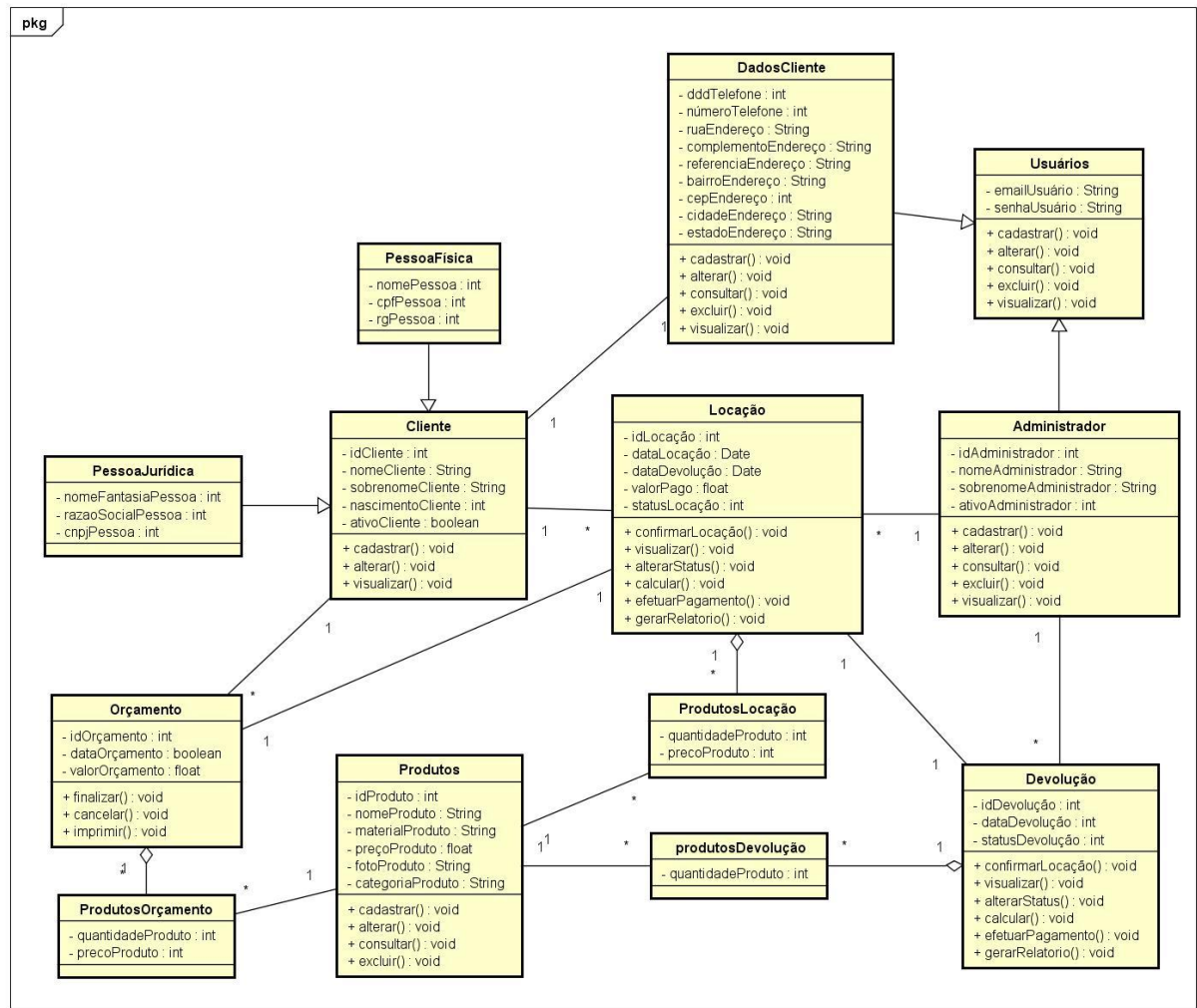
Tabela 7 Documentação de Caso de Uso: UC 06 – Efetuar Pagamento

8.7.7 Documentação de Caso de Uso: UC 07 – Gerar Relatório

Nome do caso de uso	Gerar Relatório
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	Cliente
Resumo	Este caso de uso tem como objetivo gerar relatórios para a empresa
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	Após gerar os relatórios a empresa terá todas as informações lucrativas e negativas da empresa.
Fluxo Principal	O fluxo principal inicia quando o administrador solicita os relatórios gerenciais da empresa.
Ações do Ator	Ações do Sistema
1. O ator entra na tela de gerar relatórios e solicita-o.	
	2. O sistema gera o relatório no banco de dados e emite em uma tela de retorno.
3. O ator visualiza e finaliza a tarefa.	
Restrições/Validações	

Tabela 8 Documentação de Caso de Uso: UC 07 – Gerar Relatório

8.8 Diagrama de Classe



powered by Astah

Figura 10 Diagrama de Classes

8.9 Modelo Entidade Relacionamento – MER

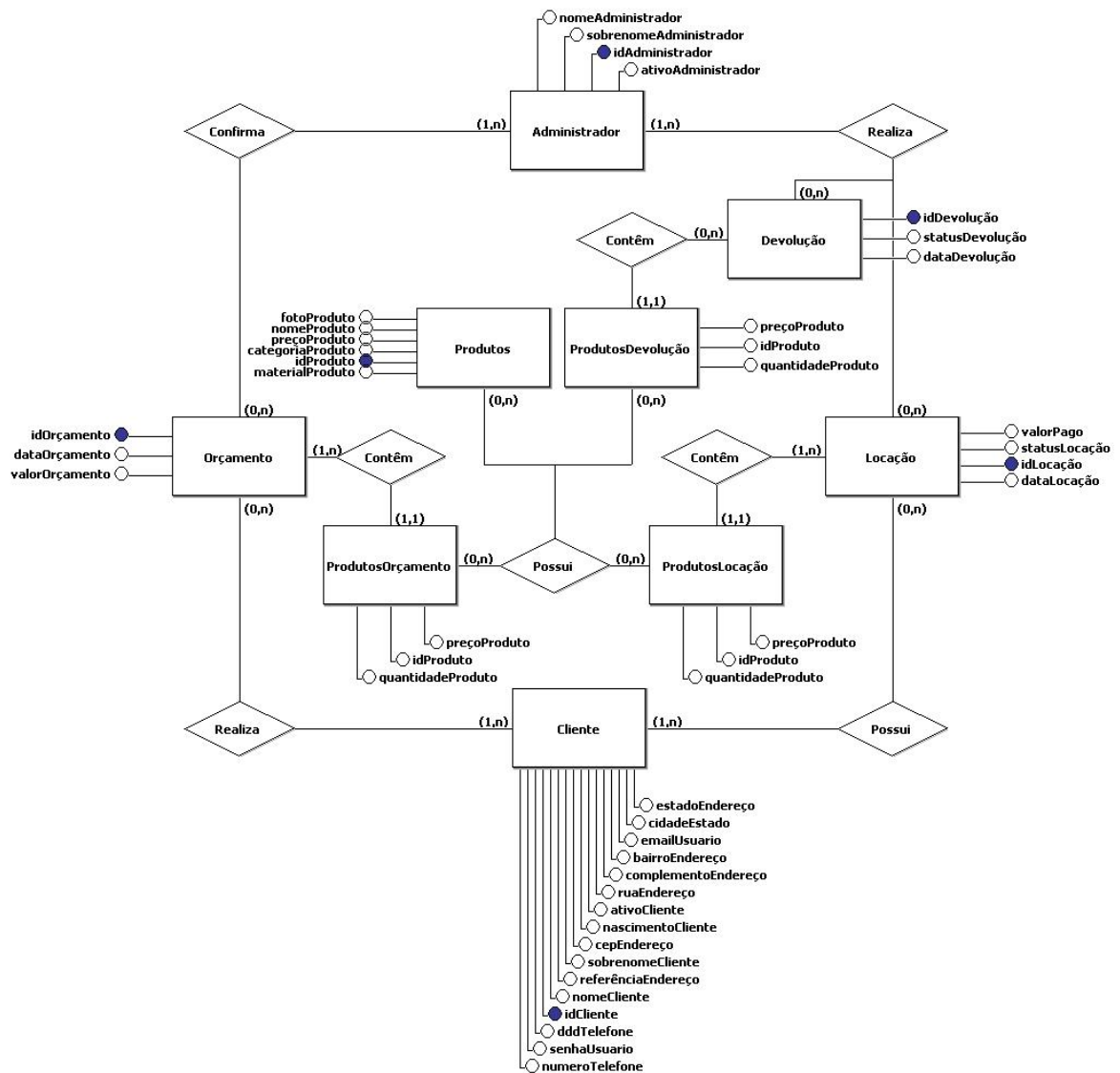


Figura 11 MER

8.10 Modelo Relacional Normalizado – MRN

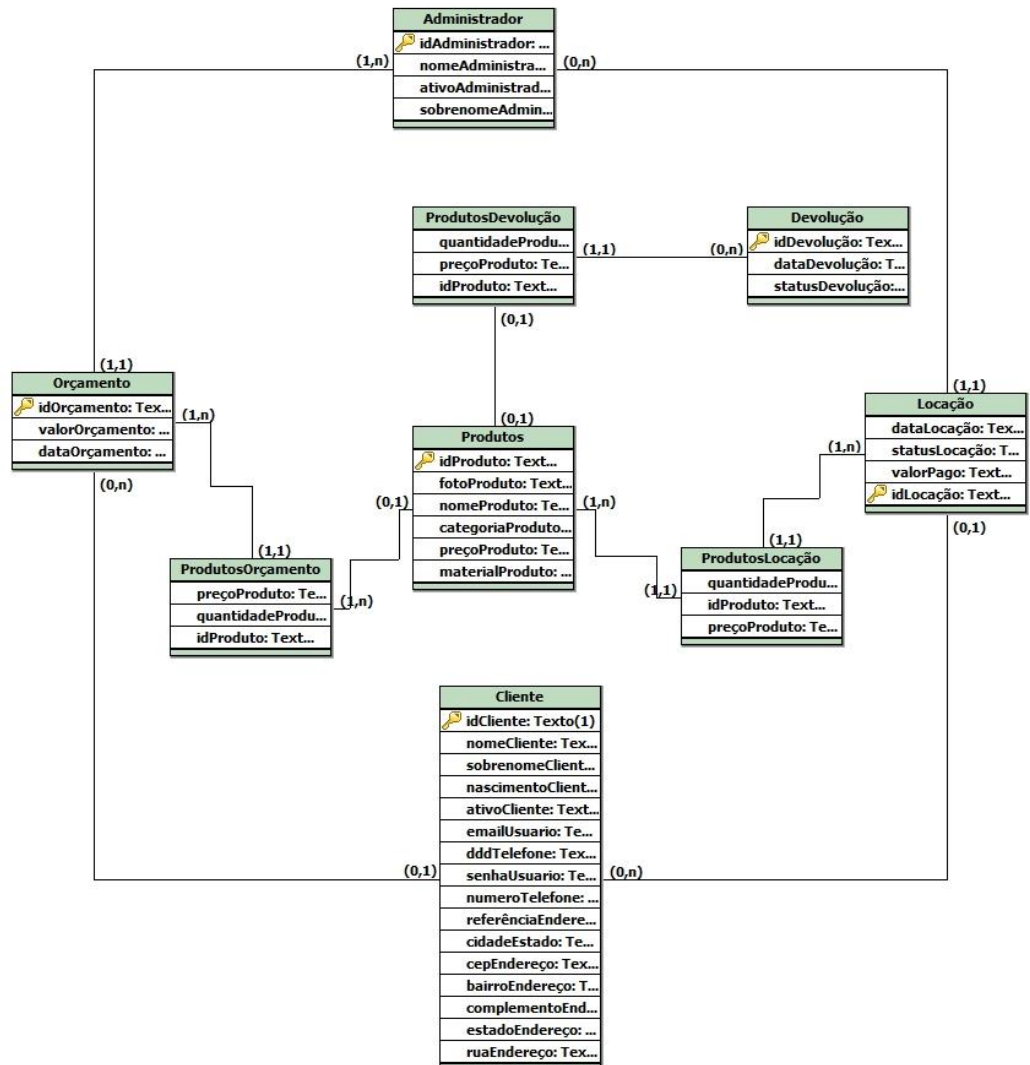


Figura 12 MRN

8.11 Prototipação

8.11.1 Tela Inicial

Tela inicial do no nosso sistema, quando o cliente acessa o site, ela já abre diretamente. Possui um menu pra ele navegar, um bloco com slide de decorações já realizadas, e imagens de produtos em destaque.



Figura 13 Tela Inicial

8.11.2 Tela de Login

Tela em que o usuário realiza o login no sistema, se ele não possuir cadastro, tem a opção pra ele realizar seu cadastro.



A captura de tela de uma janela de login intitulada "Faça seu Login". A janela possui um título "Faça seu Login" e um ícone de fechar (X) no canto superior direito. Abaixo do título, há dois campos de entrada: "Login*" e "Senha*", ambos com ícones de lupa para pesquisa. Abaixo dos campos, há um link "Cadastre-se" em azul. Na parte inferior direita, há dois botões: "Entrar" em verde e "Fechar" em vermelho.

Figura 14 Tela de Login

8.11.3 Tela de Cadastro de Cliente

Tela em que o cliente realiza seu cadastro no sistema, informando seus dados.



Cadastro de Cliente

Nome*: Sobrenome*:

Login*: Email*:

Senha*: Confirmar Senha*:

CPF: Telefone*:

UF: Cidade:

Figura 15 Tela de Cadastro do Cliente

8.11.4 Tela de Produtos

Tela em que aparece nossos produtos que podem ser locados, e o cliente seleciona o produto que quer e é redirecionado a uma página específica do produto.

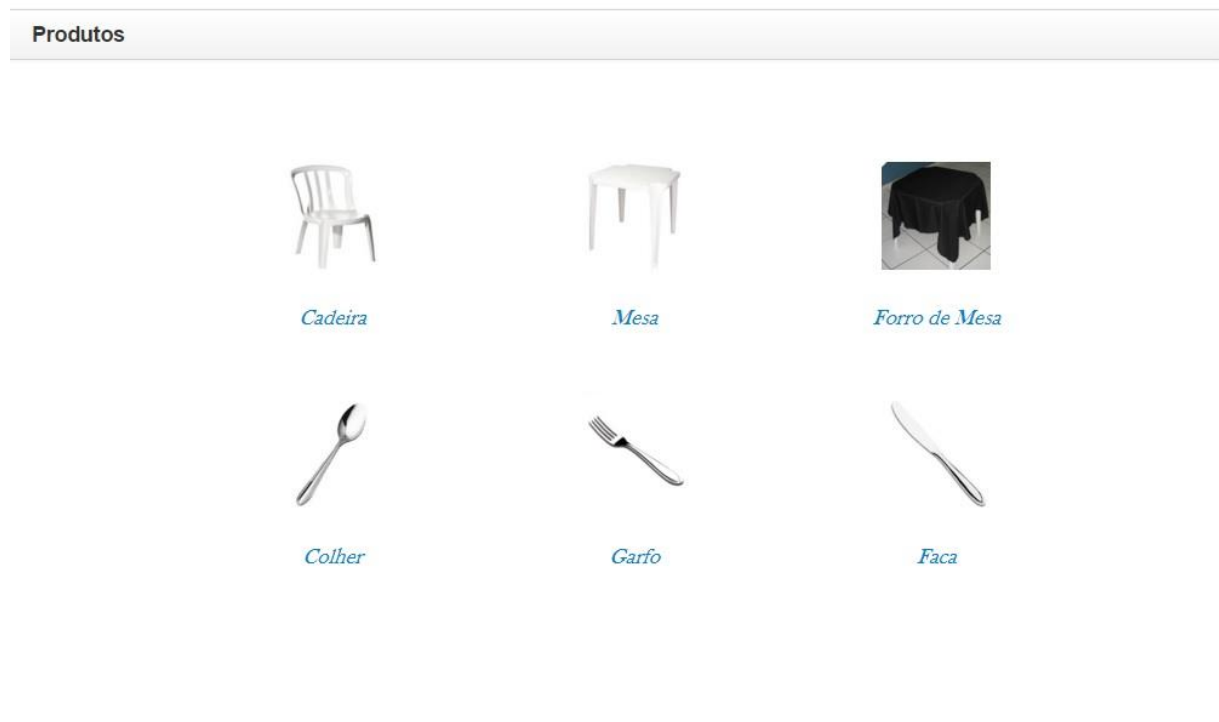


Figura 16 Tela de Produtos

8.11.5 Tela de um Produto Especifico

Tela de um de nossos produtos disponíveis pra locação, em que é encontrado as características do produto, e é possível verificar a disponibilidade do produto no dia que quiser realizar a locação, e também a opção de adicionar ao carrinho.

Cadeira



CODIGO: 001
NOME: Cadeira
CATEGORIA: Cadeiras

★★★★★

R\$ 0,80
Em Estoque

Quantidade:

Data:

Figura 17 Tela de um Produto Especifico

8.11.6 Tela de Gerenciar Locação

Tela em que o administrador pode gerenciar as locações, fazendo devoluções ao estoque e também gerar relatórios das locações.

Decoração Chuva de Ouro
Sua Festa em grande Estilo!!!

Faça seu Login [Login](#) [Sair](#)

[Início](#) [Empresa](#) [Administração](#) [Produtos](#) [Decoração](#) [Contato](#) [Carrinho](#) [Cadastre-se](#)

Gerenciar Locação

	Codigo	Data Locação	Data Devolução	Valor	Status
●	001	30/10/2015	02/11/2015	R\$ 320	Devolvido
●	002	05/11/2015	08/11/2015	R\$ 500	Devolvido
●	003	10/11/2015	13/11/2015	R\$ 400	Pendente
●	004	15/11/2015	18/11/2015	R\$ 600	Pendente
●	005	17/11/2015	19/11/2015	R\$ 350	Pendente
Gerar Relatório					Fazer Devolução

Figura 18 Tela de Gerenciar Locação

8.11.7 Tela do Carrinho

Tela em que aparece os itens que o usuário está colocando no seu carrinho, ou seja, os produtos que o usuário pretende locar, podendo então finalizar sua locação.

Decoração Chuva de Ouro
Sua Festa em grande Estilo!!!

Faça seu Login [Login](#) [Sair](#)

[Início](#) [Empresa](#) [Produtos](#) [Decoração](#) [Contato](#) [Carrinho](#) [Cadastre-se](#)

Carrinho

Código	Descrição	Categoria	Quantidade	Valor	Remover
001	Cadeira	Cadeira	200	R\$ 160,00	
002	Mesa	Mesa	50	R\$ 60,00	
005	Garfo	Talher	150	R\$ 75,00	
006	Colher	Talher	50	R\$ 25,00	
				Total à pagar: R\$ 320,00	

[Adicionar Itens](#) [Finalizar Locação](#)

Figura 19 Tela do Carrinho

8.11.8 Tela da Decoração

Tela em que o cliente verifica se temos disponibilidade no dia do evento que ele quer para realizarmos a sua decoração do evento.



The screenshot displays a web interface for 'Decoração Chuva de Ouro'. At the top, a golden banner features the text 'Decoração Chuva de Ouro Sua Festa em grande Estilo!!!'. Below the banner is a navigation menu with links for 'Início', 'Empresa', 'Produtos', 'Decoração', 'Contato', 'Carrinho', and 'Cadastre-se'. To the right of the menu, there is a 'Login' button and a 'Sair' button, with the text 'Por favor faça seu Login' above them. The main content area is titled 'Verificar disponibilidade' and contains two input fields: 'Data:' with a placeholder 'dd/mm/aaaa' and 'Hora:' with a placeholder '--:--'. A green 'Verificar' button is positioned below the input fields.

Figura 20 Tela de Decoração

CONSIDERAÇÕES FINAIS

O projeto tem como objetivo analisar e desenvolver um sistema web para gerenciamento de locação de artigos para festas. Analisando esse projeto, foi verificado que a empresa estava tendo dificuldade em controlar o estoque de produtos e locações. No ambiente financeiro da empresa, vimos que até o momento está controlada, caso precise de algo nessa área futuramente poderá ser implementado dentro do sistema.

As pesquisas realizadas deram impulso pra iniciar o processo de desenvolvimento e trouxe os resultados no projeto final, os objetivos propostos foram alcançados, através de entrevistas e uma análise direta com o cliente.

A metodologia usada foi a qualitativa, visando obter um contato direto com o cliente, tornando assim mais fácil entender o real problema que a empresa enfrenta. Uma das principais dificuldades encontradas dentro da pesquisa foi a automatização do sistema, pois a proprietária da empresa domina pouco de computador, é uma pessoa leiga nessa área, mas ao perceber que o sistema facilitaria e agilizaria o processo dentro da empresa, a proprietária se interessou bastante e se mostrou satisfeita com a ideia proposta.

O período de desenvolvimento é curto, caso tivéssemos mais tempo, percebendo o alto crescimento e a evolução dos celulares e smartphones, poderíamos criar uma versão mobile, onde facilitaria mais ainda a interação entre o cliente e a empresa, tornando o processo de locação mais prático e interativo, onde pela palma da sua mão, o cliente poderia realizar sua locação sem sair de casa.

REFERÊNCIAS BIBLIOGRÁFICAS

SOMMERVILLE, Ian **Engenharia de Software**. 8ª Ed. Addison-Wesley, 2007.

BEZERRA, Eduardo **Princípios de análise e projeto de sistemas com UML** 2ª Ed. Elsevier, 2007.

GUEDES, Gilleanes T. A. **UML 2 : Uma abordagem prática**. 2º ed. São Paulo: Novatec Editora, 2011.

PRESSMAM, Roger S. **Engenharia de Software**. São Paulo: Pearson Malkron Books, 1995.

TEOREY TOBY, **Projeto e Modelagem de Bancos de Dados**. Tradução da Quarta Edição. São Paulo: Elsevier, 2006.

BOOCH, Grandy **UML: guia do usuário** / Grandy Booch, James Rumbaugh, Ivar Jacobson; 2ª Edição Elsevier, 2005.

CUNNINGHAM, M. J. **B2B: business to business: como implementar estratégias de e-commerce entre empresas**. Tradução de Carlos Augusto Caldas de Moraes. Rio de Janeiro : Campus, 2001.

LAUDON, KENNETH C.. **Sistemas de Informações Gerenciais: Administrando a economia digital**. Tradução Arlete Simille Marques, Revisão Técnica Érico Veras

ALVES, Luiz. **Vencendo na Economia Digital**. São Paulo: Makron, 2002.

BHON, Daniela. **Sites Dinâmicos com Dreamweaver, PHP & MySQL**. Editora Visual Books. Florianópolis - SC, 2004.

APÊNDICE

Apêndice A - Entrevista

Data: 05/04/2015 Tempo: 25 min

Entrevistadores: Danillo Cardoso de Aguas e Luís Henrique Pessoa da Silva

Empresa: Decoração Chuva de Ouro

Entrevistada: Claudia Jacinto Cardoso (Proprietário).

1 - A empresa já possui um sistema para sua Gestão?

R: Não

2 - Como funciona o atendimento de sua empresa?

R: Nosso atendimento funciona de uma forma, onde os clientes possam vir até nosso estabelecimento para fazer a locação de seus produtos.

3 - A empresa sempre utilizou este método de atendimento?

R: Sim, até hoje estamos com essa forma de atendimento.

4 - De que forma e feita a anotação das locações e serviços?

R: As anotações são feitas em uma agenda comprada com as datas dos serviços.

5 - Já ouviu alguma perda de informações por causa das anotações em papéis?

R: Sim, várias vezes.

6 - Qual é a forma do controle de estoque da empresa?

R: A única forma que utilizamos é ficar atentos nas locações feitas para não ter nenhum problema de ficar sem mercadorias para outros clientes.

7 - Já ocorreu de locar algum produto sem ter ele no estoque? Como este problema foi resolvido?

R: Já sim, várias vezes. Quando ocorreu de faltar produtos a empresa teve que correr atrás de outra empresa que tivesse o mesmo produto para não deixar o cliente na mão.

8 - Algum cliente já ficou insatisfeito com o atendimento da empresa?

R: Sim, muitos clientes questionam a falta de um sistema de atendimento mais organizado na empresa.

9 - Já aconteceu de ter alguma perda de locação por falta de um sistema que pudesse ajudá-lo?

R: Sim.

10 - Já ouviu algum tipo de erro por causa do diálogo entre você e o cliente?

R: Sim, várias vezes.

11 - A empresa fornece algum tipo de relatório para que o cliente possa conferir os serviços contratados?

R: Não.

Apêndice B - Glossário de Mensagens

Código de Rastreabilidade	Descrição da Mensagem	Caso de Uso
MSG 001	Estoque atualizado!	1. Gerenciar Produtos
MSG 002	Preencha o formulário!	2. Gerenciar Clientes
MSG 003	Salvo com sucesso!	2. Gerenciar Clientes
MSG 004	Dados atualizados!	2. Gerenciar Clientes
MSG 005	Preencha os campos!	3. Gerenciar Locação
MSG 006	Logado com sucesso!	3. Gerenciar Locação
MSG 007	Campos obrigatórios!	3. Gerenciar Locação
MSG 008	Preencha os campos corretamente!	3. Gerenciar Locação
MSG 009	E-mail invalido!	3. Gerenciar Locação
MSG 0010	CPF invalido!	3. Gerenciar Locação
MSG 0011	Usuário já cadastrado!	3. Gerenciar Locação
MSG 0012	Alteração realizada!	3. Gerenciar Locação
MSG 0013	Digite e-mail e senha!	4. Efetuar Login
MSG 0014	Verificando dados!	4. Efetuar Login
MSG 0015	Preencha os campos corretamente!	4. Efetuar Login
MSG 0016	E-mail invalido!	4. Efetuar Login
MSG 0017	Senha invalida!	4. Efetuar Login

MSG 0018	Dados não encontrados!	4. Efetuar Login
MSG 0019	Selecione os produtos desejados!	5. Realizar Orçamento
MSG 0020	Deseja finalizar o orçamento?	5. Realizar Orçamento
MSG 0021	Disponível!	5. Realizar Orçamento
MSG 0022	Locação realizada com sucesso!	5. Realizar Orçamento
MSG 0023	Por favor selecione a forma de pagamento!	5. Realizar Orçamento
MSG 0024	Por favor digite o local de entrega!	5. Realizar Orçamento
MSG 0025	Selecione a forma de pagamento!	6. Efetuar Pagamento
MSG 0026	Orçamento finalizado com sucesso!	6. Efetuar Pagamento
MSG 0027	Relatório gerado com sucesso!	7. Gerar Relatório

Tabela 9 Glossário de Mensagens

Apêndice C - Cronograma

ATIVIDADES	MAR/15	ABR/15	MAI/15	JUN/15	JUL/15	AGO/15	SET/15	OUT/15	NOV/15
Elaboração do Projeto	X	X							
Coleta de Dados		X							
Levantamento de Requisitos		X							
Implementação			X	X					
Apresentação e discussão do dados						X			
Testes e Validação								X	
Documentação			X	X	X				
Desenvolvimento						X	X	X	
Conclusão									X
Entrega do TCC									X

Tabela 10 Cronograma